High-Rank Irreducible Cartesian Tensor Decomposition and Bases of Equivariant Spaces

Shihao Shao[†]   SHAOSHIHAO@PKU.EDU.CN
State Key Lab of Vascular Homeostasis and Remodeling
Department of Bioinformatics
School of Basic Medical Sciences, Peking University

Yikang Li   LIYK18@PKU.EDU.CN
State Key Lab for General AI
School of Intelligence Science and Technology
Peking University, China

Zhouchen Lin[†]   ZLIN@PKU.EDU.CN
State Key Lab for General AI
School of Intelligence Science and Technology, Peking University
Pazhou Laboratory (Huangpu), China

Qinghua Cui[†]   CUIQINGHUA@BJMU.EDU.CN
State Key Lab of Vascular Homeostasis and Remodeling
Department of Bioinformatics
School of Basic Medical Sciences, Peking University

Editor: Michael Mahoney

Abstract

Irreducible Cartesian tensors (ICTs) play a crucial role in the design of equivariant graph neural networks, as well as in theoretical chemistry and chemical physics. Meanwhile, the design space of available linear operations on tensors that preserve symmetry presents a significant challenge. The ICT decomposition and a basis of this equivariant space are difficult to obtain for high-rank tensors. After decades of research, Bonvicini (2024) has recently achieved an explicit ICT decomposition for $n = 5$ with factorial time/space complexity. In this work we, for the first time, obtain decomposition matrices for ICTs up to rank $n = 9$ with reduced and affordable complexity, by constructing what we call path matrices. The path matrices are obtained via performing chain-like contractions with Clebsch-Gordan matrices following the parentage scheme. We prove and leverage that the concatenation of path matrices is an orthonormal change-of-basis matrix between the Cartesian tensor product space and the spherical direct sum spaces. Furthermore, we identify a complete orthogonal basis for the equivariant space, rather than a spanning set (Pearce-Crump, 2023b), through this path matrices technique. Our method avoids the RREF algorithm and maintains a fully analytical derivation of each ICT decomposition matrix, thereby significantly improving the algorithm's speed to obtain arbitrary rank orthogonal ICT decomposition matrices and orthogonal equivariant bases. We further extend our result to the arbitrary tensor product and direct sum spaces, enabling free design between different spaces while keeping symmetry. The Python code is available at https://github.com/ShihaoShao-GH/ICT-decomposition-and-equivariant-bases, where the

---

[†]. To whom correspondence should be addressed

$n = 6, \dots, 9$ ICT decomposition matrices are obtained in 1s, 3s, 11s, and 4m32s on 28-core Intel® Xeon® Gold 6330 CPU @ 2.00GHz, respectively.

Keywords: irreducible Cartesian tensor decomposition, Clebsch-Gordan coefficient, parentage scheme, equivariant graph neural network, equivariant linear mapping

1. Introduction

The deployment of equivariant neural networks in physics, chemistry, and robotics often requires certain equivariance or invariance to effectively incorporate inductive bias. One typical equivariance requirement is equivariance under the $O(3)$ group (rotations and reflections). Equivariant Graph Neural Networks (EGNNs) are well-suited for this purpose, as they can maintain equivariance while processing features within the graph (Zaverkin et al., 2024; Simeon and De Fabritiis, 2024; Batzner et al., 2022). One of the most promising approaches to constructing equivariant layers involves leveraging irreducible representations. Depending on the basis of the space, two distinct types of irreducible representations are utilized in different EGNNs: spherical (spherical tensors) (Musaelian et al., 2023; Batatia et al., 2022) and Cartesian irreducible representations (Irreducible Cartesian Tensors, ICTs) (Zaverkin et al., 2024; Schütt et al., 2021) for $O(3)$. Spherical tensors serve as the starting point for EGNNs based on irreducible representations. However, they require Clebsch-Gordan transforms with significant computational overhead to derive desired irreducible representations from existing ones during each inference, making them computationally intensive. In contrast, ICTs are more efficient, as high-rank representations can be easily computed using tensor products (specifically, outer products in EGNNs) and contractions, making them more computationally friendly for most deep learning frameworks. The ICTs are obtained through the ICT decomposition matrices H,

$$\text{vec}\left(\hat{T}^{(n;l;q)}\right) = H^{(n;l;q)}\text{vec}\left(T^{(n)}\right), \tag{1}$$

where $\hat{T}^{(n;l;q)}$ is an ICT, vec is the vectorization operation via flattening the tensor, $T^{(n)}$ is a given tensor, n represents the rank of the tensor, and l is called the weight (the order of irreducible representations, which is distinct from the learnable weights in neural networks) of the irreducible representation, and q denotes the label (since the multiplicity for given n and l is sometimes greater than 1). In this paper, we adopt the convention that vec flattens a tensor by stacking its entries from the rightmost to the leftmost index; concretely, $\text{vec}(T) = \{T_{\dots 111}, T_{\dots 112}, \cdots\}$. The decomposition matrices $H^{(n;l;q)}$ should be constructed such that $\hat{T}^{(n;l;q)}$ are irreducible. A formal definition is introduced in Section 3.1. The challenge lies in the fact that $H^{(n;l;q)}$ of rank $n > 5$ are unknown. The recent development of EGNNs research benefits from the irreducible decomposition matrices H obtained from the physics and chemistry communities. The decomposition matrices for $n = 4$ have been established by Andrews and Ghoul (1982). However, only very recently has Bonvicini (2024) obtained the decomposition matrices for $n = 5$. Due to the lack of high-rank ICT decomposition matrices, it is impossible to construct high-rank Cartesian EGNNs that incorporate valid equivariant linear layers. That is because only irreducible representations of same weight can be linearly combined, but the irreducible representations under Cartesian basis are mixed together. In other words, the irreducible representations of the same weight are impossible to be separated, so we cannot linearly combine those of the same type, which

limits the available rank of irreducible representations in EGNNs with full access to the equivariant linear layer designs. The fact that utilizing high-rank irreducible representations improves performance has been well investigated by predecessors, with the support of several empirical evidence. For example, see Table 2 in the paper by Batzner et al. (2022), where rank $l = 3$ NequIP significantly outperforms the same model with lower ranks. On the other hand, in the work done by Liao et al. (2023), Table 1(c) shows that the performance is better with higher ranks. Therefore, the matrices for high-rank orthogonal ICT decomposition are of great importance. Based on this, we can obtain valid equivariant linear operations applied to the Cartesian tensors, as these operations constitute the design space of EGNNs. In other words, what is the vector space of linear mappings f such that

$$\rho(g)f\left(\mathrm{vec}(T)\right) = f\left(\rho(g)\mathrm{vec}(T)\right) \tag{2}$$

where $g \in G$, an element of a group G. $T \in \left(\mathbb{R}^3\right)^{\otimes n}$ is an arbitrary rank-n tensor. $\rho : G \to GL\left(\left(\mathbb{R}^3\right)^{\otimes n}\right)$ is the group representation, and $GL(V)$ is a space consisting of the invertible matrices that operate on vector space V. A detailed introduction to the basic and necessary group theory is in Section 2.1. This vector space is called an equivariant space, also denoted as $\mathrm{End}_G\left(\left(\mathbb{R}^3\right)^{\otimes n}\right)$. More generally, for different input and output spaces, this equivariant space can be denoted as $\mathrm{Hom}_G\left(\left(\mathbb{R}^3\right)^{\otimes l}, \left(\mathbb{R}^3\right)^{\otimes k}\right)$. Pearce-Crump (2023b) leverages Brauer (1937)'s theorem to find the spanning set of the equivariant space. However, as the rank increases, the number of elements in the spanning set becomes greater than the dimension of the space. Thus, it is also essential to identify a basis to reduce the number of parameters required.

In this work, one of our contributions is the proposal of a method that does not rely on RREF algorithm, for efficiently obtaining ICT decomposition matrices, through which we achieve $6 \leq n \leq 9$ in an affordable setting. Several previous methods (Andrews and Ghoul, 1982; Dinçkal, 2013; Bonvicini, 2024) rely on evaluating the full permutation of n (e.g., fundamental isotropic Cartesian tensors) and $2n$ (e.g., natural projection tensors) indices, which requires factorial complexity, making it impractical for application to high-rank tensors. Bonvicini (2024) proposes a novel method to automate the procedure but does not reduce the complexity. The Reduced Row Echelon Form (RREF) algorithm is also required to run on an $n! \times n!$ matrix. Instead, we leverage the fact that the Cartesian tensor product space is equivalent to a direct product of spherical spaces, up to a change of basis. In spherical space, it is straightforward to identify the desired irreducible representations, as they are diagonalized; then, the inverse change-of-basis can bring us back to Cartesian space. The challenge lies in finding such a change-of-basis matrix. We prove that one can construct this matrix by concatenating the results of sequential contractions and the outer products of Clebsch-Gordan matrices (CG matrices) according to the parentage scheme (Coope et al., 1965), which we refer to as path matrices. The orthogonality of the path matrices further allows us to eliminate the time-consuming inverse operation, rendering this approach an orthogonal decomposition. Both theoretical and experimental analyses confirm the efficiency of this approach. Inspired by Schur's lemma, which states that only irreducible representations of the same weight can be linearly combined, we construct an equivariant basis via a path-matrix technique. This construction makes it possible to design high-rank

Method	Rank	Orthogonality	Time and Space Complexity	
			w.r.t. rank	w.r.t. dimension
Coope et al. (1965)	≤ 3	No	calculated by hand	
Andrews and Ghoul (1982)	≤ 4	No	calculated by hand	
Dinçkal (2013)	≤ 3	Yes	calculated by hand	
Bonvicini (2024)	≤ 5	No	factorial / sub-exponential	
Ours	≤ 9	Yes	exponential / polynomial	

Table 1: Comparison with previous works. The time and space complexities are with respect to rank and dimension, respectively.

equivariant linear layers in EGNNs. We then address the lack of a suitable parentage scheme for general spaces. To this end, we propose a generalized parentage scheme that yields an equivariant basis for arbitrary spaces. These advances collectively enable the efficient construction of equivariant layers across different spaces, thereby expanding the design space of EGNNs.

In summary, this work has five main contributions:

- We propose a method to efficiently obtain high-rank ICT decomposition matrices and, for the first time, obtain the decomposition matrices for $6 \leq n \leq 9$, significantly improving upon the current limit of $n = 5$, as shown in Table 1.

- We propose a general parentage scheme to extend ICT decomposition into a generalized ICT decomposition applicable to arbitrary tensor product spaces.

- We propose to directly obtain orthogonal bases of equivariant spaces, whereas the method proposed by Pearce-Crump (2023b) yields a spanning set, requiring an expensive post-processing step to extract a basis that is not orthogonal in general.

- Our method avoids the RREF algorithm and derives each ICT decomposition matrix in a fully analytical manner, thereby yielding a significantly more efficient algorithm.

- We extend our approach to efficiently obtain bases of equivariant spaces mapping between different spaces, including spherical tensor product spaces.

These contributions enable EGNNs to process high-rank ICTs and flexibly design equivariant layers between desired spaces. Furthermore, this work benefits the physics and chemistry communities by facilitating theoretical analysis based on high-rank ICTs.

Table 2: Mathematical symbols and their meanings

Symbol	Meaning	Symbol	Meaning
$\|\cdot\|_{col}$	The L_2-norm of an arbitrary column of $\cdot$	G	An arbitrary group
$SO(n)$	Special orthogonal group on n-dimensional space	$SU(n)$	Special unitary group on n-dimensional space
$O(n)$	Orthogonal group on n-dimensional space	$U(n)$	Unitary group on n-dimensional space
$\mathcal{V}$	A vector space	$\rho_{\mathcal{V}}$	A group representation
$\mathcal{V}^{\otimes n}$	A rank n tensor product space from $\mathcal{V}$	$N^{(n;l)}$	The multiplicity of the ICT of rank n and weight l
T	An arbitrary tensor $G \to \mathcal{V}$	$T^{(n)}$	A Cartesian tensor of rank n
$\hat{T}^{(n;l;q;p)}$	An ICT of rank n, weight l, index q, and parity p (sometimes omitted)	$Q_{\mathcal{V}}^{(l;p)}$	The set of all possible paths to decompose a tensor in space $\mathcal{V}$
$H^{(n;l;q)}$	An ICT decomposition matrix to obtain ICTs of rank n, weight l, and index q	$\hat{H}^{(\mathcal{V};l;q)}$	A general ICT decomposition matrix of space $\mathcal{V}$, weight l and index q
$C_{m_1 m_2 m_o}^{l_1 l_2 l_o}$	A CG coefficient with input weights l_1, l_2, output weight l_o, and the corresponding indices m_1, m_2, and m_o (the angular-momentum quantum numbers in physics)		
$C^{l_1 l_2 l_o}$	A CG tensor with input weights l_1, l_2, and output weight l_o	$\hat{C}^{l_1 l_2 l_o}$	A CG matrix by merging the first two dimensions of the original CG tensor
$P^{(path)}$	A path matrix generated by $path$, e.g., $(1 \to 3 \to 2)$	$\hat{P}^{(path)}$	A normalized path matrix generated by $path$
$D^{(\mathcal{V} \to \mathcal{V}')}$	A change-of-basis matrix from space $\mathcal{V}$ to $\mathcal{V}'$	$v^{(l;p)}$	A vector on which weight l and parity p irreducible representation acts
Hom_G	The set of G-equivariant maps from one space to another	End_G	The set of G-equivariant maps from one space to itself
irr	Returns all weights of irreducible representations in a decomposition	dim	The number of dimensions of a space
$(i_1 \overset{k_1}{\to} i_2 \overset{k_2}{\to} \dots)$	A path where k_* are bridge numbers (omitted if all being 1)	(l, p)	An irreducible representation space of weight l and parity p
$\otimes$	Tensor product	$\oplus$	Direct sum
$\odot$	Contraction	$\#$	Number of elements in a set

The following contents are organized as follows. In Section 2, we recall basic concepts from representation theory, spherical harmonics, tensors, and tensor product spaces, and we introduce the CG coefficients along with the parentage scheme. In Section 3, we formulate our problems: obtaining high-rank ICT decomposition matrices and bases for arbitrary tensor product spaces. In Section 4, we introduce our theory and method to obtain high-rank ICT decomposition matrices. In Section 5, we present the theory and method to obtain bases of equivariant spaces where the input and output spaces are the same. In Section 6, we extend the theory to the equivariant design spaces where the input and output spaces are different, and show the method to obtain bases for such general spaces. In Section 7, we review related works on designing EGNNs and calculating ICT decomposition matrices. In Section 8, we discuss the relation between Pearce-Crump (2023b) and this work, as well as the theory for $SO(n)$, $O(n)$, and $SU(n)$. We provide concluding remarks in Section 9, and introduce the reproducibility details in Section 10. The acknowledgments are presented after the main text. The proofs of supportive lemmas are shown in Appendix A. The complexity comparison with Bonvicini (2024) is in Appendix B. We give an example for generating rank-2 ICT decomposition matrices in Appendix C. The visualizations of ICT decomposition matrices and equivariant bases are shown in Appendix D. A list of mathematical symbols is provided in Table 2, and a mind map of this work is shown in Figure 1.

2. Preliminaries

In this section, we recap necessary mathematical preliminaries.

2.1 Group, Irreducible Representations and Spherical Spaces

We first recall some basic facts of group theory. A thorough introduction to group theory can be found in Inui et al. (2012) and Bröcker and Tom Dieck (2013). A group G is a set equipped with a binary operation $*$, such that the following five properties hold.

- **Non-emptiness**:
  The set G is non-empty.

- **Closure**:
  For all $g_1, g_2 \in G$, we have $g_1 * g_2 \in G$.

- **Associativity**:
  For all $g_1, g_2, g_3 \in G$, we have $(g_1 * g_2) * g_3 = g_1 * (g_2 * g_3)$.

- **Identity element**:
  There exists $e \in G$ such that, for every $g \in G$, it holds that $e * g = g * e = g$.

- **Inverse element**:
  For every $g \in G$ there exists $g^{-1} \in G$ satisfying $g * g^{-1} = g^{-1} * g = e$.

Group is a powerful tool to study symmetry. Rotations and rotations with reflections are both examples of groups. For example, it is easy to verify that 2-d rotations satisfy the above requirements: clockwise rotating by $30°$, followed by counterclockwise $15°$, is equal to clockwise rotating by $15°$, and the remained properties can be examined similarly. Until

now, the group remains abstract, and we cannot bring it into computation. Representation theory provides us with a bridge to connect group to linear algebra. A representation is a mapping $\rho : G \to GL(V)$ such that

$$\rho(g_1 * g_2) = \rho(g_1) \cdot \rho(g_2)$$

holds for any $g_1, g_2 \in G$, where $GL(V)$ is a space consisting of the invertible matrices that operate on vector space V. In other words, they are invertible matrices that operate on vector space V. Here, we introduce groups that we will consider in this work. The orthogonal group, $O(n)$, consisting of orthogonal matrices in $\mathbb{R}^{n \times n}$, represents n-dimensional rotations and reflections; The special orthogonal group, $SO(n)$, consisting of orthogonal matrices with determinant equaling 1, represents n-dimensional rotations; The unitary group, $U(n)$, consisting of unitary complex matrices in $\mathbb{C}^{n \times n}$, represents n-dimensional rotations and reflections on $\mathbb{C}^{n \times n}$, and $SU(n)$ with additional unit determinant constraint, represents rotations only. Consider a vector $v \in V$. A group element g acting on the vector v can therefore be represented as a matrix-vector multiplication $\rho(g) \cdot v$. Sometimes, one may also refer to the space of v as a representation space. If the results of the group action remain in the same space, i.e., $\rho(g)v \in V$, this space is also called a G-invariant space or subspace. It is important to note that the co-domain of ρ can differ, meaning that it can also be represented by matrices of other dimensions. In this paper, we denote the co-domain of ρ using a subscript when it should be made explicit, e.g., $\rho_{\mathbb{R}^3}$. Now, it suffices to define equivariance. We claim that a linear function $f : V \to V'$ is equivariant if and only if

$$\rho_{V'} f(x) = f(\rho_V(x)).$$

We first focus on the simpler case where $V = V'$. We will later generalize this to the case where $V \neq V'$. When it is necessary to emphasize the change in space, we refer to a linear mapping f as G-(V, V')-equivariant. We then introduce the definition of reducible and irreducible representation.

Definition 1 *A group representation $\rho_V : G \to GL(V)$ is said to be reducible, if there is a non-zero subspace $V' \subset V$, for which we have a representation $\rho_{V'}$, such that for every $g \in G$ and $v \in V'$, it holds that*

$$\rho_{V'}(g)v \in V',$$

where we also call V' a G-invariant subspace. Conversely, if there is no such subspace V' and the corresponding representation $\rho_{V'}$ on it for a given space V satisfying the above conditions, then we claim that ρ_V is irreducible.

If a representation is irreducible, then the vectors space it acts on is also said to be irreducible. The 3×3 rotation and reflection matrices are irreducible. More generally,

Proposition 2 *The natural representation $\{M \in GL(n, \mathbb{R}) \mid MM^\top = M^\top M = I\}$ of $O(n)$ is irreducible.*

Irreducible representations play a very important role in theoretical chemistry and physics, as they represent different symmetries. It is also one of the major building blocks for EGNNs in recent years. Note that the matrix representation will change if we change the basis of

7

*the space $\mathcal{V}$ it acts on. Thus, the representation will be different according to the basis we choose. In this paper, we first focus on the $O(3)$ group, but we also extend our conclusions to $O(n)$, $SO(n)$, $U(n)$, and $SU(n)$ in Section 8.2. For the spherical basis, each irreducible representations space is spanned by the **real** spherical harmonics Y_{lm} as a basis of different degree l and order m. The complex spherical harmonics $Y_l^m : S^2 \to \mathbb{C}$ are the solutions of Laplace's equation, which are in the form of associated Legendre polynomials P_l^m taking spherical coordinates (θ, φ) as variables (or equivalently (x, y, z) after some change-of-basis). l can take values from $\{\frac{1}{2}, 1, \frac{3}{2}, \dots\}$. They are highly related to $SU(2)$ group, which is of great importance for quantum physics. Leveraging the fact that $SU(2)$ is a double-cover of $SO(3)$, analysis implies that the real form of spherical harmonics Y_{lm} of integer l form a mapping from S^2 to S^{2l}, via $(x, y, z) \mapsto (Y_{l,-l}(x, y, z), \cdots, Y_{l,l}(x, y, z))$ that the weight-l $O(3)$ irreducible representation (the term weights in this paper refer to the order of irreducible representations, following the convention) operate on. In this case, $l \in \{1, 2, \dots\}$ and $m \in \{-l, \dots, l\}$, and accordingly $O(3)$ has 3×3, 5×5, $\dots$ matrices as representations. We sometimes use $(l = k)$ to denote a weight-k spherical space. As said above, an important property of spherical harmonics is:*

Proposition 3 *Given a space $\mathcal{V}$ spanned by the real spherical harmonics,*

$$\left\{ (Y_{l,-l}(v), \cdots, Y_{l,l}(v)) \in S^{2l} : v \in S^2 \right\},$$

it holds that, for any $v \in \mathcal{V}$ and $g \in O(3)$, we always have $\rho_{S^{2l}}(g)v \in \mathcal{V}$.

The above proposition directly follows from the fact that the mapping $(Y_{l,-l}, \cdots, Y_{l,l})$ is $O(3)$-equivariant. There is also a connection between spherical and Cartesian spaces, which is one of the building blocks of our arguments:

Lemma 4 *The weight one real spherical space is exactly the 3-d Cartesian space up to an orthogonal change-of-basis.*

This lemma is often utilized in some engineering implementations (Thomas et al., 2018; Weiler et al., 2018; Kondor et al., 2018). In this work, it is a key connecting the spherical and Cartesian space. Also, we have the following properties:

Proposition 5 *The l-degree real spherical harmonics are homogeneous harmonic polynomials, $Y_{lm} \in H_l(S^2)$, where Y_{lm} is orthogonal to $Y_{l'm'}$ for $l' \neq l$ or $m' \neq m$, under $L^2(S^2)$. $\{Y_{lm}\}$ forms an orthonormal basis for $(2l+1)$-dimensional vector spaces, with unit L_2-norms.*

2.2 Clebsch-Gordan Coefficients

An important tool utilized in our work is the Clebsch-Gordan coefficients (CG coefficients). They are real numbers, and describe how to couple vectors from two spherical spaces into a new one. There are some rules for this coupling. First, given two vectors in weights l_1 and l_2 spherical spaces, the weights of the resulting vectors can only lie in $|l_1 - l_2| \leq l_o \leq l_1 + l_2$. For each index m_o of the obtained vector $v_{l_o} \in (l = l_o)$, it is calculated by homogeneous quadratic polynomials. Specifically, we have

$$v_{m_o}^{(l_o)} = \sum_{m_o = m_1 + m_2} C_{m_1 m_2 m_o}^{l_1 l_2 l_o} v_{m_1}^{(l_1)} v_{m_2}^{(l_2)}, \tag{3}$$

and if we consider $O(n)$ instead of $SO(n)$, parity p should also be considered. The parity is a property additional to the rank and weight, being 1 or -1. Having -1 parity means that the representation will be inverse when the space is inverse, and vice versa. For example, the natural representations of $O(n)$ always operate on $p = -1$ vector spaces. Taking the parity into consideration, we can guarantee $O(n)$ equivariance, and equation (3) becomes

$$v_{m_o}^{(l_o;p_o)} = \mathbb{1}_{(p_o = p_1 p_2)} \sum_{m_o = m_1 + m_2} C_{m_1 m_2 m_o}^{l_1 l_2 l_o} v_{m_1}^{(l_1;p_1)} v_{m_2}^{(l_2;p_2)},$$

where $\mathbb{1}_{(exp)}$ is the indicator function, outputting 1 if the expression exp holds true, or 0 otherwise. Our discussion will be the same no matter we consider $O(n)$ or $SO(n)$. But in and after Section 6, we need to take care of the parity which will play some roles in the further discussion. The CG coefficients are highly sparse, since $C_{m_1 m_2 m_o}^{l_1 l_2 l_o} = 0$ if $m_o \neq m_1 + m_2$. The coefficients are identical to Wigner-3j symbols up to multiplicative factors, so we will adopt Wigner-3j for the implementation. The evaluation of CG coefficients for $SU(2)$ is nearly $\mathcal{O}(1)$ (if we do not consider the few calculations for the closed-form formula, which is cheap enough). To obtain the coefficients of $O(3)$ or $SO(3)$, one often leverages the relation between $SU(2)$ and $SO(3)$: $SU(2)$ is a double cover of $SO(3)$, so the CG matrices of $SU(2)$ and $SO(3)$ are the same up to a change-of-basis, from complex to real. The closed-form formulae for $SU(2)$ CG coefficients are already known (Racah, 1942). Thus, this process can be efficiently evaluated without obvious overhead.

2.3 Tensors and Tensor Product Spaces

To formally define a tensor, we need to consider vector space V and covector space V^*. A tensor product is a bilinear map $\otimes : \mathcal{V} \times \mathcal{V}' \to \mathcal{V} \otimes \mathcal{V}'$. The basis of $\mathcal{V} \otimes \mathcal{V}'$ is the set of $\hat{x}_i \otimes \hat{y}_j \in \mathcal{V} \otimes \mathcal{V}'$, where $\hat{x}_i$ and $\hat{y}_j$ are basis vectors of $\mathcal{V}$ and $\mathcal{V}'$, respectively. That is why one implements outer product as tensor product, as it meets the definition of tensor product. A tensor $T : V^{\otimes r} \otimes V^{* \otimes s} \to F$, where F is some field, is a mathematical object that converts things between different tensor product spaces. For example, the representation $\rho(g)$ is an $(r = 1, s = 1)$-tensor that receives a vector v which is an $(r = 0, s = 1)$-tensor, and output another $(r = 0, s = 1)$-tensor $\rho(g)v$, as r and s vanish correspondingly. Tensor product can help us raise the ranks of tensors, e.g., $V \otimes V \otimes V$ is an $(r = 0, s = 3)$ space. The representation space is accordingly $(r = 3, s = 3)$. Conversely, contraction can reduce the ranks of tensors. We express these operations via some index notations. For example, $T_{i_1 i_2} = T'_{i_1} T''_{i_2}$ is a tensor product, and $T_{i_1} = \sum_{j_1} T'_{i_1 j_1} T''_{j_1}$ is a contraction. The contractions and tensor products are both $O(n)$-equivariant. Obviously, such operations can be mixed together, and we agree that i_k will be placed at the kth index of the output tensor by default. We also agree that j_1 and j_2 vanish due to the contraction. In other words, j_1 and j_2 serve as dummy indices and do not appear in the final expression. As shown in the contraction example above, j_1 is summed over and thus disappears from the left-hand side of the equation. Meanwhile, we will not use Einstein summation convention in this work to keep the conventions of machine learning community, and to avoid ambiguity during derivations. One should note that contractions can be expressed in term of matrix representation. Consider $T_{i_1} = \sum_{j_1, j_2} T'_{i_1 j_1 j_2} T''_{j_1 j_2}$, we have
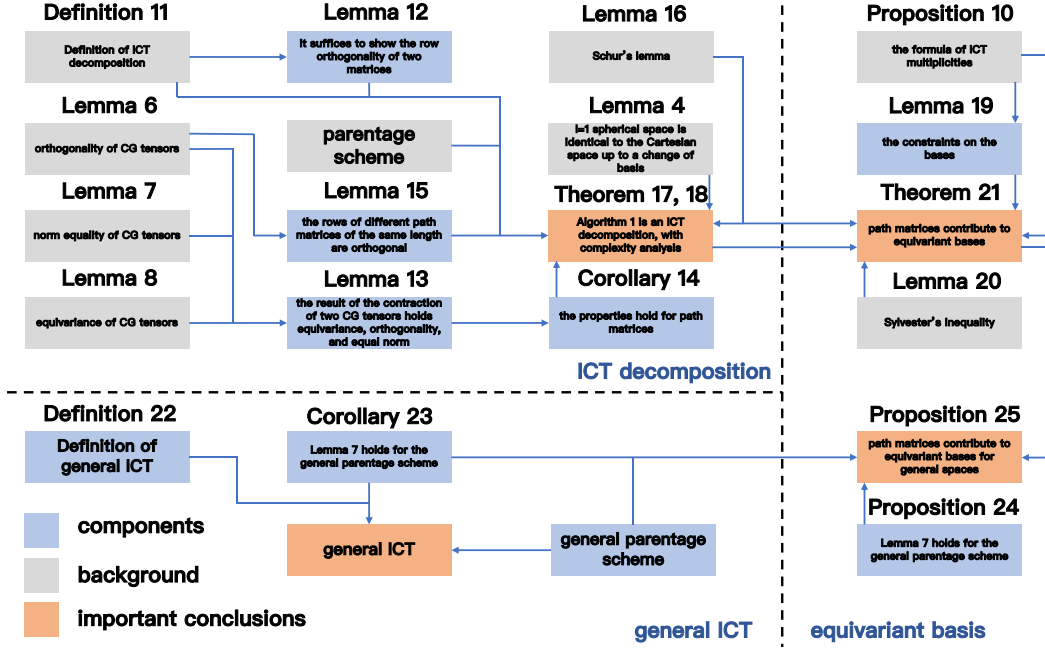
Figure 1: The mind map of our method. The blocks in orange color are aligned with the contributions of this work. The ICT decomposition algorithm (Theorem 17), together with Schur's lemma (Lemma 16), shows that path matrices can also be used to construct an equivariant basis. Building on the fact that our ICT decomposition relies on the parentage scheme, we propose a generalized parentage scheme that allows decomposition in arbitrary spaces and thus yields equivariant bases for those spaces.

$$
\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} T'_{111} & T'_{112} & T'_{121} & T'_{122} \\ T'_{211} & T'_{212} & T'_{221} & T'_{222} \end{bmatrix} \begin{bmatrix} T''_{11} \\ T''_{12} \\ T''_{21} \\ T''_{22} \end{bmatrix}.
$$

It is useful when we need to obtain the inverse of some linear operations. Thus, we sometimes denote $v \in \mathbb{R}^{3^n}$ (instead of $\left(\mathbb{R}^3\right)^{\otimes n}$) as a vector, if the only possible operation is tensor contraction with $M \in \mathbb{R}^{3^n} \times \mathbb{R}^{3^n}$, where M can also be viewed as a matrix. However, by its nature, v is an $(r = 0, s = n)$-tensor, and M is an $(r = n, s = n)$-tensor. For a multi-index tensor $T_{i_1 i_2 i_3}$, sometimes we need to flatten some of the indices. During this operation, we consistently apply a "right-to-left" flattening style, similar to the vectorization vec in Section 1. For example, consider $T \in \mathbb{R}^{3 \times 3 \times 3}$. If we flatten its first two indices, we obtain a
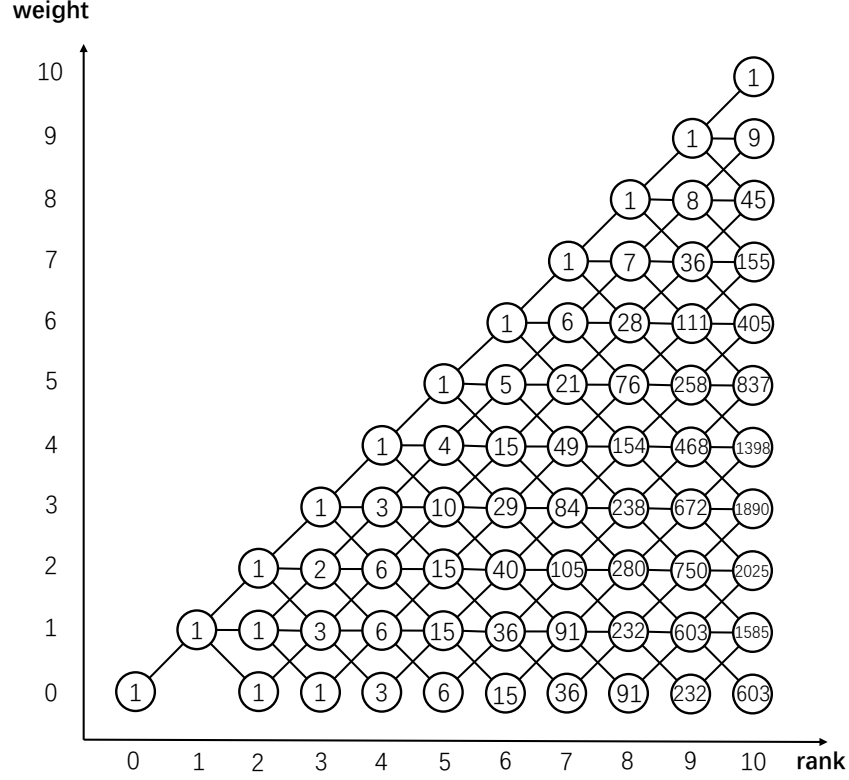
Figure 2: The parentage scheme. For ICTs $\hat{T}^{(n;l;q)}$, the x-axis corresponds to the rank n, the y-axis corresponds to the weight l, and the numbers in the circles represent the multiplicity $N^{(n;l)}$ of index q in equation (4). The line connection indicates that the irreducible representation with the weight in the left column can be decomposed into the irreducible representation with the weight in the right column, when coupling with an $l = 1$ irreducible representation.

matrix,

$$
\begin{pmatrix}
T_{xxx} & T_{xxy} & T_{xxz} \\
T_{xyx} & T_{xyy} & T_{xyz} \\
T_{xzx} & T_{xzy} & T_{xzz} \\
T_{yxx} & T_{yxy} & T_{yxz} \\
T_{yyx} & T_{yyy} & T_{yyz} \\
T_{yzx} & T_{yzy} & T_{yzz} \\
T_{zxx} & T_{zxy} & T_{zxz} \\
T_{zyx} & T_{zyy} & T_{zyz} \\
T_{zzx} & T_{zzy} & T_{zzz}
\end{pmatrix} .
$$

2.4 CG Matrices and the Parentage Scheme

The $O(3)$ group has its representations in tensor product space $(\mathbb{R}^3)^{\otimes n}$, which are some $3^n \times 3^n$ matrices, or equivalently tensor product of n $(r = 1, s = 1)$-tensors, which depends

on one's view. If we interpret them as matrices, then the objects they act on are simply 3^n-dimensional vectors. The tensor product spaces $(\mathbb{R}^3)^{\otimes n}$ for $n > 1$ are generally reducible. Recall CG coefficients, two irreducible spherical vectors v and v' coupling can also be viewed as that they first form a rank 2 tensor, and then decompose into different new spherical vectors. These operations can also be viewed as a product of matrices and vectors. In this context, the CG coefficients $\hat{C}^{l_1 l_2 l_o}$ form a $(2l_1 + 1)(2l_2 + 1) \times (2l_o + 1)$ matrix, which we call a CG matrix. If we reshape it to a $(2l_1 + 1) \times (2l_2 + 1) \times (2l_o + 1)$ tensor, then we instead call it an (l_1, l_2, l_o)-tensor $C^{l_1 l_2 l_o}$. Here are some important properties.

Lemma 6 (Orthogonality) Given an (l_1, l_2, l_o)-CG tensor $C^{l_1 l_2 l_o}$, we have

$$\sum_{j_1, j_2} C^{l_1 l_2 l_o}_{j_1 j_2 k} C^{l_1 l_2 l_o}_{j_1 j_2 k'} = \sum_{j_1, j_2} C^{l_1 l_2 l_o}_{j_1 k j_2} C^{l_1 l_2 l_o}_{j_1 k' j_2} = \sum_{j_1, j_2} C^{l_1 l_2 l_o}_{k j_1 j_2} C^{l_1 l_2 l_o}_{k' j_1 j_2} = 0$$

for $k \neq k'$. More generally, different CG tensors with the same l_1 and l_2, l_1 and l_o, or l_2 and l_o, but differing in other weights, are also orthogonal in this way. Additionally, in such cases, this orthogonality holds true even when $k = k'$.

Lemma 7 (Norm) Given an (l_1, l_2, l_o)-CG tensor $C^{l_1 l_2 l_o}$, we have

$$\sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{j_1 j_2 k})^2 = \sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{j_1 j_2 k'})^2; \quad \sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{j_1 k j_2})^2 = \sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{j_1 k' j_2})^2; \quad \sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{k j_1 j_2})^2 = \sum_{j_1, j_2} (C^{l_1 l_2 l_o}_{k' j_1 j_2})^2$$

for any pair of k and k'.

Lemma 8 (Equivariance) An (l_1, l_2, l_o)-CG tensor $C^{l_1 l_2 l_o}$ maintains the following equivariances,

$$\rho_{l_1 \otimes l_2}(g) \odot C^{l_1 l_2 l_o} = C^{l_1 l_2 l_o} \odot \rho_{l_o}(g),$$

where $\odot$ denotes the tensor contraction. This equation holds for other permutations as Lemmas 6 and 7.

The above lemmas cover the CG matrices case. Since the columns of CG matrices have the same L_2 norm, we denote as $\|\cdot\|_{col}$ for shorthand, and we sometimes call the last index of the CG tensors as the index of their columns. Recall that CG matrices are non-zero if and only if $|l_1 - l_2| \leq l_o \leq l_1 + l_2$. If we start from an $l = 1$ vector and couple an additional $l = 1$ once at a time, then this process has been drawn as a parentage scheme by Coope et al. (1965), as shown in Figure 2. Each time we move from left to the adjacent right column we couple an additional $l = 1$ vector. The parentage scheme also tells us which irreducible representations can a rank n tensor be decomposed into. Note that a rank n tensor can be decomposed into multiple spherical vectors, and the multiplicity for each weight given the rank is written in the scheme. Equivalently, it can also be decomposed into the same numbers of ICTs, but in the Cartesian spaces. We first review the definition of the ICT:

Definition 9 (ICT space) A rank n, weight l, and index q ICT space is a subspace, denoted as $\mathcal{V}^{(n;l;q)} \subseteq (\mathbb{R}^3)^{\otimes n}$, such that for any $T \in \mathcal{V}^{(n;l;q)}$ and $g \in O(3)$, it holds true that

$$\rho_{(\mathbb{R}^3)^{\otimes n}}(g) \odot T \in \mathcal{V}^{(n;l;q)},$$

12

and we also have

$$\dim \mathcal{V}^{(n;l;q)} = 2l + 1.$$

Meanwhile, there is no non-trivial $O(3)$-invariant subspace of $\mathcal{V}^{(n;l;q)}$.

Note that there can be multiple rank n, weight l ICT spaces with different indices q. The tensor product space $\left(\mathbb{R}^3\right)^{\otimes n}$ can be decomposed into multiple ICT spaces, and different ICT spaces only intersect at $\{0\}$. Given a tensor from the tensor product space, the sum of all the ICT decomposition matrices should equal I. We also have a closed-form formula for the multiplicity, which is obtained by predecessors (Mihailov, 1977; Andrews, 2023),

Proposition 10 *The multiplicity of the weight l ICTs of a rank n tensor is*

$$N^{(n;l)} = \sum_{i=0}^{\lfloor (n-l)/3 \rfloor} (-1)^i \frac{n!(2n-3i-l-2)!}{i!(n-i)!(n-2)!(n-3i-l)!}. \tag{4}$$

Note that the number scales exponentially with respect to n, but it is smaller than 3^n, the dimension of the tensor product space.

3. Problem Formulation

We first formulate the task of obtaining ICT decomposition matrices. The properties derived during the process of obtaining these decomposition matrices are further leveraged to construct the bases for equivariant linear mappings between tensor product spaces.

3.1 ICT Decomposition

In this paper, one of our goals is to find an efficient way to obtain a set of matrices $\{H^{(n;l;q)}\}$ to decompose a rank n tensor into ICTs with different weight l and index q, as shown in equation (1). From the previous discussion, we know that ICT decomposition matrices remain invariant regardless of the tensor being decomposed, as the decomposition matrices are constructed with respect to the tensor product spaces. Each decomposition matrix should be able to map a tensor from the tensor product space to its corresponding ICT subspace with rank n, weight l, and index q. Given the definition of ICT space (Definition 9), we precisely define what an ICT decomposition is. In other words, we specify the properties that the matrices H must satisfy:

Definition 11 (ICT decomposition) *Given a Cartesian tensor product space $(\mathbb{R}^3)^{\otimes n} \cong \mathbb{R}^{3^n}$ formed by tensor product of Cartesian spaces $\mathbb{R}^3$, an ICT decomposition is a set of decomposition matrices $H^{(n;l;q)} \in \mathbb{R}^{3^n \times 3^n}$, where l is the weight of ICTs, and q is the index of the multiplicities, such that:*

1) $H^{(n;l;q)}$ are $O(3)$-$((\mathbb{R}^3)^{\otimes n}, (\mathbb{R}^3)^{\otimes n})$-equivariant.

2) The summation of the decomposition matrices is an identity matrix, i.e., $\sum_{l,q} H^{(n;l;q)} = I$.

3) The rank (matrix rank) of the decomposition matrix $H^{(n;l;q)}$ is $2l + 1$.

4) $\{H^{(n;l;q)}v \mid v \in \mathbb{R}^{3^n}\}$ has no non-trivial $O(3)$-invariant subspace.

5) $\{H^{(n;l;q)}v \mid v \in \mathbb{R}^{3^n}\} \bigcap \{H^{(n;l';q')}v \mid v \in \mathbb{R}^{3^n}\} = \{0\}$ holds for $l \neq l'$ or $q \neq q'$.

Furthermore, if we have

$$\left(H^{(n;l;q)}v\right) \odot \left(H^{(n;l';q')}v'\right) = 0 \tag{5}$$

for any $v, v' \in \mathbb{R}^{3^n}$, and $l \neq l'$ or $q \neq q'$, then we further call it an orthogonal ICT decomposition, and the matrices orthogonal decomposition matrices.

Consider condition 1), for any $g \in O(3)$ and $v \in R^{3^n}$, we must have

$$\rho_{\mathbb{R}^{3n}}(g)H^{(n;l;q)}v \in \{H^{(n;l;q)}v \mid v \in R^{3^n}\},$$

as

$$\rho_{\mathbb{R}^{3n}}(g)H^{(n;l;q)}v = H^{(n;l;q)}\rho_{\mathbb{R}^{3n}}(g)v,$$

where $\rho_{\mathbb{R}^{3n}}(g)v \in R^{3^n}$. Thus, condition 1) makes $\{H^{(n;l;q)}v \mid v \in R^{3^n}\}$ an $O(3)$-invariant subspace. Meanwhile, condition 3) can also guarantee that

$$\dim \{H^{(n;l;q)}v \mid v \in R^{3^n}\} = 2l + 1,$$

and therefore $\{H^{(n;l;q)}v \mid v \in R^{3^n}\}$ is a rank n, weight l, and index q ICT space. Note that ICT decomposition is not necessary to be orthogonal, but being orthogonal is the ultimate goal for ICT decomposition. From now on, the proofs of supportive lemmas are shown in the Appendix A to save space and maintain a clear flow. To prove the orthogonality, we have

Lemma 12 Given matrices $M : \mathcal{V} \to \mathcal{V}'$ and $M' : \mathcal{V} \to \mathcal{V}'$ of the same shape, if $M^\top M' = 0$, then

$$(Mv) \odot (M'v') = 0 \tag{6}$$

for any $v, v' \in \mathcal{V}$.

This work obtains orthogonal ICT decomposition matrices for tensor product spaces with rank $6 \leq n \leq 9$. A comparison with predecessors is shown in Table 1. Theoretically, our method can obtain ICT decomposition matrices with arbitrary ranks analytically. While the full set of $n = 10$ ICT decomposition matrices can be computed in a reasonable amount of time, storing such a massive number of matrices poses significant challenges due to the required storage space. However, in practical implementation, it is possible to obtain a subset of ICT decomposition matrices with desired weights, rather than generating the entire set of matrices.

3.2 Basis of the Equivariant Design Space

Given a tensor $T \in (\mathbb{R}^3)^{\otimes n}$, a critical challenge in the design of EGNNs is determining which linear operations acting on the tensor are equivariant, i.e., find the set of f in equation (2), denoted as $\text{End}_{O(3)}((\mathbb{R}^3)^{\otimes n})$. In real implementation, we flatten T to vectors v, and the linear mappings f are therefore matrices. Pearce-Crump (2023b) leverages Brauer (1937)'s

theorem to obtain spanning sets for homomorphism spaces. However, the spanning set becomes increasingly redundant as the rank of tensors rises. In this work, we aim to directly find orthogonal bases, instead of spanning sets, for equivariant spaces.

Additionally, we seek to find bases for the equivariant spaces $\mathrm{Hom}_{O(3)}(\mathcal{V}, \mathcal{V}')$, where $\mathcal{V}$ is not necessarily equal to $\mathcal{V}'$. Here, $\mathcal{V}$ and $\mathcal{V}'$ can be arbitrary direct sums of tensor product spherical spaces (which also include Cartesian spaces, recall Lemma 4). This approach enables us to design EGNNs more freely by making the domain and co-domain of equivariant layers steerable. In other words, given an arbitrary domain and its co-domain, one can efficiently construct equivariant layers without redundant parameters. A discussion of the relationship between Pearce-Crump (2023b) and this work is provided in Section 8.1.

4. High-Rank Orthogonal ICT Decomposition

In this section, we aim to develop a method to efficiently obtain the ICT decomposition matrices, i.e., to address the challenge outlined in Section 3.1. In Section 4.1, we first introduce the theoretical foundation, along with proofs, which form the basis of our method. Then, in Section 4.2, we provide an example to demonstrate how our method works.

4.1 The Theory and Method

*According to the previous discussion, given a vector $v \in \mathbb{R}^{3^n}$ (flattened from the tensor), we can decompose it into multiple irreducible vectors $v^{(1)}, \ldots, v^{(k)}$. Under the Cartesian basis, the vector v equals to the **sum** of the irreducible vectors, as*

$$v = \sum_i v^{(i)}.$$

However, the things are easier when we consider under the spherical basis. The above relation becomes

$$v = \mathrm{concat}(v^{(1)}, \ldots, v^{(k)}),$$

*i.e., **concatenation** of vectors. It is easy to select out the desired irreducible vectors in this scenario. Thus, we first convert the Cartesian space to the direct sum of the spherical spaces, then, choose the desired irreducible representation, and finally convert it back to the Cartesian tensor product space. Therefore, the key here is to find a change-of-basis matrix between these two spaces. But how to construct it? Inspired by the fact that CG matrix is an equivariant mapping from the tensor product of two spaces to a single space, we use contraction to connect multiple CG matrices and tensor. This yields an equivariant mapping from tensor product of many spaces to a single space, which we call a path matrix. Then, a concatenation of path matrices becomes an equivariant mapping from tensor product of spaces to the direct sum of spaces. In the following we will introduce how we construct these path matrices in detail, and why they exhibit the properties we want. We first obtain the following lemmas:*

Lemma 13 *Given an $(l_x \otimes l_y, l_2)$-CG matrix $\hat{C}^{l_x l_y l_2}$ and an (l_1, l_2, l_3)-CG tensor $C^{l_1 l_2 l_3}$, if we let*
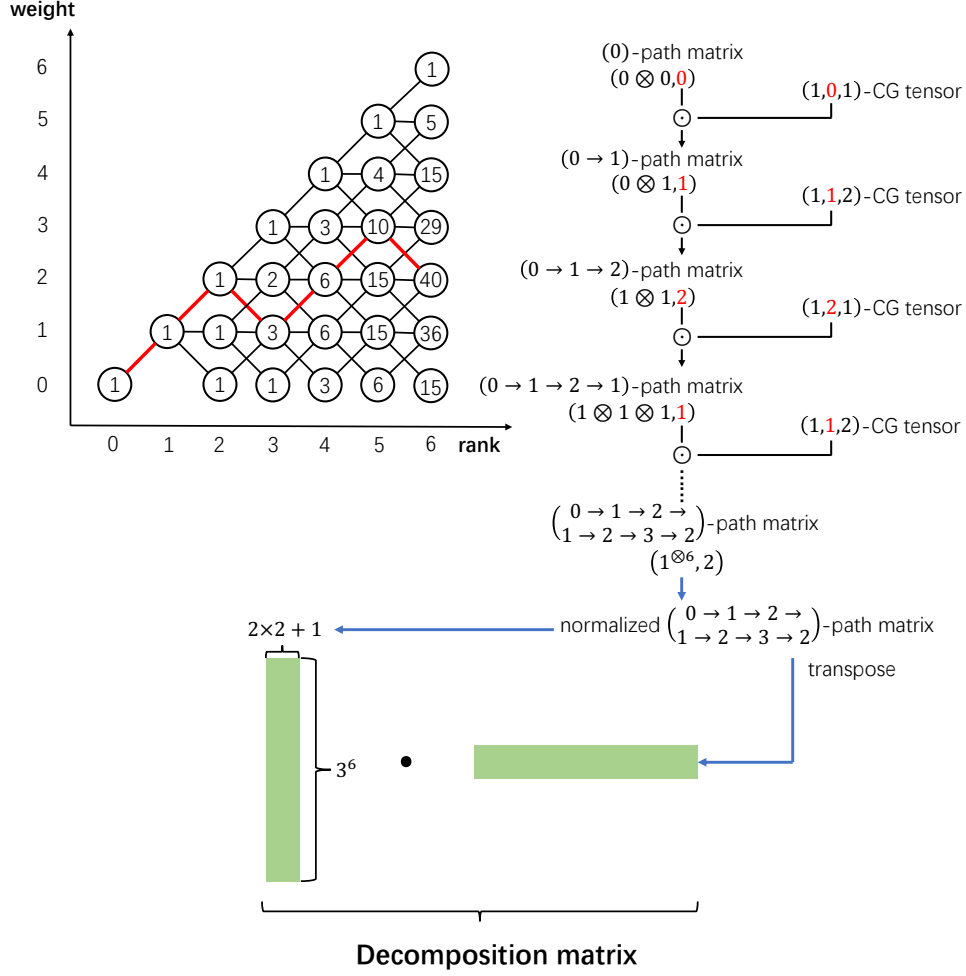
Figure 3: Construction of decomposition matrix. **Top:** The path matrices are formed via sequential contractions according to the paths in the parentage scheme. **Bottom:** The decomposition matrix is obtained as the product of the path matrix and its transpose.

$$T_{i_1 i_2 i_3} = \sum_{j_1} \hat{C}_{i_2 j_1}^{l_x l_y l_2} C_{i_1 j_1 i_3}^{l_1 l_2 l_3} \tag{7}$$

and we flatten the first two indices of T, converting it into a $(2l_1+1)(2l_x+1)(2l_y+1) \times (2l_3+1)$ matrix. Then the following statements hold true:

1) T is $O(3)$-$(l_1 \otimes l_x \otimes l_y, l_3)$-equivariant.

2) The columns of T are orthogonal.

3) The columns of T have the same L_2-norms.

Here is the strategy for obtaining the path matrices, we first let a $(0 \otimes 0, 0)$-CG matrix to contract with a $(1, 0, 1)$-CG tensor to obtain a $(0 \otimes 1, 1)$-matrix. This matrix can further

contract with $(1, 1, 2)$, $(1, 1, 1)$, *and* $(1, 1, 0)$-*CG tensors to obtain the associated matrix. For example, we call the obtained* $(0 \otimes 1 \otimes 1, 2)$ *a* $(0 \rightarrow 1 \rightarrow 2)$-*path matrix* $P^{(0 \rightarrow 1 \rightarrow 2)}$. *The path matrices are constructed via the parentage scheme, which will decide what is the next CG tensor we are going to contract with, as shown in Figure 3. A rigorous description and the related properties are:*

Corollary 14 *Consider a path matrix* $P^{(0 \rightarrow l_1 \rightarrow \cdots \rightarrow l_k)}$, *with* $l_1 = 1$, *and* $|l_{i-1} - 1| \leq l_i \leq l_{i-1} + 1$, *for* $2 \leq i \leq k$. *This path matrix is iteratively constructed in the following way,*

$$P_{i_1 i_2 i_3}^{(0 \rightarrow \cdots \rightarrow l_u \rightarrow l_t)} = \sum_{j_1} P_{i_2 j_1}^{(0 \rightarrow \cdots \rightarrow l_u)} C_{i_1 j_1 i_3}^{1, l_u l_t},$$

with the initial path matrix $P^{(0)}$ *being a* $(0 \otimes 0, 0)$-*CG matrix, where* $C^{1, l_2 l_3}$ *is a CG tensor. After this contraction, we flatten the first two indices of* $P^{(0 \rightarrow \cdots \rightarrow l_u \rightarrow l_t)}$. *Then, the following properties hold,*

    1) $P^{(0 \rightarrow l_1 \rightarrow \cdots \rightarrow l_k)}$ *is* $O(3)$-$((l = 1)^{\otimes k}, l_k)$-*equivariant.*

    2) The columns of $P^{(0 \rightarrow l_1 \rightarrow \cdots \rightarrow l_k)}$ *are orthogonal.*

    3) The columns of $P^{(0 \rightarrow l_1 \rightarrow \cdots \rightarrow l_k)}$ *have the same* L_2-*norms.*

Proof *The path matrices have all the properties we used in Lemma 13. We prove this by a simple deduction. From the above lemma we know that* $(0 \rightarrow 1)$-*path matrix holds all the properties. This gives us the initial condition. Assume that a matrix holds the properties, Lemma 13 tells us the next obtained path matrix also shares them.* ∎

    Next, we prove that

Lemma 15 *The columns of different path matrices of the same path length are orthogonal. Specifically, given two path matrices* $P^{(p)}$ *and* $P^{(p')}$ *with different paths* p *and* p', *it holds that*

$$\sum_j P_{j i_1}^{(p)} P_{j i_2}^{(p')} = 0.$$

    Meanwhile, Schur's lemma is also required for the proof of the ICT decomposition.

Lemma 16 (Schur's lemma) *Given finite-dimensional irreducible representations* $\rho_{\mathcal{V}} : G \rightarrow \mathcal{V}$ *and* $\rho_{\mathcal{V}'} : G \rightarrow \mathcal{V}'$ *of group* G, *and a matrix* $M : \mathcal{V} \rightarrow \mathcal{V}'$, *suppose we have that*

$$\rho_{\mathcal{V}'}(g) M = M \rho_{\mathcal{V}}(g) \tag{8}$$

holds for any $g \in G$, *then*

    1) $M = 0$ *if* $\mathcal{V}$ *and* $\mathcal{V}'$ *are not equivalent, i.e., there is no such matrix* Q *that*

$$\rho_{\mathcal{V}}(g) = Q \rho_{\mathcal{V}'}(g) Q^{-1} \tag{9}$$

holds for any $g \in G$.

    2) $M = cI$, *where* c *is a constant, if* $\mathcal{V} = \mathcal{V}'$.

    Till now, it is adequate to show our solution to the problem:

Algorithm 1 Decomposition Matrices Generation

Input: Target rank n;

Step 1. Initialize (0)-path matrices $P^{(p_0)} = C^{000}$, and flatten the first two indices. Next, maintain the path $p_0 = (0)$, the current column number of the parentage scheme $i = 0$, and the current weight $l_0 = 0$.

Step 2. For those available weights in the $(i+1)$-th column, i.e., weights $|l_i - 1| \leq l_{i+1} \leq l_i + 1$, we update the path by adding l_{i+1} to the end of path p_i to obtain p_{i+1}, and calculate the contraction,

$$P^{(p_{i+1})} = \text{flatten}_2(P^{(p_i)} \odot C^{1l_i l_{i+1}}),$$

where flatten_2 denotes flattening the first two indices. Then, we update the column number $i \leftarrow i + 1$.

Step 3. Repeat **Step 2** until $i = n$. For each obtained path matrices $P^{(p)}$, we normalize the columns by

$$\hat{P}^{(p)}_{*,c} = \frac{P^{(p)}_{*,c}}{\|P^{(p)}_{*,c}\|_2}.$$

Then, we multiply each path matrix with its transpose,

$$H^{(n;l;q)} = \hat{P}^{(p)} \cdot (\hat{P}^{(p)})^\top.$$

Output: The list of all $H^{(n;l;q)}$.

Theorem 17 *As in Algorithm 1, given the set of all the paths $\{p_i\}$ of the same length $n+1$ with $\#\{p_i\} = K$ and path matrices $\{P^{(p_i)}\}$, the set of $\left\{ \left(\|P^{(p_i)}\|_{col}\right)^{-2} P^{(p_i)} \left(P^{(p_i)}\right)^\top \right\}$ is a rank n orthogonal ICT decomposition.*

Proof *From Corollary 14, we know that each $P^{(p_i)}$ has full rank (but not a square matrix), and has the same L_2 norm of each column, $\|P^{(p_i)}\|_{col}$. Subsequently, normalized path matrices $\hat{P}^{(p_i)} = \left(\|P^{(p_i)}\|_{col}\right)^{-1} P^{(p_i)}$ are with a unit L_2 norm, as $\|\left(\|P^{(p_i)}\|_{col}\right)^{-1} P^{(p_i)}\|_{col} = 1$. This does not affect the equivariance of the path matrices, as the L_2-norm is invariant. We concatenate all the normalized path matrices $\hat{P}^{(p_i)}$ along the last dimension,*

$$D^{(l=1)^{\otimes n} \to \text{irr}((l=1)^{\otimes n})} = \left[\left(\|P^{(p_1)}\|_{col}\right)^{-1} P^{(p_1)} \quad \cdots \quad \left(\|P^{(p_K)}\|_{col}\right)^{-1} P^{(p_K)} \right] \tag{10}$$

where irr outputs all the possible spherical irreducible representation spaces. The above equation forms a $3^n \times 3^n$ matrix. In this proof, we will use D to represent $D^{(l=1)^{\otimes n} \to \text{irr}((l=1)^{\otimes n})}$ to save space. It is an orthogonal matrix: $D^\top D = DD^\top = I$, as its columns are all orthogonal and normalized to unit L_2 norm as we discussed above. From Corollary 14, we know that $D^\top$ is $O(3)$-$\left((l=1)^{\otimes n}, \oplus_i^K \left(l^{(i)}\right)\right)$-equivariant, where $l^{(i)} \in \text{irr}((l=1)^{\otimes n})$. If we left multiply it with a selection matrix (a diagonal matrix with ones and zeros on the diagonal) $I^{(j)}$ to select out the jth irreducible spherical representation, also defined as the last weight of the path p_j, the resulting $I^{(j)} D^\top$ is $O(3)$-$\left((l=1)^{\otimes n}, l^{(j)}\right)$-equivariant. Similarly, the inverse D is $O(3)$-$\left(\oplus_i^K \left(l^{(i)}\right), (l=1)^{\otimes n}\right)$-equivariant. With the right multiplied $I^{(j)}$, it becomes $O(3)$-$\left(l^{(j)}, (l=1)^{\otimes n}\right)$-equivariant. Together, $DI^{(j)} D^\top$ is $O(3)$-$((l=1)^{\otimes n}, (l=1)^{\otimes n})$-equivariant.

18

From Lemma 4, this relation is equivalent to $O(3)$-$\left((\mathbb{R}^3)^{\otimes n}, (\mathbb{R}^3)^{\otimes n}\right)$-equivariance in the n-rank Cartesian tensor product space. Meanwhile,

$$DI^{(j)}D^\top = \hat{P}^{(p_j)}(\hat{P}^{(p_j)})^\top, \tag{11}$$

which follows from the fact that D is an orthogonal matrix, and the concatenation of $\hat{P}$ forms D. It therefore meets condition 1) of Definition 11.

Next we prove 2), it holds true because

$$\sum_j \hat{P}^{(p_j)}\left(\hat{P}^{(p_j)}\right)^\top = \sum_j DI^{(j)}D^\top = D\left(\sum_j I^{(j)}\right)D^\top = I. \tag{12}$$

Then, for 3), we prove that $\hat{P}^{(p_j)}\left(\hat{P}^{(p_j)}\right)^\top$ has matrix rank $2l' + 1$, where l' is the last weight of the path p_j. It is easy to prove, as

$$\text{Rank}\left(\hat{P}^{(p_j)}\left(\hat{P}^{(p_j)}\right)^\top\right) = \text{Rank}\left(\hat{P}^{(p_j)}\right) = \text{Rank}\left(\left(\hat{P}^{(p_j)}\right)^\top\right) = 2l' + 1. \tag{13}$$

To prove 4), we first note that the tensor product space is a direct sum of multiple irreducible representation spaces, as

$$\left(\mathbb{R}^3\right)^{\otimes n} = \oplus_i^K \hat{\mathcal{V}}^{\left((\mathbb{R}^3)^{\otimes n};i\right)}, \tag{14}$$

where $\hat{\mathcal{V}}^{(\mathcal{V};i)}$ denotes the ith irreducible representation, residing in the $\mathcal{V}$ space. Meanwhile, note that $\left(\hat{P}^{(p_j)}\right)^\top$ maps from $\left(\mathbb{R}^3\right)^{\otimes n}$ to $l^{(j)}$. According to Lemma 16 (Schur's lemma), $\left(\hat{P}^{(p_j)}\right)^\top$ is also a non-zero mapping from $\hat{\mathcal{V}}^{\left((\mathbb{R}^3)^{\otimes n};i\right)}$ to $l^{(j)}$, for some i. Conversely, the same reasoning tells us that $\hat{P}^{(p_j)}$ is a mapping from $l^{(j)}$ to $\hat{\mathcal{V}}^{\left((\mathbb{R}^3)^{\otimes n};i\right)}$. Put them together, $\hat{P}^{(p_j)}\left(\hat{P}^{(p_j)}\right)^\top$ is an equivariant mapping from $\hat{\mathcal{V}}^{\left((\mathbb{R}^3)^{\otimes n};i\right)}$ to itself. Since $\hat{\mathcal{V}}^{\left((\mathbb{R}^3)^{\otimes n};i\right)}$ is an irreducible representation space, there is no non-trivial $O(3)$-invariant subspace of the output space, as the input space does. This concludes the proof for 4).

Finally, we prove the orthogonality, given $\hat{P}^{(p_k)}(\hat{P}^{(p_k)})^\top$ and $\hat{P}^{(p_{k'})}(\hat{P}^{(p_{k'})})^\top$ with different $k \neq k'$, consider dot product of an arbitrary pair of their columns $c \neq c'$, we have

$$\left(\hat{P}^{(p_k)}\left(\hat{P}^{(p_k)}\right)^\top\right)_{*,c} \odot \left(\hat{P}^{(p_{k'})}\left(\hat{P}^{(p_{k'})}\right)^\top\right)_{*,c'}$$

$$= \sum_{j_1}\left(\sum_{j_2} \hat{P}^{(p_k)}_{j_1 j_2}\hat{P}^{(p_k)}_{c j_2}\right)\left(\sum_{j_2'} \hat{P}^{(p_{k'})}_{j_1 j_2'}\hat{P}^{(p_{k'})}_{c' j_2'}\right)$$

$$= \sum_{j_2, j_2'} \hat{P}^{(p_k)}_{c j_2}\hat{P}^{(p_{k'})}_{c' j_2'}\underbrace{\left(\sum_{j_1} \hat{P}^{(p_k)}_{j_1 j_2}\hat{P}^{(p_{k'})}_{j_1 j_2'}\right)}_{0}$$

$$=0. \tag{15}$$

The last equality follows from Lemma 15. From Lemma 12, it satisfies the orthogonality requirement of Definition 11, which is stronger than condition 5), so 5) is implicitly guaranteed. ∎

The length n of the path in the above theorem includes 0 as the first waypoint of the path. Here, we discuss why our work obtains orthogonality, while the predecessors' works (i.e., Coope et al. (1965); Andrews and Ghoul (1982); Bonvicini (2024)) do not. Given a rank-n tensor T to decompose, they first convert it to the weight-l natural tensor space by the mapping tensor $\tilde{G}$. Then, another mapping tensor G pulls the tensor back to the original rank-n tensor space. Since $\tilde{G}$ filters out irreducible representations other than the one of weight l and some index q, the final space is of rank n, weight l, and index q. $\tilde{G}$ and G are calculated from natural projection tensors and FICTs, which are, in fact, multiple δs and Levi-Civita ϵ (can be seen as rank-2 and rank-3 tensors, respectively). There are different permutation strategies of these δs and ϵ's. We only need to pick up those linearly independent components, which does not guarantee that they are orthogonal. We can make G and $\tilde{G}$ into matrices, and the ICT decomposition matrices are then $G\tilde{G}$. Since the above orthogonality does not hold, different G or $\tilde{G}$ are not orthogonal either. Now, look back to our work, the ICT decomposition matrices are $\hat{P}(\hat{P})^{\top}$, where $(\hat{P})^{\top}$ converts the tensor to the spherical irreducible space, and $\hat{P}$ pulls back the tensor to the original space. The orthogonality of different ICT decomposition matrices comes from the column orthogonality of $\hat{P}$, which further comes from the orthogonality of CG tensor. This is the key reason why our work guarantees orthogonality while the predecessors works do not. Next, we show the complexity of our algorithm for obtaining ICT decomposition matrices.

Theorem 18 *Obtaining the ICT decomposition matrices is of exponential time and space complexity, in terms of rank n.*

Proof *For time complexity, note that Parentage Scheme is faster than fully traversing a ternary tree, which consists of 3^n operations. Next, we consider the algebraic complexity. Suppose that we are in the i-th column of Parentage Scheme. For each weight l_i, suppose that we maintain a path p_i, and there are at most three valid contractions, $P^{(p_i)}$ contracts with $C^{(1l_i(l_i-1))}$, $C^{(1l_i(l_i))}$, and $C^{(1l_i(l_i+1))}$, respectively. Since $P^{(p_i)}$ has shape $3^i \times (2l_i+1)$, the number of operations of the above contractions are $3^i \times (2l_i+1) \times 3 \times (2l_i-1)$, $3^i \times (2l_i+1) \times 3 \times (2l_i+1)$, and $3^i \times (2l_i+1) \times 3 \times (2l_i+3)$, respectively. Note that $2l_i+3 \leq 2i+3$, thus the total of the above three contractions occupy no more than $(2i+3)^2 3^{i+2}$ operations. The contractions will happen for at most i times, so this layer consists of at most $(2i+3)^3 3^{i+2}$ operations. There are n layers, and for each layer, the number of operations is no more than $(2n+3)^3 3^{n+2}$. Thus, all the n layers contribute to at most $(2n+3)^4 3^{n+2}$ operations. Lastly, we consider the multiplication between path matrices. There are at most 3^{2n} pairs of path matrices, each of which holds at most $(2n+1)3^{2n}$ operations. Thus, there are at most $(2n+1)3^{4n}$ operations. Together with the previous path matrices generation, there are in total at most $(2n+1)^4(3^{n+2} + 3^{4n})$ operations, which holds $\mathcal{O}(n^4 3^{4n})$ complexity. For space complexity, the largest matrices we maintain are the set of $\hat{P}^{(p)}$ for possible all paths p, of*

Rank	5	6	7	8	9
Time	<0.1s	1s	3s	11s	4m32s

Table 3: Benchmarking decomposition matrices generation speed.

which the total size is $3^{n-1} \times 3^{n-1}$. ∎

It is also noteworthy that the complexity cannot be lower than exponential in terms of rank n, as the ICT decomposition matrices themselves have an inherently exponential structure.

4.2 An Example of Constructing Path Matrices

Here, we describe the process of constructing decomposition matrices using the parentage scheme. A path in the scheme uniquely corresponds to a path matrix. We start by creating a temporary path matrix, denoted as $(0 \otimes 0, 0)$, which will be updated in each round. This matrix is reshaped from a $(0, 0, 0)$ CG tensor (merging the first two dimensions). The starting point is $(0, 0)$ in the parentage scheme, where the first coordinate represents the rank, and the second represents the weight.

Next, we move from rank 0 to rank 1, i.e., from the first column to the second column. Observe that the only line connecting $(0, 0)$ and $(1, *)$ is the one between $(0, 0)$ and $(1, 1)$. Therefore, the CG tensor to be contracted is $(1, 0, 1)$. The 1 at the third place is determined by the second 1 in $(1, 1)$ in the parentage scheme. The 0 at the second place is determined by the fact that our currently maintained temporary path matrix has 0 in the last position of $(0 \otimes 0, 0)$. The first 1 is fixed and remains unchanged regardless of which lines we are moving along (this behavior will change for the general parentage scheme, as introduced in Section 6).

The contraction is performed between the last dimension of the temporary path matrix and the second dimension of the CG tensor, as shown in equation (7). By merging the first two dimensions of the resulting tensor, we obtain a $(0 \rightarrow 1)$ path matrix, which can be represented as $(0 \otimes 0 \otimes 1, 1)$ (we sometimes omit the $0 \otimes 0$ for notational simplicity).

At this point, we are positioned at $(1, 1)$. The connected points in the next column are $(2, 0)$, $(2, 1)$, and $(2, 2)$. Consequently, the CG tensors to be contracted are $(1, 1, 0)$, $(1, 1, 1)$, and $(1, 1, 2)$, respectively. These contractions yield initial path matrices $(0 \rightarrow 1 \rightarrow 2)$, $(0 \rightarrow 1 \rightarrow 1)$, and $(0 \rightarrow 1 \rightarrow 0)$ with shapes $(1 \otimes 1, 2)$, $(1 \otimes 1, 1)$, and $(1 \otimes 1, 0)$, respectively. This process is repeated iteratively until we reach rank n. The process and an example are illustrated in Figure 3. A detailed example for generating rank-2 ICT decomposition matrices are shown in Appendix C.

We present the process in Algorithm 1. We also benchmark the speed of our algorithm for obtaining the set of all decomposition matrices for different ranks, as shown in Table 3. We run the experiments on 28-cores Intel® Xeon® Gold 6330 CPU @ 2.00GHz.

21

5. Basis of the Equivariant Design Space

Section 3.2 implies that one of our goals is to find the basis of the equivariant design space $\{M \in \mathbb{R}^{3^n} \times \mathbb{R}^{3^n} \mid \rho(g)M = M\rho(g)\}$, where M are some linear mappings. Recall Lemma 16 (Schur's lemma), for an endomorphism space from a single finite-dimensional irreducible representation space to itself, the basis is simply an identity matrix I and it is a single dimension space. Note that a rank-n Cartesian tensor is a direct sum of several such irreducible representation spaces. In such situation, we obtain the following lemma:

Lemma 19 *Given a representation $\mathcal{V} = \oplus_i^n \mathcal{V}^{(i)}$ that can be written as the direct sum of several finite-dimensional irreducible representations $\mathcal{V}^{(i)}$. If they are isomorphic to each other, i.e.,*

$$\mathcal{V}^{(0)} \cong \ldots \cong \mathcal{V}^{(n)},$$

then the number of dimensions of the equivariant design space is n^2, and the basis consists of one-dimensional linear projections with a scaling factor from $\mathcal{V}^{(i)}$ to $\mathcal{V}^{(j)}$ for any i and j.

Proof *First note that*

$$(\mathcal{V}^{(i)} \oplus \mathcal{V}^{(j)}) \to \mathcal{V}' \cong (\mathcal{V}^{(i)} \to \mathcal{V}') + (\mathcal{V}^{(j)} \to \mathcal{V}'), \tag{16}$$

where $+$ represents linear combinations in the same codomain, and conversely

$$\mathcal{V}' \to (\mathcal{V}^{(i)} \oplus \mathcal{V}^{(j)}) \cong (S' \to \mathcal{V}^{(i)}) \oplus (\mathcal{V}' \to \mathcal{V}^{(j)}). \tag{17}$$

Given a linear projection $f : \oplus_i^n \mathcal{V}^{(i)} \to \oplus_i^n \mathcal{V}^{(i)}$, we have

$$\oplus_i^n \mathcal{V}^{(i)} \to \oplus_i^n \mathcal{V}^{(i)}$$
$$\cong \oplus_j^n \left(\oplus_i^n \mathcal{V}^{(i)} \to \mathcal{V}^{(j)} \right)$$
$$\cong \oplus_j^n \left(+_i^n \left(\mathcal{V}^{(i)} \to \mathcal{V}^{(j)} \right) \right). \tag{18}$$

Note that the mapping spaces $(\mathcal{V}^{(i_1)} \to \mathcal{V}^{(j)})$ and $(\mathcal{V}^{(i_2)} \to \mathcal{V}^{(j)})$ form a basis for the sum $(\mathcal{V}^{(i_1)} \to \mathcal{V}^{(j)}) + (\mathcal{V}^{(i_2)} \to \mathcal{V}^{(j)})$ when $i_1 \neq i_2$. According to the definition of a direct sum, the union of the basis elements of spaces $\mathcal{A}$ and $\mathcal{B}$ must form a basis of $\mathcal{A} \oplus \mathcal{B}$. By Lemma 16, each such mapping space is one-dimensional. Therefore, the mapping from $\oplus_{i=1}^n \mathcal{V}^{(i)}$ to itself, denoted as $\oplus_{i=1}^n \mathcal{V}^{(i)} \to \oplus_{i=1}^n \mathcal{V}^{(i)}$, has a total of n^2 basis elements of the form $\mathcal{V}^{(i)} \to \mathcal{V}^{(j)}$. ∎

This lemma hints us that the keypoint of the basis construction is to find a set of matrices that map between different irreducible Cartesian tensors with the same weight. Similarly to the ICT decomposition, it is easy to find such mappings in spherical spaces rather than the Cartesian ones. Once we find such mappings, we can also use change-of-basis matrices to pull us back to the Cartesian space. Meanwhile, proper construction and the orthogonality of the path matrices could help us find an orthogonal basis.

Before formally proving the following construction is an orthogonal basis, we first introduce the idea of the construction. From the perspective of matrices, it is equivalent to say that the matrices which can scale an irreducible representation and add it to another (including itself) form a basis for the equivariant design space. Here, we provide an example.

$$
\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\
\cdots & 0 & 0 & 0 & k & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & k & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & k & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}_{A^{(p_i \to p_j)}}
\begin{bmatrix}
\vdots \\
v_{-1}^{(p_j)} \\
v_0^{(p_j)} \\
v_1^{(p_j)} \\
v_{-1}^{(p_i)} \\
v_0^{(p_i)} \\
v_1^{(p_i)} \\
\vdots
\end{bmatrix}
=
\begin{bmatrix}
\vdots \\
kv_{-1}^{(p_i)} \\
kv_0^{(p_i)} \\
kv_1^{(p_i)} \\
0 \\
0 \\
0 \\
\vdots
\end{bmatrix},
\tag{19}
$$

where $A^{(p_i \to p_j)}$ adds $kv^{(p_i)}$ to the position of $v^{(p_j)}$, where $v^{(p_i)}$ belongs to the spherical space mapped by the p_i path matrix, and k is a scalar. Note that $A^{(p_i \to p_j)}$ resides in the spherical space and is an $(r = 1, s = 1)$-tensor. To construct a matrix that performs the adding operation in the irreducible Cartesian space, we need to convert it back to Cartesian space via the change-of-basis matrices D and $D^{-1} = D^\top$. Since it is an $(r = 1, s = 1)$-tensor, we need two change-of-basis matrices to multiply with it to convert both the vector component and the covector component, where D This gives us

$$
\begin{bmatrix}
\vdots \\
(\hat{P}_{*,0}^{(p_j)})^\top \\
(\hat{P}_{*,1}^{(p_j)})^\top \\
(\hat{P}_{*,2}^{(p_j)})^\top \\
(\hat{P}_{*,0}^{(p_i)})^\top \\
(\hat{P}_{*,1}^{(p_i)})^\top \\
(\hat{P}_{*,2}^{(p_i)})^\top \\
\vdots
\end{bmatrix}^\top_{D}
\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\
\cdots & 0 & 0 & 0 & k & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & k & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & k & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}_{A^{(p_i \to p_j)}}
\begin{bmatrix}
\vdots \\
(\hat{P}_{*,0}^{(p_j)})^\top \\
(\hat{P}_{*,1}^{(p_j)})^\top \\
(\hat{P}_{*,2}^{(p_j)})^\top \\
(\hat{P}_{*,0}^{(p_i)})^\top \\
(\hat{P}_{*,1}^{(p_i)})^\top \\
(\hat{P}_{*,2}^{(p_i)})^\top \\
\vdots
\end{bmatrix}_{D^\top}
$$

$$
=
\begin{bmatrix}
\vdots \\
(\hat{P}_{*,0}^{(p_j)})^\top \\
(\hat{P}_{*,1}^{(p_j)})^\top \\
(\hat{P}_{*,2}^{(p_j)})^\top \\
(\hat{P}_{*,0}^{(p_i)})^\top \\
(\hat{P}_{*,1}^{(p_i)})^\top \\
(\hat{P}_{*,2}^{(p_i)})^\top \\
\vdots
\end{bmatrix}^\top
\begin{bmatrix}
\vdots \\
k(\hat{P}_{*,0}^{(p_i)})^\top \\
k(\hat{P}_{*,1}^{(p_i)})^\top \\
k(\hat{P}_{*,2}^{(p_i)})^\top \\
0 \\
0 \\
0 \\
\vdots
\end{bmatrix}
$$

$$
= k \sum_j \hat{P}_{i_1 j}^{(p_j)} \left(\hat{P}^{(p_i)} \right)^\top_{i_2 j}.
\tag{20}
$$

Thus, we can reduce the multiplication of three square matrices to a series of products of sub-matrices. To formally prove that the above construction works, we also need the help from the Sylvester's inequality (Marsaglia and PH Styan, 1974).

Lemma 20 (Sylvester's inequality) *Given $M \in \mathbb{C}^{m \times n}$ and $M' \in \mathbb{C}^{n \times p}$, we have*

$$\text{Rank}(MM') + n \geq \text{Rank}(M) + \text{Rank}(M'),$$

where Rank *outputs the matrix rank.*

We are ready to show

Theorem 21 *Given a rank n Cartesian tensor product space, $\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}$ for all pair of paths p_i and p_j of which the last weights are the same, form an orthogonal basis of the equivariant design space with respect to the Cartesian space. Here, the orthogonality means that the Frobenius inner products of the different path matrices are always zero,*

$$\left\langle \hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}, \hat{P}^{(p_{i'})} \left(\hat{P}^{(p_{j'})} \right)^{\top} \right\rangle_F = 0, \tag{21}$$

for any $i \neq i'$ or $j \neq j'$. Given rank n, the dimension of the equivariant space equals

$$\sum_{k=0}^{n} \left(\sum_{u=0}^{\lfloor (n-k)/3 \rfloor} (-1)^u \frac{n!(2n - 3u - k - 2)!}{u!(n-u)!(n-2)!(n-3u-k)!} \right)^2. \tag{22}$$

Proof *To utilize Lemma 19, we need to prove that each of the $\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}$ is a linear mapping from a weight l ICT space to some other (including itself) ICT space. Thus, these matrix (linear mapping) need to satisfy the requirements of Definition 11, except for 2), i.e., the sum of these matrices need not to be an identity matrix. Also, the orthogonality is defined based on the Frobenius inner product. We first prove 1), the equivariance. For arbitrary $\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}$, and group element g, we have*

$$\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top} \rho_{(R^3)^{\otimes n}}(g) = \hat{P}^{(p_i)} D^{-1} A^{(p_j \rightarrow p_j)} \rho_{(R^3)^{\otimes n}}(g)$$

$$= \hat{P}^{(p_i)} \rho_l(g) \left(\hat{P}^{(p_j)} \right)^{\top} = \rho_{(R^3)^{\otimes n}}(g) \hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}, \tag{23}$$

where A is a selection matrix defined in equation (19). Then, for 3), we prove that $\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top}$ has matrix rank $2l' + 1$, where l' is the last weight of the path p_j. We know that the shapes of these two matrices are both $3^n \times (2l' + 1)$, and the transpose has shape $(2l' + 1) \times 3^n$. From the rank inequality, it holds that

$$\text{Rank} \left(\hat{P}^{(p_i)} \left(\hat{P}^{(p_j)} \right)^{\top} \right) \leq \min \left(\text{Rank} \left(\hat{P}^{(p_i)} \right), \text{Rank} \left(\left(\hat{P}^{(p_j)} \right)^{\top} \right) \right)$$

$$= 2l' + 1. \tag{24}$$

On the other hand, from Lemma 20, we have that

$$\text{Rank}\left(\hat{P}^{(p_i)}\left(\hat{P}^{(p_j)}\right)^{\top}\right) \geq \text{Rank}\left(\hat{P}^{(p_i)}\right) + \text{Rank}\left(\left(\hat{P}^{(p_j)}\right)^{\top}\right) - 2l' - 1$$
$$=2(2l'+1) - 2l' - 1$$
$$=2l'+1. \tag{25}$$

Equations (24) and (25) together give us

$$\text{Rank}\left(\hat{P}^{(p_i)}\left(\hat{P}^{(p_j)}\right)^{\top}\right) = 2l'+1, \tag{26}$$

which satisfies condition 3) of the definition.

The proof of 4) is almost identical to that of Theorem 17.

To demonstrate orthogonality, we need to reconsider the construction of basis elements in its equivalent form: the transposed change-of-basis matrix $D^{\top}$ multiplied by the selection matrix A and then by the change-of-basis matrix D, as shown in equation (20). It holds that

$$\left\langle \hat{P}^{(p_i)}\left(\hat{P}^{(p_j)}\right)^{\top}, \hat{P}^{(p_{i'})}\left(\hat{P}^{(p_{j'})}\right)^{\top}\right\rangle_F$$
$$=\left\langle DA^{(p_j \to p_i)}D^{\top}, DA^{(p_{j'} \to p_{i'})}D^{\top}\right\rangle_F$$
$$=\left\langle \underbrace{D^{\top}D}_{I} A^{(p_j \to p_i)} \underbrace{D^{\top}D}_{I}, A^{(p_{j'} \to p_{i'})}\right\rangle_F$$
$$=\left\langle A^{(p_j \to p_i)}, A^{(p_{j'} \to p_{i'})}\right\rangle_F$$
$$=0, \tag{27}$$

for any $i \neq i'$ or $j \neq j'$. Here $D^{\top}D = I$ holds because of the fact that the change-of-basis matrix D is an orthogonal matrix, as we stated in the proof of Theorem 17. The dimension calculation is based on Proposition 10. ∎

It is important to note that we cannot define a basis where each basis element maps a vector to an orthogonal one, as was done for the ICT decomposition matrices in Theorem 17. The proof is simple: By calculating the cardinality of the basis using equation (22), we find that the cardinality of basis of the rank 5 tensor product space is 603. However, the number of dimensions in this tensor product space is $3^5 = 243$, which is less than 603. Thus, it is impossible to generate so many orthogonal vectors using this equivariant basis. Instead, we define the orthogonality based on the Frobenius norm. Frobenius orthogonality is a natural generalization of L_2 orthogonality for vectors. There are two main advantages of maintaining such orthogonality. First, it is very easy to project an arbitrary matrix onto an equivariant linear mapping. Specifically, let $\{\hat{M}_i\}_{1 \leq i \leq n}$ be the orthogonal basis, and given a matrix M, we can immediately obtain the projection in the equivariant space by

$$\hat{M} = \sum_{i=1}^{n} \langle \hat{M}_i, M \rangle_F \hat{M}_i,$$

which is not easy for spectral norm, for example, as it is not induced by inner product. Second, given the orthonormal basis, it can avoid the situation that the weights blow up. For instance, given a 3-d space (x, y, z), suppose that our target space is the 2-d space $z = 0$, we have a basis of two elements to span this space, $(1, 0, 0)$ and $(1, 10^{-5}, 0)$, which requires very high weights to merely describe $(1, 1, 0)$. Instead, if our basis is $(1, 0, 0)$ and $(0, 1, 0)$, we can just let $(1, 1, 0) = 1 * (1, 0, 0) + 1 * (0, 1, 0)$. So the Frobenius orthogonality enhances the numerical stability. Note that the ICT decomposition matrices are also part of the basis of the equivariant space.

6. Extension to General Spaces

So far, we have introduced the method to find bases for equivariant design spaces where the input and the output spaces are the same. Here, we focus on extending the method to equivariant spaces between different spaces.

6.1 General ICT Decomposition

It is appealing to build equivariant layers between different desired spaces. So far, we have only discussed Cartesian tensor product spaces. The spherical tensor product spaces enable us to utilize high-order many-body information while remaining flexible in terms of memory cost and inference speed. However, it is difficult to construct such a layer. For example, if the input space is $(l = 4 \otimes l = 5)$ and the output space is $(l = 2 \otimes l = 6)$, how can we parameterize the equivariant layer? More specifically, what is the basis of this layer? It is even not possible to find a spanning set by Pearce-Crump (2023b), as it is for Cartesian tensor product spaces. Before answering this question, we first introduce the general problem of ICT decomposition and how to efficiently perform a general ICT decomposition on arbitrary input spaces.

Definition 22 *Given a space $\mathcal{V} = (l_{11} \otimes \cdots \otimes l_{1i_1}) \oplus \cdots \oplus (l_{k1} \otimes \cdots \otimes l_{ki_k})$ with dimension d, a general irreducible decomposition is a series of matrices $\hat{H}^{(\mathcal{V};l;q)} \in \mathbb{R}^{d \times d}$, such that:*

    1) $\hat{H}^{(\mathcal{V};l;q)}$ is $O(3)$-$(\mathcal{V}, \mathcal{V})$-equivariant.

    2) The summation of the decomposition equals to identity matrix, $\sum_{l,q} \hat{H}^{(\mathcal{V};l;q)} = I$.

    3) The rank (matrix rank) of the decomposition matrix $\hat{H}^{(\mathcal{V};l;q)}$ is $2l + 1$.

    4) $\{\hat{H}^{(\mathcal{V};l;q)} v \mid v \in \mathcal{V}\}$ has no non-trivial $O(3)$-invariant subspace.

    5) $\{\hat{H}^{(\mathcal{V};l;q)} v \mid v \in \mathcal{V}\} \bigcap \{\hat{H}^{(\mathcal{V};l';q')} v \mid v \in \mathcal{V}\} = \{0\}$ holds for $l \neq l'$ or $q \neq q'$.

If $\hat{H}^{(\mathcal{V};l;q)}$ are orthogonal to each other, as Lemma 12, we further call it an orthogonal general irreducible decomposition.

Intuitively, an orthogonal general ICT decomposition can be obtained using a strategy similar to that of the classic ICT decomposition. However, we cannot use the parentage scheme directly, as it only accepts a tensor product with an $(l = 1)$ one step at a time, proceeding from left to right. Here, we introduce a general parentage scheme. Unlike the classic scheme, this is not a fixed procedure.

Given a space

$$\mathcal{V} = \underbrace{(l_{11} \otimes \cdots \otimes l_{1i_1})}_{\mathcal{V}_1} \oplus \cdots \oplus \underbrace{(l_{k1} \otimes \cdots \otimes l_{ki_k})}_{\mathcal{V}_k},$$

we construct the general parentage scheme as follows. For each $\mathcal{V}_{1 \leq t \leq j}$, we read the first weight l_{t1}, of this tensor product space, and place multiplicity 1 in the position of column 1 (sometimes we omit drawing explicit x-axis) and row l_{t1}. This is what we draw in the first column. After we finish drawing in column j, we draw connected lines between the j-th and $(j + 1)$-th columns, and write multiplicities of each row of the $(j + 1)$-th column. To do this, we traverse each row of the multiplicities q in the j-th column. For a given multiplicity q of weight (row index) l_j, we pick up the row with weight l_{j+1} in the next column $j + 1$ that satisfies $|l_j - l_{t(j+1)}| \leq l_{j+1} \leq l_j + l_{t(j+1)}$, then we write multiplicity q in $(j + 1, l_{j+1})$, and draw a line connecting (j, l_j) and $(j + 1, l_{j+1})$. After that, we write $l_{t(j+1)}$ on top of the line. If this position already has a multiplicity, we add q to the existed multiplicity. We repeat this process until we traverse to the end of this tensor product space. Different schemes from different tensor product spaces are drawn independently. Clearly, this is a generalization of the classic parentage scheme. The latter can be seen as a general parentage scheme when the only tensor product space is $(l = 1)^{\otimes n}$, and the previous steps exactly follow. The general parentage scheme can handle those situations that the space is complicated rather than just a tensor product of Cartesian spaces.

For example, given a space $(3 \otimes 4 \otimes 2) \oplus (2 \otimes 3)$, we first separate the terms by the direct sum $\oplus$, resulting in $(3 \otimes 4 \otimes 2)$ and $(2 \otimes 3)$. Then, we use the general parentage scheme to decompose $(3 \otimes 4 \otimes 2)$ into irreducible spherical spaces and construct the path matrices. We begin by writing a multiplicity of one in the weight-3 row, in the leftmost column. Next, we take the tensor product with $(l = 4)$, of which the decomposition results in multiplicities of one in $(l = 7), \ldots, (l = 1)$. We thereby construct the path matrices $(3 \xrightarrow{4} 7), \ldots, (3 \xrightarrow{4} 1)$. Note that we mark 4 on top of the arrow, which we refer to as the bridge number. Then, 2 is taken as the bridge number, and we accordingly generate new path matrices. Consequently, we apply the same procedure for $(2 \otimes 3)$. An important property is

Corollary 23 (from Lemma 13) *Given a tensor product space, the path matrices for **the same** general parentage scheme have orthogonal columns, and the L_2 norms of columns of each path matrix are the same.*

Proof *The proof is mostly identical to that of Lemma 13.* ∎

*Here the words **the same** suggest that we need to fix our bridge number to obtain orthogonal path matrices. As our previous example, $(3 \xrightarrow{4} 7 \xrightarrow{2} 6)$ and $(4 \xrightarrow{3} 7 \xrightarrow{2} 6)$ will generate two different path matrices. But they are actually identical up to an index permutation. The result will be the same if we accordingly permute the index of our input vector. We can multiply the path matrices with their inverse to obtain decomposition matrices, as the classic ICT decomposition does. Here we should note that the decomposition matrices of different sub-schemes separated by the direct sum cannot be generally concatenated. We can, in general, concatenate some zeros to the path matrices and their transpose with short rows and columns, respectively. Or we can just leave the decomposition matrices of different shapes as they are, and multiply the matrices with different segments of the input vector.*

Constructing such path matrices, and multiplying with its inverse, can contribute to a general irreducible decomposition. The proof that it satisfies Definition 22 is very similar to that of Theorem 17.

6.2 Equivariant Bases of Arbitrary Spaces

The general ICT decomposition gives us some hints. The constructed change-of-basis matrix $D^\top$ changes the basis from $(3 \otimes 4 \otimes 2) \oplus (2 \otimes 3)$ to its decomposed irreducible spherical spaces, denoted as $\mathrm{irr}\left((3 \otimes 4 \otimes 2) \oplus (2 \otimes 3)\right)$, and its inverse D changes back the basis. An orthogonal basis from spaces $\mathcal{V}_1$ to $\mathcal{V}_2$ can be obtained via the change-of-basis matrices constructed based on the general ICT decompositions of these two spaces, but we need to take care of more when considering $O(3)$. The above discussion is identical for $SO(3)$ and $O(3)$. Instead, here, the parity must get involved when we consider arbitrary spaces. First, one should note that a tensor product space, as irreducible spherical spaces, have parity 1 or −1. We have kept in mind that Cartesian space is of parity −1, which means that if we do the reflection, then the space will be reflected accordingly. It is equivalent to an $(l = 1, p = -1)$ spherical space, from Lemma 4. Meanwhile, we should remember that there is also $(l = 1, p = 1)$ spherical space, as a cannot-be-reflected version of the Cartesian space. The parity of the tensor product space is only decided by the number of the componential spaces with odd parities. If the number is odd, then the tensor product space has odd parity, and vice versa. Meanwhile,

Proposition 24 *Given a tensor product space of rank n, the resulting spaces of the ICT decomposition are with the same parity as the tensor product space.*

Proof *Suppose that we have a rank n tensor T in the tensor product space, and assume that the parities of the ICTs and the original rank n tensor T are not the same. If a tensor T has an even parity, then reflection will not change T. Let $\hat{T}_1^{(p=-1)}, \ldots, \hat{T}_k^{(p=-1)}$ be the ICTs with odd parity. According to Definition 22, we should have*

$$
\begin{aligned}
T &= \sum_i \hat{T}_i^{(p=-1)} + \sum_{i'} \hat{T}_{i'}^{(p=1)} \\
&= -\sum_i \hat{T}_i^{(p=-1)} + \sum_{i'} \hat{T}_{i'}^{(p=1)},
\end{aligned}
\tag{28}
$$

which gives us

$$
\sum_i \hat{T}_i^{(p=-1)} = 0.
\tag{29}
$$

This contradicts with the definition that $\hat{T}_i^{(p=-1)}$ are linearly independent. The same arguments can be given for the case that T has odd parity. ∎

Now we are ready to introduce the procedure for obtaining an orthogonal basis for different spaces. Given spaces $\mathcal{V}^{in}$ and $\mathcal{V}^{out}$, we want to find a basis for the equivariant design space from $\mathcal{V}^{in}$ to $\mathcal{V}^{out}$. We first find the irreducible spherical spaces of $\mathcal{V}^{in}$ and $\mathcal{V}^{out}$ via the general parentage scheme, denoted as $\mathrm{irr}(\mathcal{V}^{in})$ and $\mathrm{irr}(\mathcal{V}^{out})$, and write down the paths. Then,

the paths are grouped according to the last weight in them, as well as the parities of the original tensor product spaces, denoted as $Q_{\mathcal{V}^{in}}^{(l;p)}$ and $Q_{\mathcal{V}^{out}}^{(l;p)}$. Let our equivariant space be $\mathrm{Hom}_{O(3)}(\mathcal{V}^{in}, \mathcal{V}^{out})$, following from Lemma 16 (Schur's lemma), we have

$$\dim \mathrm{Hom}_{O(3)}(\mathcal{V}^{in}, \mathcal{V}^{out}) = \sum_{l,p} \# Q_{\mathcal{V}^{in}}^{(l;p)} \# Q_{\mathcal{V}^{out}}^{(l;p)}, \tag{30}$$

where $\#\cdot$ represents the number of elements in the set. We drop those paths sets that the last weight and the same parities do not show in both spaces. Then, we can generate path matrices based on those remaining sets. Finally, we conclude that

Proposition 25 Given spaces $\mathcal{V}^{in} = \left((l_1^1 \otimes \cdots \otimes l_{k_1}^1), \hat{p}^1\right) \oplus \cdots \oplus \left((l_1^t \otimes \cdots \otimes l_{k_t}^t), \hat{p}^t\right)$, and $\mathcal{V}^{out} = \left((l_1'^1 \otimes \cdots \otimes l_{k_1'}'^1), \hat{p}'^1\right) \oplus \cdots \oplus \left((l_1'^{t'} \otimes \cdots \otimes l_{k_{t'}'}'^{t'}), \hat{p}'^{t'}\right)$, where l and $\hat{p}$ are weights and parities, respectively.

$$\left\{ \hat{P}^{(p^{(out;l;\hat{p};q')})} \left(\hat{P}^{(p^{(in;l;\hat{p};q)})}\right)^{\top} \;\middle|\; p^{(in;l;\hat{p};q)} \in Q_{\mathcal{V}^{in}}^{(l;\hat{p})}, p^{(out;l;\hat{p};q')} \in Q_{\mathcal{V}^{out}}^{(l;\hat{p})}, \hat{p} \in \{-1, 1\} \right\}$$

forms an orthogonal basis of $\mathrm{Hom}_{O(3)}(\mathcal{V}^{in}, \mathcal{V}^{out})$.

Proof The proof generally follows Theorem 21. Here, we focus on the orthogonality. Since $\hat{P}^{(p^{(out;l;\hat{p};q')})} \left(\hat{P}^{(p^{(in;l;\hat{p};q)})}\right)^{\top}$ are mappings between $\mathcal{V}^{in}$ and $\mathcal{V}^{out}$, they can be represented as a $\sum_{j=1}^{t} \left(\Pi_{i=1}^{k_j} \left(2l_i^j + 1\right)\right) \times \sum_{j=1}^{t'} \left(\Pi_{i=1}^{k_j'} \left(2l_i^j + 1\right)\right)$ matrix. From the mapping relations, we know that it is a block matrix as

$$\begin{bmatrix} M_{11} & \cdots & \cdots & M_{1t} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ M_{t'1} & \cdots & \cdots & M_{t't} \end{bmatrix}. \tag{31}$$

The sub-matrices are of shape $\Pi_{i=1}^{k_{j'}'} \left(2l_i^{j'} + 1\right) \times \Pi_{i=1}^{k_j} \left(2l_i^j + 1\right)$ for some j and j'. $\hat{P}^{(p^{(out;l;\hat{p};q')})} \left(\hat{P}^{(p^{(in;l;\hat{p};q)})}\right)^{\top}$ will be in the same column for those $(l; \hat{p}; q')$ belonging to the same direct sum component of $\mathcal{V}^{out}$, and the same row for those $(l; \hat{p}; q)$ belonging to the same direct sum component of $\mathcal{V}^{in}$. When the non-zero values of $\hat{P}^{(p^{(out;l;\hat{p};q')})} \left(\hat{P}^{(p^{(in;l;\hat{p};q)})}\right)^{\top}$ appear in a block M, the rest of the sub-matrices will be zeros. Thus, given two elements in the basis, their corresponding Frobenius inner product must be zero if the non-zero values do not show in the same block M. If they appear in the same M, then the question is equivalent to Theorem 21, where we can prove that it is also zero. ∎

The algorithm for generating equivariant bases is presented in Algorithm 2. Here, CG_decomp_generator receives the current weight and the bridge number as input and
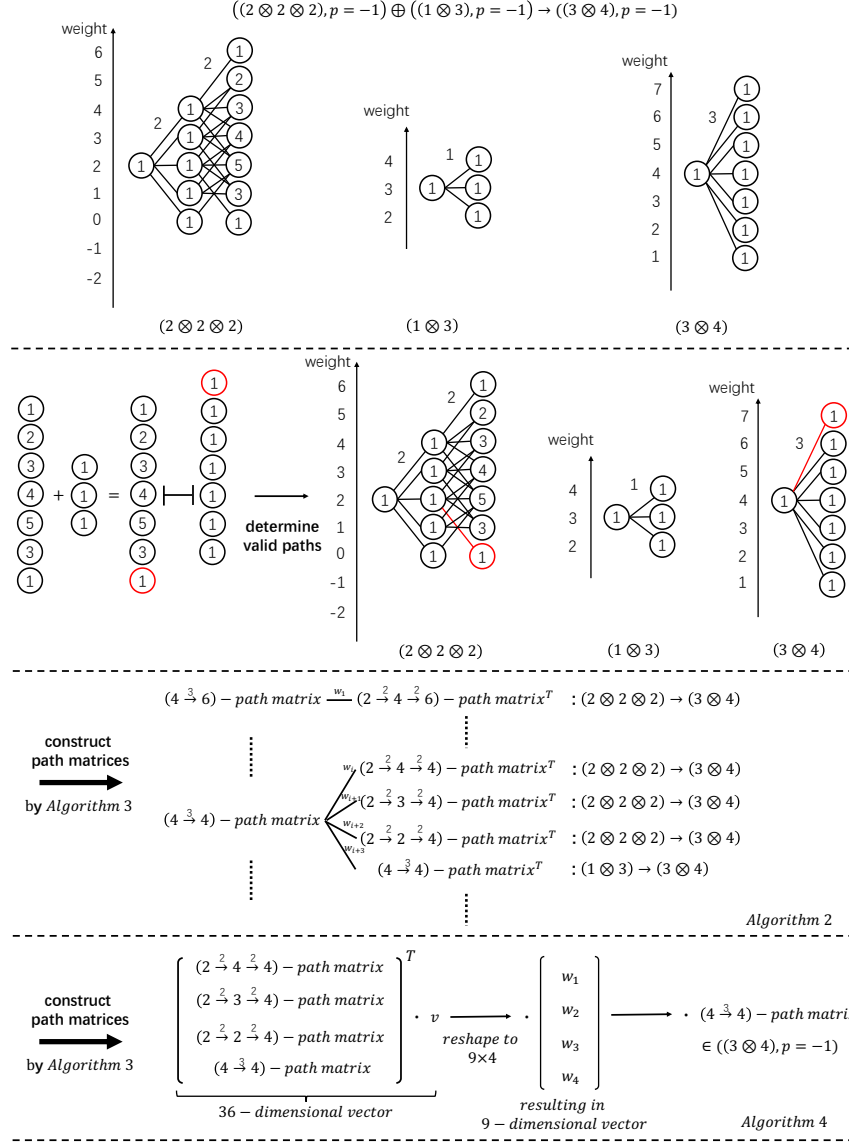
Figure 4: The general parentage scheme with the example of finding basis for $((2 \otimes 2 \otimes 2), p = -1) \oplus ((1 \otimes 3), p = -1) \to ((3 \otimes 4), p = -1)$. Compared to the classic parentage scheme, we need to additionally mark the bridge number in the scheme.

outputs the possible weights according to the selection rules of CG coefficients. These rules depend on the group type. For instance, in the case of $O(3)$, l_1 and l_2 generate possible weights ranging from $|l_1 - l_2|$ to $l_1 + l_2$.

6.3 An Example for Obtaining a Basis of an Equivariant Space

Here, we show an example for obtaining a basis of the equivariant space from

$$((2 \otimes 2 \otimes 2), p = -1) \oplus ((1 \otimes 3), p = -1)$$

to

$$((3 \otimes 4), p = -1).$$

Algorithm 2 Equivariant Basis Generation

Input: Input space

$$\mathcal{V}_{in} = \underbrace{(l_{11}^{in} \otimes \cdots \otimes l_{1i_1}^{in})}_{\mathcal{V}_{in}^1} \oplus \cdots \oplus \underbrace{(l_{k1}^{in} \otimes \cdots \otimes l_{ki_k}^{in})}_{\mathcal{V}_{in}^k},$$

and output space

$$\mathcal{V}_{out} = \underbrace{(l_{11}^{out} \otimes \cdots \otimes l_{1i_1}^{out})}_{\mathcal{V}_{out}^1} \oplus \cdots \oplus \underbrace{(l_{k1}^{out} \otimes \cdots \otimes l_{ki_k}^{out})}_{\mathcal{V}_{out}^k}.$$

Step 1. For each tensor product space $\mathcal{V}^t$ from $\mathcal{V}_{in}$ and $\mathcal{V}_{out}$, we draw the corresponding general parentage scheme as follows. Initialize a matrix $P = C^{000}$, and flatten the first two indices. Then, let it contract with CG tensor $C^{l_{t1}0l_{t1}}$,

$$P^{(l_{t1})} = \text{flatten}_2(P \odot C^{l_{t1}0l_{t1}}),$$

and from now on we maintain path p_j with $p_1 = l_{t1}$, where j is the column index. We start from $j = 1$.

Step 2. Suppose that we are at the j-th column, we pick up those available weights in the $(j+1)$-th column, i.e., weights $|l_j - l_{t(j+1)}| \leq l_{j+1} \leq l_{t(j+1)} + 1$, update the path by adding l_{j+1} to the end of path p_j to obtain p_{j+1}, and calculate the contraction,

$$P^{(p_{j+1})} = \text{flatten}_2(P^{(p_j)} \odot C^{l_{t(j+1)}l_j l_{j+1}}).$$

Then, we update the column number $j \leftarrow j + 1$.

Step 3. Repeat **Step 2** until j reaches the number of components of the tensor product space. For each obtained path matrices $P^{(p)}$, we normalize its columns by

$$\hat{P}_{*,c}^{(p)} = \frac{P_{*,c}^{(p)}}{\|P_{*,c}^{(p)}\|_2}.$$

Step 4. Repeat **Step 3** for each space, and divide the path matrices into two groups, from the input and the output space, respectively.

Step 5. For each path matrix $\hat{P}_{in}^{(p)}$ from input space $\hat{P}_{in}$, multiply it with the transpose of one with the same termination of the path p, denoted by $\hat{P}_{out}^{(p')}$ from the output space $\hat{P}_{out}$. Lastly, multiply a learnable parameter w, and we get $w\hat{P}_{out}^{(p')}(\hat{P}_{in}^{(p)})^\top$.

Output: The list of all $w\hat{P}_{out}^{(p')}(\hat{P}_{in}^{(p)})^\top$.

This process is illustrated in Figure 4. First, we need to construct our general parentage schemes to identify all possible paths. For the tensor product $(2 \otimes 2 \otimes 2)$, the starting point of our scheme must be 2. We then choose one of the remaining two 2s as the bridge number,

which gives us five possible paths, ranging from $(2 \xrightarrow{2} 0)$ to $(2 \xrightarrow{2} 4)$. Next, we select the last 2 as the bridge number, leading to additional paths and resulting in a total of 19 paths. We repeat this process to draw general schemes for $(1 \otimes 3)$ and $(3 \otimes 4)$, yielding 3 and 7 paths, respectively. Each scheme must have a parity, determined by the corresponding tensor product space; in this case, all the parities are equal to -1. Next, we focus on the weights in the rightmost column of each scheme. We omit those weights that do not appear in both the input and the output spaces. In this example, the omitted weights are 0 (which does not appear in the output space) and 7 (which does not appear in the input space). We then construct the path matrices based on the remaining paths. The path matrices of the input and output spaces with matching last weights are grouped into pairs. The products of the path matrices of the output space and the transpose of the corresponding input space matrices form the basis elements of our equivariant maps. This procedure is described in Algorithm 2. Generating equivariant bases as matrices is elegant and enables efficient computations. These bases can also be efficiently obtained. However, when we tested the spanning set method proposed by Pearce-Crump (2023b), assigning a scalar parameter to each basis and combining these bases to form a single matrix, we encountered memory issues in real-world neural network frameworks (e.g., PyTorch). This is because maintaining a large computation graph is challenging when the rank is very high. To address this issue in high-rank cases, it is preferable to store the path matrices with the same last weights for the input and output spaces separately. For each available last weight, assuming we have k paths for the input space and k' paths for the output space, we maintain a $k' \times k$ matrix with learnable parameters. During training and inference, the inverse path matrices of the input space first convert the input into the spherical direct sum space. Then, the $k' \times k$ matrices linearly combine the spherical vectors. Finally, the path matrices of the output space convert the vectors back to the output space. The entire process is also illustrated in Figure 4. This approach follows from the previous proofs and aligns with the core idea of this paper: performing operations in spherical spaces is simpler and more efficient. This process is detailed in Algorithm 3.

6.4 Multi-Channel Features, Activation Functions, Normalizations, and Biases

We have already introduced how to construct equivariant linear operations between some $\mathcal{V}^{in}$ and $\mathcal{V}^{out}$. In a real implementation, additional considerations are required to make it an equivariant layer in a modern neural network. First, if we take vectors in $\mathcal{V}^{in}$ and $\mathcal{V}^{out}$ as features, it is both standard and beneficial for a neural network to have multi-channel features. Fortunately, this is straightforward in our framework. We can simply let $\left(\mathcal{V}^{in}\right)^{\oplus K^{in}}$ and $\left(\mathcal{V}^{out}\right)^{\oplus K^{out}}$ serve as our input and output spaces, respectively. In this case, the conclusion in Section 6.2 can be seamlessly applied. Notably, Proposition 25 already addresses the scenarios described here. Thus, for a space

$$\underbrace{\left((l_1^1 \otimes \cdots \otimes l_{k_1}^1), p^1\right)}_{\mathcal{V}_\oplus^1} \oplus \cdots \oplus \underbrace{\left((l_1^t \otimes \cdots \otimes l_{k_t}^t), p^t\right)}_{\mathcal{V}_\oplus^t}, \tag{32}$$

each direct sum component $\mathcal{V}_\oplus^i$ is a single-channel feature in a multi-channel view, which can be different depending on the positions. This is more general than the usual multi-channel features.

Algorithm 3 Equivariant Basis Generation—Linear Combinations in Spherical Spaces

Input: Input space

$$\mathcal{V}_{in} = \underbrace{(l_{11}^{in} \otimes \cdots \otimes l_{1i_1}^{in})}_{\mathcal{V}_{in}^1} \oplus \cdots \oplus \underbrace{(l_{k1}^{in} \otimes \cdots \otimes l_{ki_k}^{in})}_{\mathcal{V}_{in}^k},$$

and output space

$$\mathcal{V}_{out} = \underbrace{(l_{11}^{out} \otimes \cdots \otimes l_{1i_1}^{out})}_{\mathcal{V}_{out}^1} \oplus \cdots \oplus \underbrace{(l_{k1}^{out} \otimes \cdots \otimes l_{ki_k}^{out})}_{\mathcal{V}_{out}^k}.$$

Step 1. Run the same **Steps 1-4** in Algorithm 2, by which we obtain the path matrices $\hat{P}_{out}^{(p')}$ and $\hat{P}_{in}^{(p)}$.

Step 2. Collect the last weight appearing in the path p of each path matrix.

Step 3. For the input space, maintain those path matrices that have the corresponding matrices in the output space with the same last weight in the paths (one can also run this step for the output space without loss of generality).

Step 4. For each last weight, we calculate the number of path matrices whose path terminates at that weight, in both input and output spaces, denoted by n and m, respectively.

Step 5. Create an $n \times m$ learnable weight matrix W for each last weight.

Output: The list of W, $\hat{P}_{out}^{(p')}$, and $\hat{P}_{in}^{(p)}$

On the other hand, activation functions introduce non-linearity to neural networks. As non-linear functions, their design space is very limited in order to preserve symmetry. However, it is completely safe if we can construct invariant linear mappings of the form $f^{inv} : v^{out} \in \mathcal{V}^{out} \mapsto c \in \mathbb{R}$. Then, we can apply an activation function to c, obtaining $\phi(c)$, and multiply $\phi(c)$ with v^{out} to produce the activated result $\phi(f^{inv}(v^{out}))v^{out}$. This process is equivariant, as

$$\phi\left(f^{inv}\left(\rho_{\mathcal{V}^{out}}(g) \odot v^{out}\right)\right) \rho_{\mathcal{V}^{out}}(g) \odot v^{out}$$

$$= \phi\left(\underbrace{\rho_{\mathbb{R}}(g)}_{I} \odot f^{inv}\left(v^{out}\right)\right) \rho_{\mathcal{V}^{out}}(g) \odot v^{out}$$

$$= \rho_{\mathcal{V}^{out}}(g) \odot \left(\phi\left(f^{inv}\left(v^{out}\right)\right) \odot v^{out}\right). \tag{33}$$

This process is similar to that of the spherical space-based methods (Batzner et al., 2022; Musaelian et al., 2023), but here the strategy is applicable to more general spaces.

For the normalizations, one can leverage the fact that the L_2-norm of the vectors are invariant to the rotation group, in general spaces. This is because the matrix representations of $O(n)$ always have an absolute determinant that is equal to one. Thus, for a vector $v \in \mathcal{V}$, we can simply calculate $\|v\|_2$ to construct an equivariant normalization layer. The discussion on biases is mainly identical to Pearce-Crump (2023b). We can construct equivariant basis from $\mathbb{R}$ to arbitrary $\mathcal{V}^{out}$. In our work, we can find an orthogonal basis

Rank	4	5	6	7	8	9
EMLP (Finzi et al., 2021)	2m52s	26m	OOM	OOM	OOM	OOM
Algorithm 2	<0.1s	1s	3s	10s	OOM	OOM
Algorithm 3	<0.1s	<0.1s	<0.1s	<0.1s	0.8s	3.4s

Table 4: Comparison on the basis generation speed of $\mathrm{End}_{O(3)}\left(\left(\mathbb{R}^3\right)^{\otimes n}\right)$. The out-of-memory error occurs when generating the bases using EMLP. In contrast, for Algorithm 2, the error occurs when multiplying the learnable parameters.

through Proposition 25 instead of the spanning set. Here we should note that there are some situations that we cannot find such a mapping to $\mathcal{V}^{out}$. For example, $\mathbb{R}$ with odd parity cannot map to $\mathcal{V}^{out}$ with even parity, and vice versa. Also, $(l = 0, p = 1)$ cannot map to $(l = 3, p = 1) \otimes (l = 2, p = 1)$ either, as $(l = 0, p = 1)$ is not in their irreducible decomposition set $\mathrm{irr}\left((l = 3, p = 1) \otimes (l = 2, p = 1)\right) = \{(l = 5, p = 1), (l = 4, p = 1), (l = 3, p = 1), (l = 2, p = 1), (l = 1, p = 1)\}$. *Thus, it is convenient to always maintain $l = 0$ irreducible representations for the ease of bias functions. Beyond the linear layer constraint, we can also build bias functions via multiplying $c \in \mathbb{R}$ with each irreducible representations of unit L_2 norm, and adding the results with the irreducible representations.*

7. Related Works

Research on equivariant neural networks is rooted in an abstract concept raised by Cohen and Welling (2016), where they extend the translation invariance of CNNs (LeCun et al., 1989) to a wide range of groups. Meanwhile, graph neural networks (Gori et al., 2005) are proven to be suitable tools for handling tasks involving point clouds. Their coupling, Equivariant Graph Neural Networks (EGNNs) has initiated an emerging area with very important applications to physics-informed tasks. Among the implementation, Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017) are the most popular, where the constructed message is propagated to the neighboring points. Before that, Schütt et al. (2017) observe that group CNN frameworks require the points to lie on some grids and propose SchNet to flexibly handle atoms that randomly appear in space. Gasteiger et al. (2020) further leverage directional information to enhance the model expressivity. At this point, the features utilized are just scalars and vectors, and subsequent works begin to use high-weight irreducible representations to effectively encode many-body interactions. Thomas et al. (2018) propose an abstract framework, the Tensor Field Network, to incorporate spherical irreducible representations into MPNNs. This abstract framework has been implemented with different designs to achieve SOTA performances on multiple popular benchmarks (Chmiela et al., 2017; Ruddigkeit et al., 2012; Fu et al., 2023). Gasteiger et al. (2021) demonstrate the universality of using such spherical irreducible representations. Batzner et al. (2022) implement the Tensor Field Network with high-weight irreducible representations, showing strong performance. On the other hand, Drautz (2019) proposes a local descriptor-based

method, Atomic Cluster Expansion (ACE), to encode the atomic information via polynomial basis functions. Batatia et al. (2022) further use high-order message-passing techniques to reduce the required number of layers utilizing spherical irreducible representations. Musaelian et al. (2023) propose a strictly local EGNN, constructing high-weight spherical irreducible representations among edges. Recently, several studies leverage the fact that Cartesian tensor is more computationally efficient. Simeon and De Fabritiis (2024) first propose to use Cartesian tensor but only with rank-2 ICT decomposition, and they further apply it to charged molecules and spin states (Simeon et al., 2024). Cheng (2024) uses high-rank Cartesian tensors but did not manipulate the irreducible components. Zaverkin et al. (2024) successfully incorporate high-weight ICT to MPNNs but only address the highest weight of a given Cartesian tensor, lacking a full treatment of all ICTs for a given Cartesian tensor.

Research on ICT decomposition has a long history and remains a highly active area in theoretical chemistry and chemistry physics communities, in addition to the recent strong interest from the machine learning community. The rank-2 ICT decomposition is well known. Since mid-20th century, Coope et al. (1965) provide the explicit ICT decomposition for rank-3 Cartesian tensors. Seventeen years later, Andrews and Ghoul (1982) derives the rank-4 formula for this problem. Dinçkal (2013) further refines Coope et al. (1965)'s results to make them orthogonal. Very recently, Bonvicini (2024) uses the RREF algorithm to handle the core part of finding linearly independent components in Andrews and Ghoul (1982)'s method, to successfully find rank-5 decomposition matrices. Zou et al. (2001) also propose a general method to perform the decomposition in a recursive style. However, this approach requires recursively decomposing each tensor, rather than having a decomposition tensor that can be directly applied to arbitrary tensors. Moreover, previous works rely on finding the linearly independent components of isotropic tensors, which scales in a factorial level, limiting the efficiency. Snider (2017) summarizes the development for ICT in his book.

Meanwhile, researchers aim to understand what operations can be performed on a Cartesian tensor to maintain equivariance. This is one of the reasons ICTs are significant, as linear combinations of them are naturally equivariant. Pearce-Crump (2023b) leverages Brauer (1937)'s theorem to obtain a spanning set for the equivariant design space. Based on this, Pearce-Crump (2023a) further proposes an algorithm to generate the equivariant-preserving operation on the tensor product space. However, a direct construction of an orthogonal equivariant basis remains unknown. This work proposes a method to directly obtain orthogonal bases of equivariant design spaces.

8. Discussion

In this section, we discuss the relationship between our work and the studies by Pearce-Crump (2023b) and Finzi et al. (2021). Additionally, we explore the application of our method to other groups.

8.1 Relation to Other Works

The remarkable work done by Pearce-Crump (2023b) has successfully found spanning sets for the equivariant design spaces for various groups. Here, we discuss the relation to this work. First, we recall the theorem for $O(n)$:
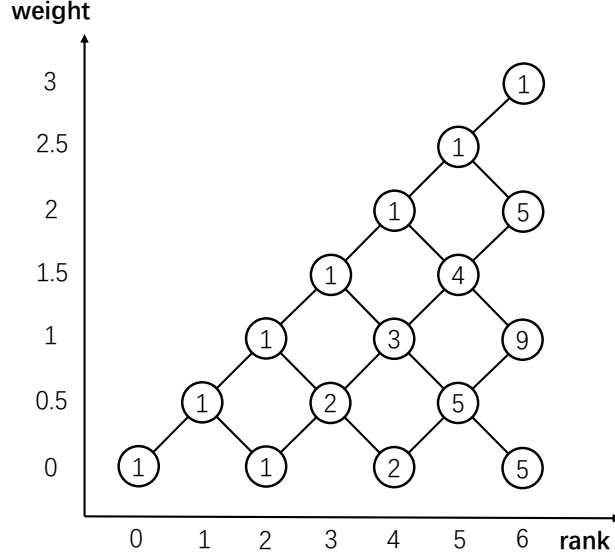
Figure 5: The parentage scheme of decomposition of $(l = 0.5)^{\otimes n}$ for $SU(2)$ and $U(2)$, where n is the rank. The numbers in the circle represent multiplicity.

Theorem 26 (Pearce-Crump (2023b)) $U_{i_1,j_1} U_{i_2,j_2} \ldots U_{i_{\frac{l+k}{2}} j_{\frac{l+k}{2}}}$ *form a spanning set for* $\mathrm{Hom}_{O(n)}\left((\mathbb{R}^n)^{\otimes l}, (\mathbb{R}^n)^{\otimes k}\right)$, *where* U *are arbitrary matrices (also rank-2 tensors),* i_i *and* j_i *are different permutations of integer* $1, \ldots, \frac{l+k}{2}$. $\mathrm{Hom}_{O(n)}\left((\mathbb{R}^n)^{\otimes l}, (\mathbb{R}^n)^{\otimes k}\right) = \varnothing$ *when* $l + k$ *is odd, so the above* $\frac{l+k}{2}$ *is well defined.*

Compared to his theorem, we identify orthogonal bases between arbitrary spaces where the input and the output spaces can be any spherical tensor product spaces, not limited to the Cartesian tensor product spaces. For example, Pearce-Crump (2023b)'s theorem cannot handle $((l = 3 \otimes l = 4), p = -1) \to ((l = 2 \otimes l = 2), p = -1)$, but our method can well address it, with an orthogonal basis instead of the spanning set. As he has noted, there is no equivariant operation between Cartesian spaces of odd and even ranks, respectively, while through our method we can find an orthogonal basis if their parities are the same.

There is also a striking advance made by Finzi et al. (2021), who aim to find the basis of equivariant spaces. However, they adopt a numerical algorithm, EMLP, while we provide an analytical solution. As noted by Pearce-Crump (2023b), Finzi et al. (2021) cannot generate bases for continuous group tensor product spaces with higher ranks due to memory constraints and speed limitations. To further evaluate the performance of basis generation, we compare our method with that of Finzi et al. (2021) using their open-sourced library. The results are presented in Table 4, where EMLP takes 26 minutes to compute the basis of $\mathrm{End}_{O(3)}\left((\mathbb{R}^{\otimes 5})\right)$. In contrast, our method computes the same basis in less than one second and is capable of obtaining the orthogonal basis of $\mathrm{End}_{O(3)}\left((\mathbb{R}^{\otimes 8})\right)$.

8.2 Other Groups

Although $O(3)$ is the most popular and useful group considered in equivariant neural networks, our conclusion can also be extended to $O(n)$, $SO(n)$, and $SU(n)$. The tensor product spaces can always be decomposed to smaller irreducible representation spaces, and we can always construct CG coefficients with orthogonality and normalization. The main challenge lies in obtaining the CG coefficients. Once they are obtained, the procedures outlined in this work can be repeated for those groups, and Proposition 25 will give us the equivariant basis. Fortunately, we already have a robust method for computing CG coefficients for $SU(n)$, proposed by Alex et al. (2011), who also provide an online service to freely compute the CG coefficients for $SU(n)$. One can generally follow the procedure of our work to extend the method to $SU(n)$. There are several relations between $SU(n)$ and certain $SO(n)$. For example, $SO(3)$ is double covered by $SU(2)$, which allows us to directly obtain CG coefficients for $SO(3)$ from those for $SU(2)$. The relationship stems from the fact that $SPIN(n)$ double covers $SO(n)$, and $SU(2)$ is isomorphic to $SPIN(3)$. Further, $SU(2) \otimes SU(2)$ is isomorphic to $SPIN(4)$, so it double covers $SO(4)$. Similarly, $SU(4)$ is isomorphic to $SPIN(6)$, and thus it double covers $SO(6)$. Consequently, the conclusion can also be extended to those $SO(n)$ that are covered by $SU(n')$ for high n. Further research into the Lie algebra of $SO(n)$ has the potential to directly discover the CG coefficients of $SO(n)$, rather than relying on the relationship with $SU(n)$. Additionally, $SO(n)$ can be easily converted to $O(n)$ by considering the parity. We additionally provide code for obtaining basis for equivariant spaces under $U(2)$ and $SU(2)$ in our code base. The parentage scheme is different from that of $O(3)$ and $SO(3)$, as shown in Figure 5.

9. Conclusion

In this work, we leverage the relationship between spherical spaces and their tensor product spaces, and construct path matrices to propose methods to: 1) find high-rank ICT decomposition matrices for $6 \leq n \leq 9$, 2) find orthogonal bases for equivariant layers where the input and the output spaces are identical, and 3) generalize the conclusions to arbitrary pairs of spaces. The conclusions made in this work hold true for $O(n)$, $SO(n)$, and $SU(n)$. However, currently, we can only efficiently calculate $SU(n)$, $O(3)$, and $SO(3)$ due to the availability of known CG coefficients. This also highlights our future work, which will focus on achieving efficient calculations or closed-form formulae for $O(n)$ and $SO(n)$. Furthermore, the discovery of CG coefficients for more general groups can also help us find the basis of equivariant spaces for those groups. The contributions of this work can be directly applied to EGNN designs, which has already been shown to facilitate numerous applications in physics, chemistry, and broader biomedical fields. Examples include machine learning for force fields, drug design, blind docking, molecule generation, and many others. Additionally, there is a growing demand for equivariant methods in robotics (Wang et al., 2024). Thus, our work has the potential to positively impact these related fields. For high-rank ICT decomposition, there are always requirements for its use in theoretical chemistry research, e.g., spectroscopies. Despite these advantages, it is essential to establish strict regulations for methods that could potentially harm the health of both people and animals.

10. Reproducibility Statement

The code to efficiently implement the theory is provided in `https://github.com/ShihaoShao-GH/ICT-decomposition-and-equivariant-bases`. The only external dependency packages are PyTorch and e3nn. The e3nn package can be easily installed using the command `pip install --upgrade e3nn`. The code can run on any computer with Python version > 3.0.0 and PyTorch version > 1.0. The code for generating the basis for $SU(2)$ does not require the e3nn package. However, the code for generating Clebsch-Gordan (CG) coefficients is borrowed from `http://qutip.org/docs/3.1.0/modules/qutip/utilities.html`. The CG coefficients for $SU(n)$ can be obtained from https://homepages.physik.uni-muenchen.de/~vondelft/Papers/ClebschGordan/, as provided by Alex et al. (2011). The code for generating these coefficients is detailed on pages 19–34 of their paper. The comparison with EMLP is based on their code base `https://github.com/mfinzi/equivariant-MLP`.

Appendix A. Proofs

In this section, we provide proofs for auxiliary lemmas.

Lemma 12 *Given matrices $M : \mathcal{V} \to \mathcal{V}'$ and $M' : \mathcal{V} \to \mathcal{V}'$ of the same shape, if $M^\top M' = 0$, then*

$$(Mv) \odot (M'v') = 0$$

for any $v, v' \in \mathcal{V}$.

Proof *It holds true that*

$$
\begin{aligned}
&(Mv) \odot (M'v') \\
&= \sum_i \left(\sum_j M_{ij} v_j \right) \left(\sum_{j'} M'_{ij'} v'_{j'} \right) \\
&= \sum_{j,j'} v_j v'_{j'} \left(\sum_i M_{ij} M'_{ij'} \right).
\end{aligned}
\tag{34}
$$

Assume we have $M^\top M' = 0$, then for any j and j', we have

$$
\sum_i M_{ij} M'_{ij'} = 0.
\tag{35}
$$

Plugging it back to equation (34), we obtain

$$(Mv) \odot (M'v') = \sum_{j,j'} v_j v'_{j'} \left(\sum_i M_{ij} M'_{ij'} \right) = 0. \tag{36}$$

This concludes the proof.

■

Lemma 13 *Given an* $(l_x \otimes l_y, l_2)$*-CG matrix* $\hat{C}^{l_x l_y l_2}$ *and an* (l_1, l_2, l_3)*-CG tensor* $C^{l_1 l_2 l_3}$*, if we let*

$$T_{i_1 i_2 i_3} = \sum_{j_1} \hat{C}^{l_x l_y l_2}_{i_2 j_1} C^{l_1 l_2 l_3}_{i_1 j_1 i_3}$$

and we flatten T *into a* $(2l_1 + 1)(2l_x + 1)(2l_y + 1) \times (2l_3 + 1)$ *matrix, then the following statements hold true:*

1) T *is* $O(3)$*-*$(l_1 \otimes l_x \otimes l_y, l_3)$*-equivariant.*

2) The columns of T *are orthogonal.*

3) The columns of T *have the same* L_2*-norms.*

Proof *We first prove 1), given an arbitrary group element* $g \in O(3)$*, we have that*

$$
\begin{aligned}
&\rho_{l_1 \otimes l_x \otimes l_y}(g) \odot T \\
=&\rho_{l_1 \otimes l_x \otimes l_y}(g) \odot \left(\sum_{j_1} \hat{C}^{l_x l_y l_2}_{i_2 j_1} C^{l_1 l_2 l_3}_{i_1 j_1 i_3} \right) \\
=&\sum_{i'_1, i'_2} R^{l_1}_{i_1 i'_1} R^{l_x \otimes l_y}_{i_2 i'_2} \left(\sum_{j_1} \hat{C}^{l_x l_y l_2}_{i'_2 j_1} C^{l_1 l_2 l_3}_{i'_1 j_1 i_3} \right) \\
=&\sum_{j_1} \left(\sum_{i'_2} R^{l_x \otimes l_y}_{i_2 i'_2} \hat{C}^{l_x l_y l_2}_{i'_2 j_1} \right) \left(\sum_{i'_1} R^{l_1}_{i_1 i'_1} C^{l_1 l_2 l_3}_{i'_1 j_1 i_3} \right) \\
\overset{(a)}{=}&\sum_{j_1} \left(\sum_{j'_1} \hat{C}^{l_x l_y l_2}_{i_2 j'_1} R^{l_2}_{j'_1 j_1} \right) \left(\sum_{i'_1} R^{l_1}_{i_1 i'_1} C^{l_1 l_2 l_3}_{i'_1 j_1 i_3} \right) \\
=&\sum_{j_1} \hat{C}^{l_x l_y l_2}_{i_2 j_1} \left(\sum_{i'_1, j'_1} R^{l_2}_{j'_1 j_1} R^{l_1}_{i_1 i'_1} C^{l_1 l_2 l_3}_{i'_1 j'_1 i_3} \right) \\
\overset{(b)}{=}&\sum_{j_1, i'_3} \hat{C}^{l_x l_y l_2}_{i_2 j_1} C^{l_1 l_2 l_3}_{i_1 j_1 i'_3} R^{l_3}_{i'_3 i_3} \\
=&T \odot \rho_{l_3}(g), \tag{37}
\end{aligned}
$$

so T is $O(3)$-$(l_1 \otimes l_x \otimes l_y, l_3)$-equivariant, where R^l is the matrix representation in l space, equalities (a) and (b) hold true because of Lemma 8.

Next we prove 2), for an arbitrary pair of the indices of two columns $k \neq k'$, we have

$$\sum_j T_{jk} T_{jk'}$$

$$= \sum_{j_1,j_3} \left(\sum_{j_2} \hat{C}_{j_1 j_2}^{l_x l_y l_2} C_{j_3 j_2 k}^{l_1 l_2 l_3} \right) \left(\sum_{j_2'} \hat{C}_{j_1 j_2'}^{l_x l_y l_2} C_{j_3 j_2' k'}^{l_1 l_2 l_3} \right)$$

$$= \sum_{j_2,j_2'} \left(\sum_{j_1} \hat{C}_{j_1 j_2}^{l_x l_y l_2} \hat{C}_{j_1 j_2'}^{l_x l_y l_2} \right) \left(\sum_{j_3} C_{j_3 j_2 k}^{l_1 l_2 l_3} C_{j_3 j_2' k'}^{l_1 l_2 l_3} \right)$$

$$\overset{(a)}{=} \sum_{j_1,j_2} \left(\hat{C}_{j_1 j_2}^{l_x l_y l_2} \right)^2 \left(\sum_{j_3} C_{j_3 j_2 k}^{l_1 l_2 l_3} C_{j_3 j_2 k'}^{l_1 l_2 l_3} \right)$$

$$= \| \hat{C}^{l_x l_y l_2} \|_{col}^2 \left(\sum_{j_2,j_3} C_{j_3 j_2 k}^{l_1 l_2 l_3} C_{j_3 j_2 k'}^{l_1 l_2 l_3} \right)$$

$$\overset{(b)}{=} 0, \tag{38}$$

and thus the columns of T are orthogonal. Equalities (a) and (b) follow from Lemma 6 that $\sum_{j_1} \hat{C}_{j_1 j_2}^{l_x l_y l_2} \hat{C}_{j_1 j_2'}^{l_x l_y l_2} = 0$ for $j_2 \neq j_2'$, and $\sum_{j_2,j_3} C_{j_3 j_2 k}^{l_1 l_2 l_3} C_{j_3 j_2 k'}^{l_1 l_2 l_3} = 0$.

Finally, we prove 3), for an arbitrary column with index k,

$$\sqrt{\sum_j T_{jk} T_{jk}}$$

$$= \sqrt{\sum_{j_1,j_3} \left(\sum_{j_2} \hat{C}_{j_1 j_2}^{l_x l_y l_2} C_{j_3 j_2 k}^{l_1 l_2 l_3} \right) \left(\sum_{j_2'} \hat{C}_{j_1 j_2'}^{l_x l_y l_2} C_{j_3 j_2' k}^{l_1 l_2 l_3} \right)}$$

$$= \sqrt{\sum_{j_2,j_2'} \left(\sum_{j_1} \hat{C}_{j_1 j_2}^{l_x l_y l_2} \hat{C}_{j_1 j_2'}^{l_x l_y l_2} \right) \left(\sum_{j_3} C_{j_3 j_2 k}^{l_1 l_2 l_3} C_{j_3 j_2' k}^{l_1 l_2 l_3} \right)}$$

$$= \sqrt{\sum_{j_2} \left(\sum_{j_1} \left(\hat{C}_{j_1 j_2}^{l_x l_y l_2} \right)^2 \sum_{j_3} \left(C_{j_3 j_2 k}^{l_1 l_2 l_3} \right)^2 \right)}$$

$$= \| \hat{C}^{l_x l_y l_2} \|_{col} \sqrt{\sum_{j_2,j_3} \left(C_{j_3 j_2 k}^{l_1 l_2 l_3} \right)^2}$$

$$= \| \hat{C}^{l_x l_y l_2} \|_{col} \| C^{l_1 l_2 l_3} \|_{col}. \tag{39}$$

From Lemma 7, we know that it takes the same value for different k, which satisfies 3).

■

Lemma 15 *The columns of different path matrices of the same path length are orthogonal. Specifically, given two path matrices $P^{(p)}$ and $P^{(p')}$ with different paths p and p', it holds that*

$$\sum_j P^{(p)}_{ji_1} P^{(p')}_{ji_2} = 0.$$

Proof *We prove by induction. The first to third columns of the Parentage Scheme generate the $(0 \to 1 \to 2)$, $(0 \to 1 \to 1)$, and $(0 \to 1 \to 0)$ path matrices. We first demonstrate that the columns of these matrices are orthogonal. Since they originate from the contraction of $(0 \to 1)$ path matrices and different CG tensors, we can establish a more general conclusion. Specifically, for any path matrix P and two distinct CG tensors C and C', given column indices k and k' of these two tensors, consider the dot product of these two columns. We have*

$$
\sum_{j_1, j_3} \left(\sum_{j_2} P_{j_1 j_2} C_{j_3 j_2 k} \right) \left(\sum_{j_2'} P_{j_1 j_2'} C'_{j_3 j_2' k'} \right)
$$

$$
= \sum_{j_2, j_2'} \left(\sum_{j_1} P_{j_1 j_2} P_{j_1 j_2'} \right) \left(\sum_{j_3} C_{j_3 j_2 k} C'_{j_3 j_2' k'} \right)
$$

$$
\stackrel{(a)}{=} \|P\|^2_{col} \underbrace{\left(\sum_{j_2, j_3} C_{j_3 j_2 k} C'_{j_3 j_2 k'} \right)}_{0}
$$

$$
\stackrel{(b)}{=} 0. \tag{40}
$$

Equations (a) and (b) hold by Lemma 6 in the general case. This general condition applies to the columns of path matrices of length 3 as the initial condition. Now, we assume that the columns of path matrices of the same length are mutually orthogonal and further prove that when the length is increased by 1, all path matrices continue to have orthogonal columns. Given two distinct path matrices P and P' (the case of identical paths is covered above) and two CG tensors C and C', we obtain

$$
\sum_{j_1, j_3} \left(\sum_{j_2} P_{j_1 j_2} C_{j_3 j_2 k} \right) \left(\sum_{j_2'} P'_{j_1 j_2'} C'_{j_3 j_2' k'} \right)
$$

$$
= \sum_{j_2, j_2'} \underbrace{\left(\sum_{j_1} P_{j_1 j_2} P'_{j_1 j_2'} \right)}_{0} \left(\sum_{j_3} C_{j_3 j_2 k} C'_{j_3 j_2' k'} \right)
$$

$$=0. \tag{41}$$

The final equality holds due to our assumption. Therefore, the columns of different path matrices of the same path length are orthogonal. ∎

Appendix B. Complexity Analysis

For the complexity with respect to the dimension, our ICT decomposition has polynomial complexity, while Bonvicini (2024) has $\mathcal{O}\left((\log d)!\right)$ complexity, which is higher than any polynomial complexity. Here is the derivation. Let d denotes the dimension and recall that Stirling's approximation tells us:

$$
\begin{aligned}
&(\log d)! \\
&\approx (\log d)^{\log d} \exp\left(-\log d\right) \sqrt{2\pi \log d} \\
&> (\log d)^{\log d} \exp\left(-\log d\right) \\
&= \exp\left(\log\left((\log d)^{\log d}\right) - \log d\right) \\
&= \exp\left(\log d \log\left(\log d\right) - \log d\right).
\end{aligned}
\tag{42}
$$

Take the logarithm of this equation, we obtain

$$
\begin{aligned}
&\log\left((\log d)!\right) \\
&= \log d \log\left(\log d\right) - \log d \\
&= \log d \left(\log\left(\log d\right) - 1\right).
\end{aligned}
\tag{43}
$$

On the other hand, taking the logarithm of an arbitrary polynomial d^k, where k is constant, gives us

$$\log\left(d^k\right) = k \log d. \tag{44}$$

Consequently, the ratio of equations (43) and (44) is

$$
\begin{aligned}
&\frac{\log\left((\log d)!\right)}{\log\left(d^k\right)} \\
&= \frac{\log d \left(\log\left(\log d\right) - 1\right)}{k \log d} \\
&= \frac{\left(\log\left(\log d\right) - 1\right)}{k}.
\end{aligned}
\tag{45}
$$

As d increases, $\frac{\log((\log d)!)}{\log(d^k)}$ must be eventually higher than 1, thus $\mathcal{O}\left((\log d)!\right)$ grows faster than any polynomial.

Appendix C. An Example for Generating Rank-2 ICT Decomposition Matrices

Since we only focus on the rank-2 ICT decomposition, it suffices to consider the portion of Figure 2 with ranks and weights in $\{0, 1, 2\}$. As in Algorithm 1, we first initialize (0)-path matrices $P^{(p_0)} = C^{000}$, and flatten the first two indices. Maintain the path $p_0 = (0)$, the current column number of the parentage scheme $i = 0$, and the current weight $l_0 = 0$. Then, we iterate Step 2 for those available weights in the $(i+1)$-th column, i.e., weights $|l_i - 1| \leq l_{i+1} \leq l_i + 1$ for $l_i \neq 0$. We then update the path by adding $l+1$ to the end of path p_i to obtain p_{i+1}, and calculating the contraction,

$$P^{(p_{i+1})} = \text{flatten}_2(P^{(p_i)} \odot C^{1l_i l_{i+1}}).$$

Then, we update the column number $i \leftarrow i + 1$. Specifically, since we initialize that $i = 0$, we have $i + 1 = 1$, following the rule that weights l_{i+1} must satisfy $|l_i - 1| \leq l_{i+1} \leq l_i + 1$, the only possible l_{i+1} is 1. Thus, we have

$$P^{(0 \to 1)} = \text{flatten}_2(P^{(0)} \odot C^{101}).$$

Then, we have $l_1 = 1$, and l_2 has three choices, 0, 1, and 2. When $l_2 = 2$, we have

$$P^{(0 \to 1 \to 2)} = \text{flatten}_2(P^{(0 \to 1)} \odot C^{112});$$

When $l_2 = 1$, we have

$$P^{(0 \to 1 \to 1)} = \text{flatten}_2(P^{(0 \to 1)} \odot C^{111});$$

When $l_2 = 0$, we have

$$P^{(0 \to 1 \to 0)} = \text{flatten}_2(P^{(0 \to 1)} \odot C^{110}).$$

Then, we normalize the above three path matrices,

$$\hat{P}_{*,c}^{(0 \to 1 \to 2)} = \frac{P_{*,c}^{(0 \to 1 \to 2)}}{\|P_{*,c}^{(0 \to 1 \to 2)}\|_2}, \quad \hat{P}_{*,c}^{(0 \to 1 \to 1)} = \frac{P_{*,c}^{(0 \to 1 \to 1)}}{\|P_{*,c}^{(0 \to 1 \to 1)}\|_2}, \quad \hat{P}_{*,c}^{(0 \to 1 \to 0)} = \frac{P_{*,c}^{(0 \to 1 \to 0)}}{\|P_{*,c}^{(0 \to 1 \to 0)}\|_2}.$$

*The matrix multiplications then give us all three rank-2 ICT decomposition matrices,*

$$H^{(2;2;1)} = \hat{P}^{(0 \to 1 \to 2)} \cdot (\hat{P}^{(0 \to 1 \to 2)})^\top,$$

$$H^{(2;1;1)} = \hat{P}^{(0 \to 1 \to 1)} \cdot (\hat{P}^{(0 \to 1 \to 1)})^\top,$$

$$H^{(2;0;1)} = \hat{P}^{(0 \to 1 \to 0)} \cdot (\hat{P}^{(0 \to 1 \to 0)})^\top.$$

Appendix D. Visualization of Decomposition Matrices

$$\begin{bmatrix}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.50 & 0.00 & -0.50 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.50 & 0.00 & 0.00 & 0.00 & -0.50 & 0.00 & 0.00 \\
0.00 & -0.50 & 0.00 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & -0.50 & 0.00 \\
0.00 & 0.00 & -0.50 & 0.00 & 0.00 & 0.00 & 0.50 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.50 & 0.00 & 0.50 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00
\end{bmatrix}$$

Figure 6: Decomposition matrix for rank-2 ICT with path $(0 \rightarrow 1 \rightarrow 1)$.
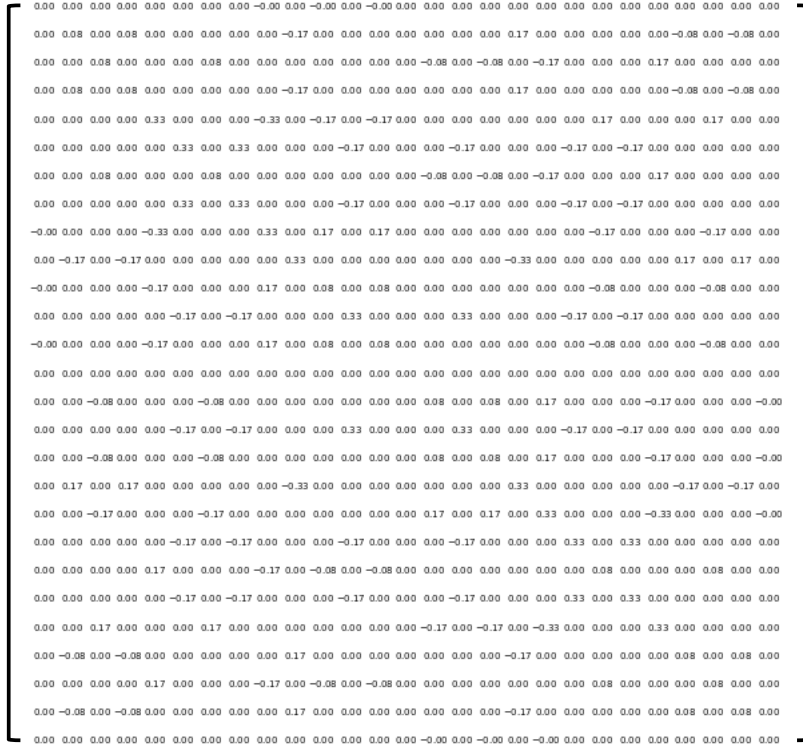
Figure 7: Decomposition matrix for rank-3 ICT with path $(0 \rightarrow 1 \rightarrow 2 \rightarrow 2)$.
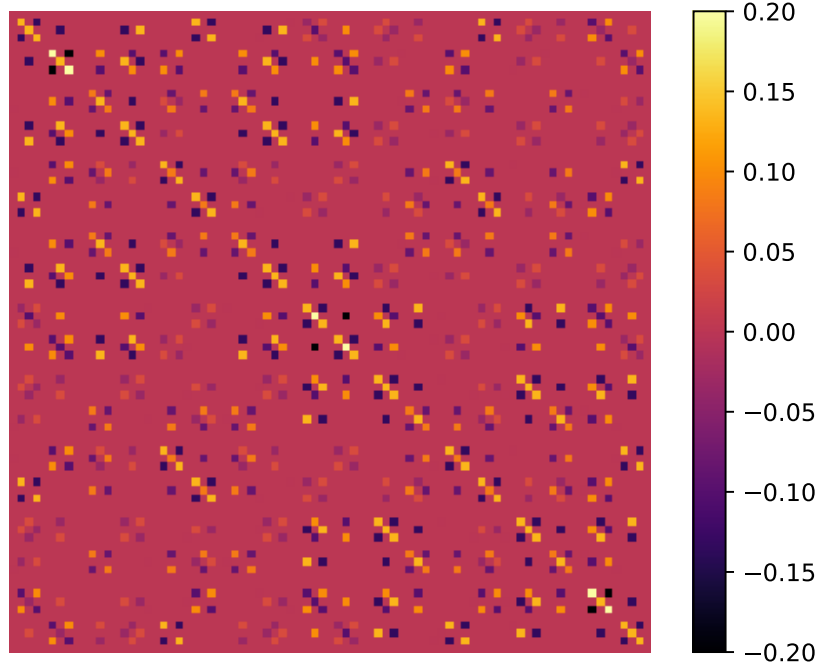
Figure 8: Decomposition matrix for rank-4 ICT with path $(0 \rightarrow 1 \rightarrow 1 \rightarrow 2 \rightarrow 3)$.
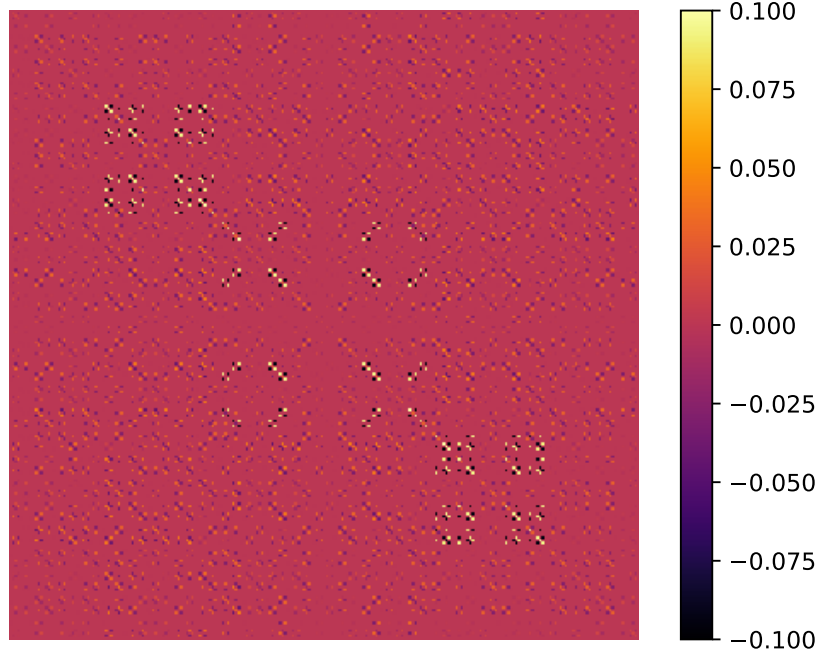
Figure 9: Decomposition matrix for rank-5 ICT with path $(0 \rightarrow 1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 3)$.
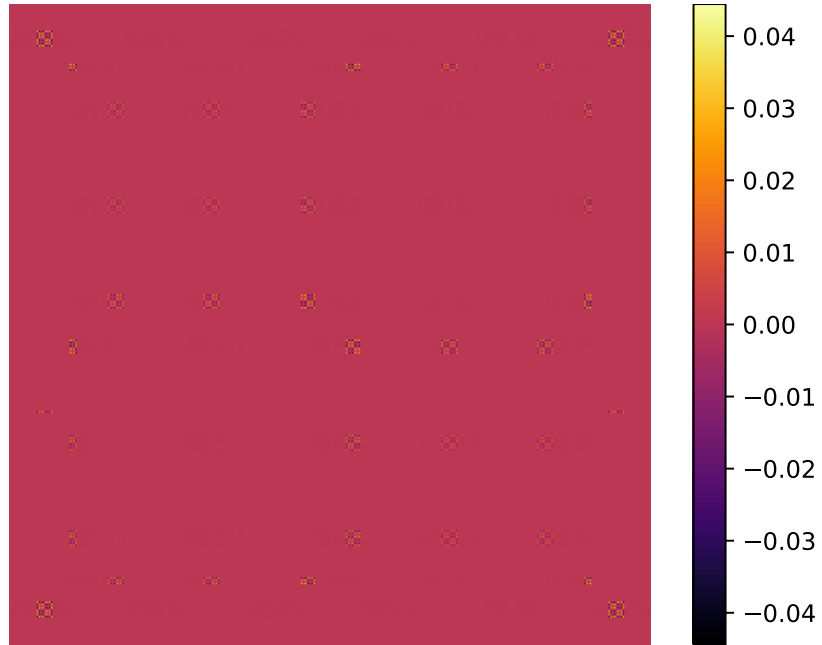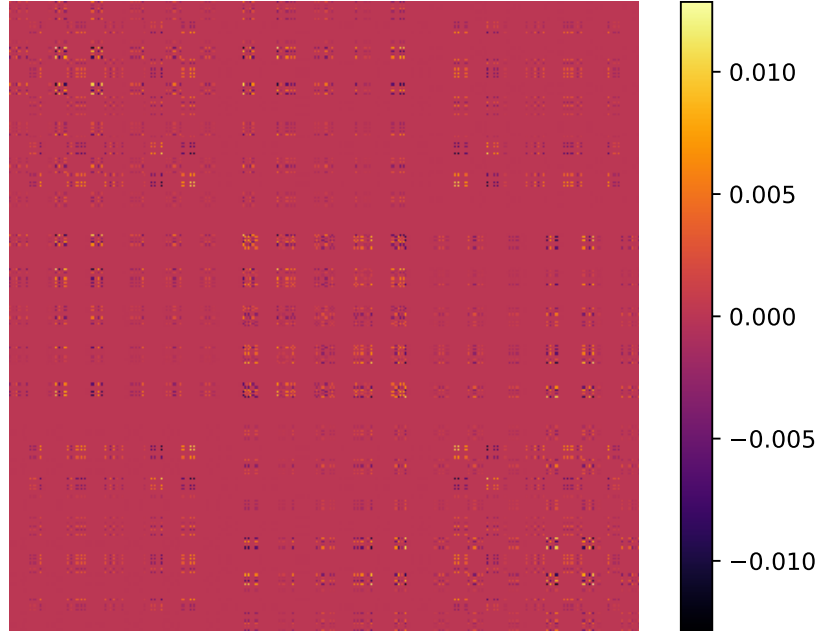

Figure 10: Decomposition matrix for rank-6 ICT with path $(0 \rightarrow 1 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 1)$.

Figure 11: Decomposition matrix for rank-7 ICT with path $(0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 2 \rightarrow 1 \rightarrow 1)$.
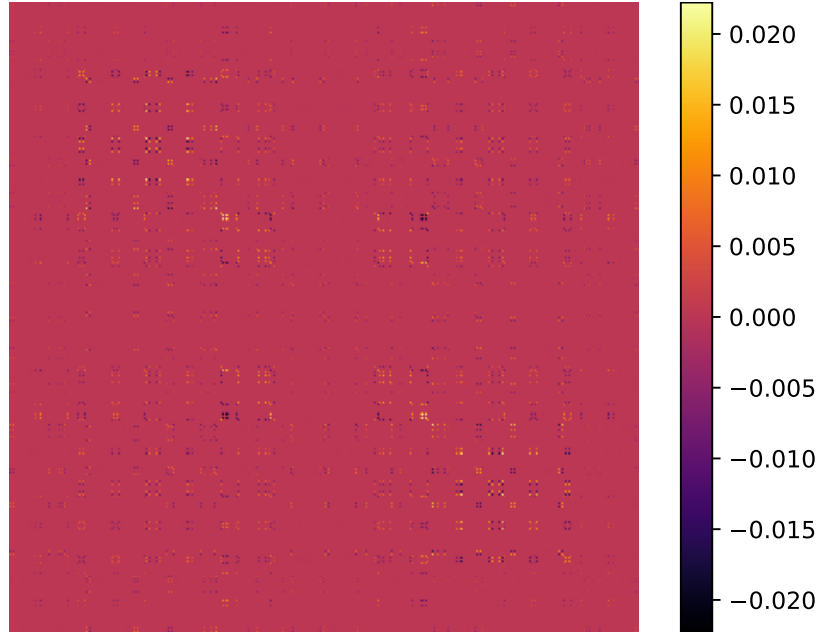

Figure 12: Decomposition matrix for rank-7 ICT with path $(0 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 2)$.
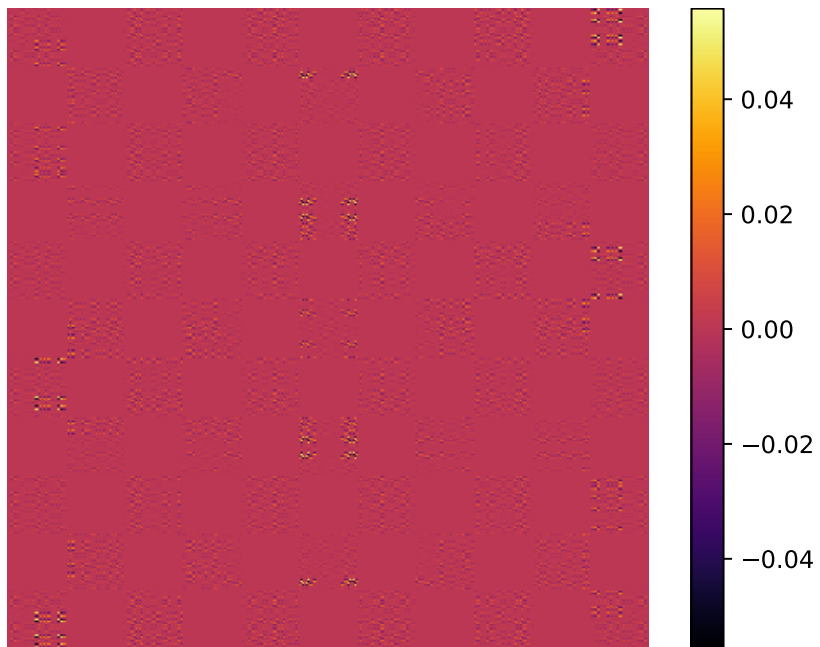
Figure 13: Equivariant basis matrix of $\mathrm{End}_{O(3)}\left(\left(\mathbb{R}^3\right)^{\otimes 6}\right)$ with paths $(1 \to 2 \to 2 \to 3 \to 3 \to 3)$ and $(1 \to 1 \to 2 \to 3 \to 4 \to 3)$.
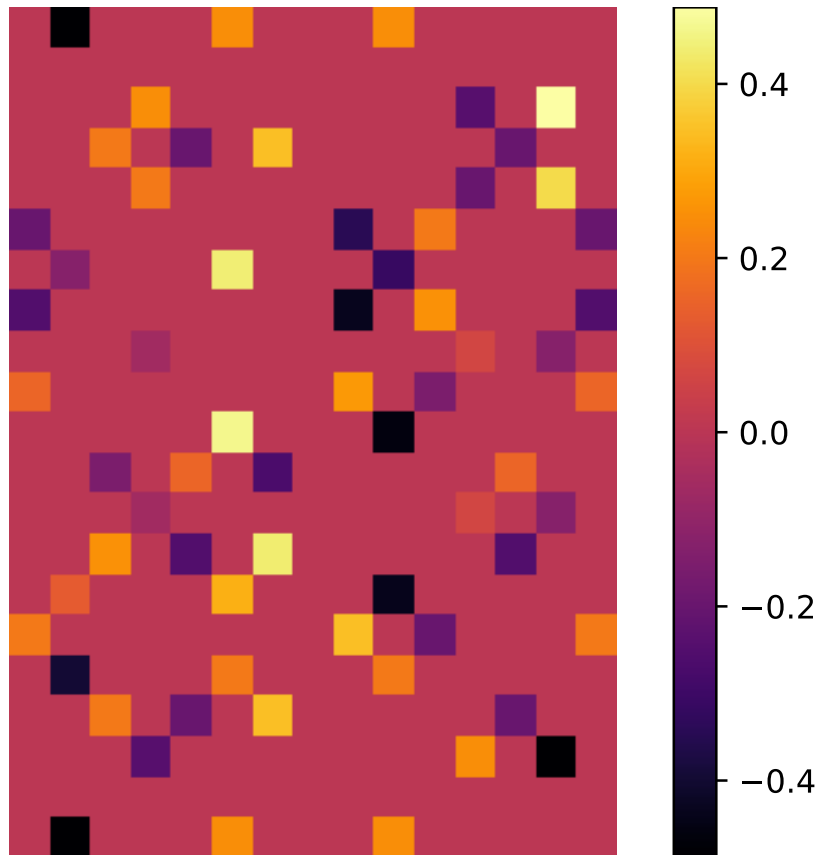
Figure 14: Equivariant basis matrix of $\mathrm{Hom}_{O(3)}\left(\left(l=1 \otimes l=2\right), \left(l=1 \otimes l=3\right)\right)$ with paths $(1 \xrightarrow{2} 2)$ and $(1 \xrightarrow{3} 2)$.
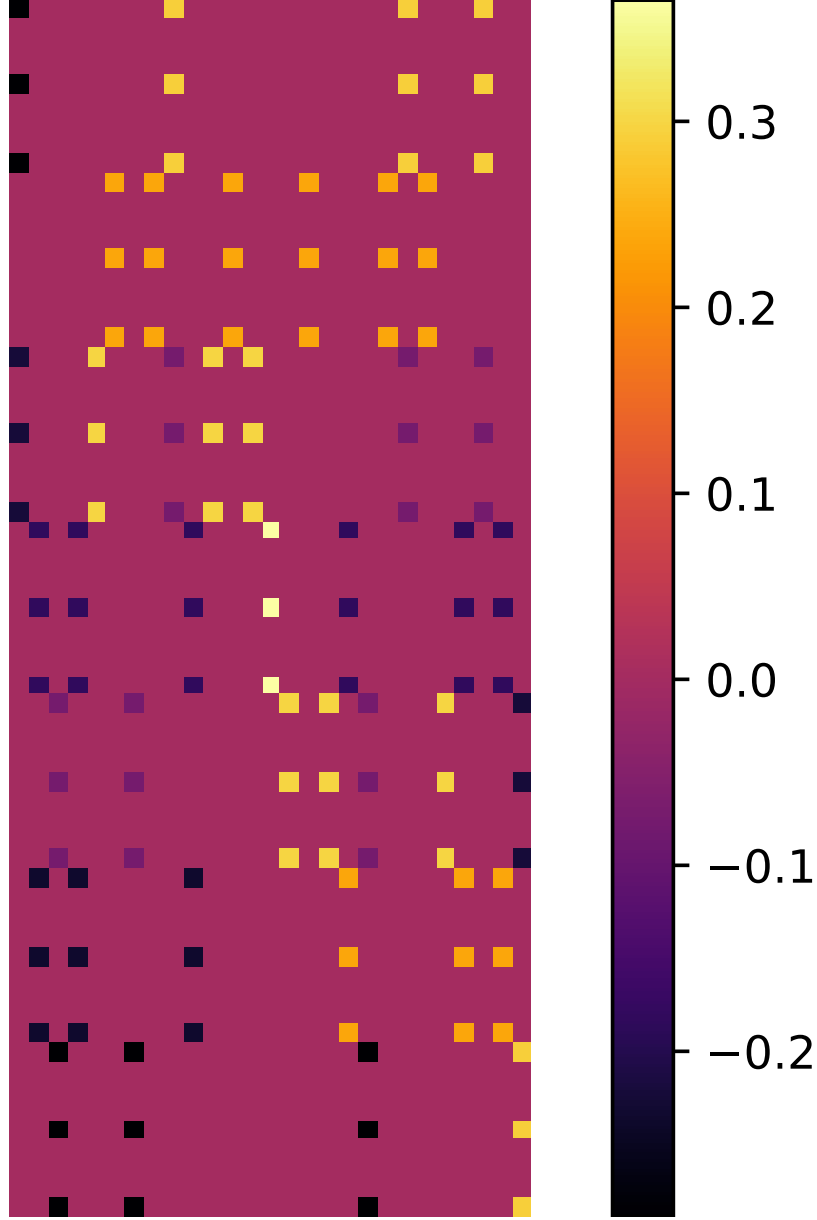
Figure 15: Equivariant basis matrix of $\mathrm{Hom}_{SU(2)}\left((0.5 \otimes 0.5 \otimes 1.5), (0.5 \otimes 0.5 \otimes 0.5)\right)$ with paths $(0.5 \xrightarrow{0.5} 0 \xrightarrow{1.5} 1.5)$ and $(0.5 \xrightarrow{0.5} 1 \xrightarrow{0.5} 1.5)$.

References

Arne Alex, Matthias Kalus, Alan Huckleberry, and Jan von Delft. *A numerical algorithm for the explicit calculation of $SU(N)$ and $SL(N,C)$ Clebsch–Gordan coefficients.* Journal of Mathematical Physics, *52(2):023507, 2011.*

David L. Andrews. *Symmetry-based identification and enumeration of independent tensor properties in nonlinear and chiral optics.* The Journal of Chemical Physics, *158(3):034101, 2023.*

DL Andrews and WA Ghoul. *Irreducible fourth-rank Cartesian tensors.* Physical Review A, *25(5):2647, 1982.*

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. *MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In* Advances in Neural Information Processing Systems (NeurIPS), *volume 35, pages 11423–11436, 2022.*

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.*

Andrea Bonvicini. *Irreducible Cartesian tensor decomposition: A computational approach.* The Journal of Chemical Physics, *160(22):224105, 2024.*

Richard Brauer. *On algebras which are connected with the semisimple continuous groups.* Annals of Mathematics, *38(4):857–872, 1937.*

Theodor Bröcker and Tammo Tom Dieck. Representations of compact Lie groups, *volume 98.* Springer Science & Business Media, 2013.

Bingqing Cheng. *Cartesian atomic cluster expansion for machine learning interatomic potentials.* npj Computational Materials, *10(1):157, 2024.*

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

Taco Cohen and Max Welling. *Group equivariant convolutional networks. In* International Conference on Machine Learning (ICML), *pages 2990–2999, 2016.*

JAR Coope, RF Snider, and FR McCourt. *Irreducible Cartesian tensors.* The Journal of Chemical Physics, *43(7):2269–2275, 1965.*

Çiğdem Dinçkal. *Orthonormal decomposition of third rank tensors and applications. In* Proceedings of the World Congress on Engineering, *volume 1, 2013.*

Ralf Drautz. *Atomic cluster expansion for accurate and transferable interatomic potentials.* Physical Review B, *99(1):014104, 2019.*

Marc Finzi, Max Welling, and Andrew Gordon Wilson. *A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In* International Conference on Machine Learning (ICML)*, pages 3318–3328, 2021.*

Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. *Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations.* Transactions on Machine Learning Research (TMLR)*, 2023.*

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. *Directional message passing for molecular graphs. In* International Conference on Learning Representations (ICLR)*, 2020.*

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. *Gemnet: Universal directional graph neural networks for molecules. In* Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 6790–6802, 2021.*

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. *Neural message passing for quantum chemistry. In* International Conference on Machine Learning (ICML)*, pages 1263–1272, 2017.*

Marco Gori, Gabriele Monfardini, and Franco Scarselli. *A new model for learning in graph domains. In* International Joint Conference on Neural Networks (IJCNN)*, volume 2, pages 729–734, 2005.*

Teturo Inui, Yukito Tanabe, and Yositaka Onodera. *Group theory and its applications in physics, volume 78. Springer Science & Business Media, 2012.*

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. *Clebsch–Gordan nets: a fully Fourier space spherical convolutional neural network.* Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.*

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. *Backpropagation applied to handwritten zip code recognition.* Neural Computation*, 1(4):541–551, 1989.*

Yi-Lun Liao, Brandon M Wood, Abhishek Das, and Tess Smidt. *Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In* International Conference on Learning Representations (ICLR)*, 2023.*

George Marsaglia and George PH Styan. *Equalities and inequalities for ranks of matrices.* Linear and Multilinear Algebra*, 2(3):269–292, 1974.*

V V Mihailov. *Addition or arbitrary number of identical angular momenta.* Journal of Physics A: Mathematical and General*, 10(2):147, 1977.*

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.*

Edward Pearce-Crump. *An algorithm for computing with Brauer's group equivariant neural network layers.* arXiv preprint arXiv:2304.14165, 2023a.

Edward Pearce-Crump. *Brauer's group equivariant neural networks. In* International Conference on Machine Learning (ICML), *pages 27461–27482, 2023b.*

Giulio Racah. *Theory of complex spectra. II.* Physical Review, *62:438–462, 1942.*

Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. *Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17.* Journal of Chemical Information and Modeling, *52(11):2864–2875, 2012.*

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. *Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In* Advances in Neural Information Processing Systems (NeurIPS), *volume 30, 2017.*

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 (ICML), *pages 9377–9388, 2021.*

Guillem Simeon and Gianni De Fabritiis. *Tensornet: Cartesian tensor representations for efficient learning of molecular potentials. In* Advances in Neural Information Processing Systems (NeurIPS), *volume 36, 2024.*

Guillem Simeon, Antonio Mirarchi, Raul P Pelaez, Raimondas Galvelis, and Gianni De Fabritiis. *On the inclusion of charge and spin states in Cartesian tensor neural network potentials.* arXiv preprint arXiv:2403.15073, 2024.

Robert F Snider. *Irreducible Cartesian Tensors, volume 43.* Walter de Gruyter GmbH & Co KG, 2017.

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.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. *Equivariant diffusion policy. In* The Conference on Robot Learning (CoRL), *2024.*

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. *3D steerable CNNs: Learning rotationally equivariant features in volumetric data.* Advances in Neural Information Processing Systems (NeurIPS), *31, 2018.*

Viktor Zaverkin, Francesco Alesiani, Takashi Maruyama, Federico Errica, Henrik Christiansen, Makoto Takamoto, Nicolas Weber, and Mathias Niepert. *Higher-rank irreducible cartesian tensors for equivariant message passing.* arXiv preprint arXiv:2405.14253, 2024.

W-N Zou, Q-S Zheng, D-X Du, and J Rychlewski. *Orthogonal irreducible decompositions of tensors of high orders.* Mathematics and Mechanics of Solids, *6(3):249–267, 2001.*