

Unifying O(3) Equivariant Neural Networks Design with Tensor-Network Formalism

Zimu Li^{1,*}, Zihan Pengmei^{2,*}, Han Zheng^{4,*}, Erik Thiede³, Junyu Liu^{5,6,7,+}, and Risi Kondor⁴

¹DAMTP, Center for Mathematical Sciences, University of Cambridge, Cambridge CB30WA, UK

²Department of Chemistry, The University of Chicago, Chicago, IL 60637, USA

³Center for Computational Mathematics, Flatiron Institute, New York NY 10010

⁴Department of Computer Science, The University of Chicago, Chicago, IL 60637, USA

⁵Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA

⁶Kadanoff Center for Theoretical Physics, The University of Chicago, Chicago, IL 60637, USA

⁷SeQure, Chicago, IL 60615, USA

+ junyuliu@uchicago.edu

*Contributed equally

ABSTRACT

Many learning tasks, including learning potential energy surfaces from ab initio calculations, involve global spatial symmetries and permutational symmetry between atoms or general particles. Equivariant graph neural networks are a standard approach to such problems, with one of the most successful methods employing tensor products between various tensors that transform under the spatial group. However, as the number of different tensors and the complexity of relationships between them increase, maintaining parsimony and equivariance becomes increasingly challenging. In this paper, we propose using fusion diagrams, a technique widely employed in simulating SU(2)-symmetric quantum many-body problems, to design new spatial equivariant components for neural networks. This results in a diagrammatic approach to constructing novel neural network architectures. When applied to particles within a given local neighborhood, the resulting components, which we term "fusion blocks," serve as universal approximators of any continuous equivariant function defined on the neighborhood. We incorporate a fusion block into pre-existing equivariant architectures (Cormorant and MACE), leading to improved performance with fewer parameters on a range of challenging chemical problems. Furthermore, we apply group-equivariant neural networks to study non-adiabatic molecular dynamics of stilbene cis-trans isomerization. Our approach, which combines tensor networks with equivariant neural networks, suggests a potentially fruitful direction for designing more expressive equivariant neural networks.

Introduction

Graph neural networks (GNNs) have recently gained prominence in the field of chemistry, owing to their ability to learn from the structural properties of molecules and materials. Nevertheless, devising an efficient and accurate GNN architecture for investigating dynamic properties of chemical systems remains a formidable challenge. GNNs are adept at learning the structure of chemical systems and predicting their properties, including potential energy, dipole moment, and atomic forces. Recently, there has been a surge of interest in employing deep learning to forecast chemical properties and expedite first-principles dynamics simulations^{1–5}. Specifically, GNNs have been utilized to estimate the potential energy with distinct atomic coordinates, where the negative gradient concerning the input coordinates naturally corresponds to the atomic force. Accurate prediction of potential energy and atomic force⁶ necessitates adherence to spatial symmetries, such as translational and rotational covariance, since these properties are continuous functions defined on three-dimensional Euclidean space.

Machine learning algorithms employed to predict properties such as potential energy and atomic forces must yield consistent results, regardless of the molecule's rotational pose or ordering. To address this challenge, researchers have developed group-equivariant neural networks that preserve these symmetries^{5,7–10}. In a group-equivariant network, symmetry operations on the data, including rotations of pictures and molecules, and permutations of the labels of each particle, commute with the network's layers, ensuring that the same physical property is predicted irrespective of the input's orientation. Many state-of-the-art spatially equivariant neural networks^{11,12} leverage the representation theory of the spatial rotation group in the so-called Fourier space¹³. These Fourier space methods employ the Clebsch-Gordan nonlinearities^{13,14}. In fact, as elucidated in the supplementary material (SM), Clebsch-Gordan nonlinearities are the sole source of nonlinearity in Fourier space determined by invariant theory in mathematics^{15–17}. The Clebsch-Gordan nonlinearities further constrain the use of linear weights, which can only act on the multiplicity space corresponding to each irreducible representation¹⁴.

Independently of work in machine learning, physicists have been using network models, called *tensor networks*, to represent complicated quantum many-body systems. Tensor networks are a family of methods for approximating larger tensors by contracting together a large collection of smaller tensors. Tensor networks have been used to successfully approximate large quantum states with low entanglement accurately by making use of the density matrix renormalization group (DMRG) in one dimension^{18,19} and introducing low-rank tensors to represent the quantum states²⁰. Applications of tensor networks include quantum simulation of quantum physics problems^{21–23}, quantum computing and quantum supremacy experiments^{24,25}, machine learning and data science^{26–28}, and quantum gravity^{29,30}. A special type of tensor networks concerns with global on-site SU(2) symmetry, called the *spin networks*, where multiple sites are fused by a prescribed fusion diagrams^{31,32} so that the global wavefunctions are SU(2) symmetric. The fusion diagrams, as we will show later and further in SM, are natural and sparse generalization of Clebsch-Gordan products among multiple sites. Fusion diagrams have found great success in simulate SU(2)-symmetric quantum systems^{20,32–34}, and we will show their potential for constructing universal equivariant neural networks.

Fusion diagrams facilitate the classification of existing neural network architectures and inspire the development of novel equivariant blocks. We showcase the computational power of these blocks using classical results from invariant theory, which establish that under certain conditions, they can achieve universality. For instance, we employ fusion diagrams to construct a new SO(3)-equivariant block, which we incorporate into two state-of-the-art neural network architectures: Cormorant³⁵ and MACE¹². We demonstrate that integrating the new equivariant layer significantly enhances the performance of both architectures, with a comparable or substantially fewer number of parameters.

To assess the validity of the fusion block, we carried out extensive experiments on various chemical systems, including standard benchmark datasets such as QM-9³⁶ and MD-17⁶, which aims to predict the quantum properties of molecules and potential energy surfaces, as well as more challenging systems like the non-adiabatic cis-trans isomerization of stilbene. Non-adiabatic isomerization of stilbene poses a considerable challenge learning the multiple boarder and reactive potential energy surfaces (PESs), necessitating accurate interpolation and extrapolation.

In summary, this paper presents a novel method for constructing group-equivariant neural network blocks using fusion diagrams, a concept borrowed from theoretical physics. Our approach alleviates the combinatorial complexity associated with preserving symmetry constraints in neural networks, enabling the construction of expressive and universal equivariant layers. We demonstrate the effectiveness of the fusion block by incorporating our new SO(3)-equivariant layer into two state-of-the-art molecular neural network architectures, Cormorant and MACE, and evaluating them on a variety of common benchmarks in companion with more complicated molecular isomerization and adsorption process. Our results indicate that the fusion block leads to improved performance with comparable or fewer parameters. Overall, our approach contributes to the developing a new routine that can be used to construct more expressive group equivariant neural networks.

Background

Before delving into the specifics of our approach, it is crucial to lay the groundwork with some foundational concepts. In this section, we offer an overview of relevant ideas from both machine learning and physics. We begin with a concise review of molecular dynamics and the significance of symmetry and equivariance in machine learning. Subsequently, we introduce the concept of tensor products and their role in theoretical physics, including a description of the fusion diagram notation.

Molecular dynamics

Molecular dynamics simulations are essential tools for studying molecular properties at the atomic level within specific timescales. To simulate atomic motion, we need to calculate the potential energy and atomic forces acting on molecules with particular geometric configurations in \mathbb{R}^3 space. Generally, potential energy and its gradients can be accurately determined by electronic structure calculations from first principles or approximated classically as simple analytical potential functions within specific chemical environments, such as atomic type, bond length, and bond angle. However, the electronic structure calculations under the ab initio molecular dynamics (AIMD) calculations are expensive.

One popular approach to overcoming this limitation is to use neural networks as interatomic potentials^{1,2}, which are trained with reference AIMD data. Training neural networks as interatomic potentials involves regressing on potential energy and atomic forces simultaneously, where predictive forces can be naturally achieved as the negative gradient of energy via back-propagation. The potential energy is invariant to 3D rigid rotations, while atomic forces are covariant to rotations, as they are vector values. The equivariant neural networks that we introduce subsequently are a powerful data-driven approach for an accurate representation of the chemical environment.

Representation theory of SU(2) and SO(3)

Rotationally equivariant nets are arguably one of the most successful types of equivariant neural networks. Let X and Y be the input and output spaces of a layer L , and let T and T' be linear actions of a group \mathbb{G} encoding the symmetry on X resp. Y . The

layer is said to be equivariant to \mathbb{G} if

$$T'_g \circ L = L \circ T_g \quad \text{for all } g \in \mathbb{G}. \quad (1)$$

If the group action on the output space is the identity transformation, i.e. $T'_g y = y$ for all elements of \mathbb{G} , the above reduces to

$$L = L \circ T_g \quad \forall g \in \mathbb{G} \quad (2)$$

and we have an *invariant* layer. Constructing an equivariant neural network requires that both the learned affine function and the fixed nonlinear function obey equivariance. Kondor et al.¹⁴ showed how to construct learned affine functions that are equivariant to compact groups (such as the group of rotations or the group of permutations) using the theory of linear representations. A linear representation¹ of a compact group G is pair (V, ρ) such that for each $g \in G$, $\rho(g)$ is assigned a linear transformation of V for which $\forall g_1, g_2 \in G, \rho(g_1)\rho(g_2) = \rho(g_1g_2)$. If the representation is finite-dimensional, the range of ρ is a subset of the space of complex $k \times k$ matrices for some k . An irreducible linear representation (irrep) is a representation where V has no proper subspaces preserved under ρ . Using well-known results in representation theory, we can apply a linear transformation that decomposes the inputs, outputs, and activations of a neural network into components that transform according to the group's irreps. Then, one can show that the most general possible equivariant linear transformation can be written as matrix multiplication against each component^{14,37–39}. The construction of linear equivariant layers where inputs and outputs transform according to linear group representations has been widely studied and used in today's neural networks^{9,10,14,37,38,40–42}.

For the rest of this work, we will focus on equivariance in the presence of $SU(2)$ and $SO(3)$ symmetries. These groups have fundamental importance in modern quantum physics and machine learning applications on geometric data. The irreps of $SU(2)$ can be indexed by a single non-negative integer or half-integer, called the spin label. For any $g \in SU(2)$ and spin label j , we denote the corresponding matrix that arises from evaluating $\rho(g)$ as $W^j(g)$. It is well known in group theory that $SO(3)$ irreps are isomorphic to the irreps of $SU(2)$ with integer spin labels^{17,43,44}. This relationship allows us to study both $SO(3)$ and $SU(2)$ at the same time. Depending on the mathematical context, vectors in V_j might transform either by $W^j(g)$ (contravariant transformation) or by its complex conjugate (covariant transformation). In what follows we focus only on irreps and omit ρ_j when we denote irreps. To distinguish these two cases, we denote components of any vector \mathbf{v} transforming contravariantly by raised index v^m with w_m being defined accordingly for the covariant case. With the notion of raised and lowered indices, one can contract vectors like $v^m w_m$, where the Einstein summation convention will be used consistently in this paper.

With the above basic notions clarified, let us formally define the Clebsch-Gordan product. Take two $SU(2)$ irreps (ρ_{j_a}, V_{j_a}) and (ρ_{j_b}, V_{j_b}) of spin j_a and j_b respectively. We can then define the tensor product representation $(\rho_{j_a} \otimes \rho_{j_b}, V_{j_a} \otimes V_{j_b})$. As this is still an $SU(2)$ representation, it can be decomposed into irreps labeled by spins. A *Clebsch-Gordan decomposition* is a matrix $C^{(j_a, j_b, j_c)}$ which transforms the tensor product $W^{j_a}(g) \otimes W^{j_b}(g)$ into $W^{j_c}(g)$ for a prescribed spin j_c and any $g \in SU(2)$. Formally,

$$C^{(j_a, j_b, j_c)\dagger} (W^{j_a}(g) \otimes W^{j_b}(g)) C^{(j_a, j_b, j_c)} = W^{j_c}(g) \quad \text{for all } g \in SU(2). \quad (3)$$

By definition, the Clebsch-Gordan decomposition is equivariant with respect to the action of $SU(2)$ as well as $SO(3)$. Formally, it can be understood as an element from the space of $SU(2)$ equivariant maps $\text{Hom}(V_{j_a} \otimes V_{j_b}, V_{j_c})$, where $V_{j_a}, V_{j_b}, V_{j_c}$ are the corresponding $SO(3)$ irreps with the angular momenta labels j_1, j_2, j_3 . In this case, we can write the Clebsch-Gordan product as a third-order tensor:

$$C^{(j_a, j_b, j_c)}{}_{m_1, m_2}{}^{m_3} \in \text{Hom}(V_{j_a} \otimes V_{j_b}, V_{j_c}), \quad (4)$$

where m_1, m_2, m_3 are the corresponding magnetic quantum numbers. There are well-established methods to compute $C^{(j_a, j_b, j_c)}$ both theoretically and algorithmically, e.g.,^{43,45,46}. Summing in Einstein notation with lower and upper indices, we call it a Clebsch-Gordan product for input vectors $\psi^{(j_a)}, \psi^{(j_b)}$:

$$\psi^{(j_c)m_c} = C^{(j_a, j_b, j_c)}{}_{m_a, m_b}{}^{m_c} \psi^{(j_a)m_a} \psi^{(j_b)m_b}. \quad (5)$$

We will also leave m as entry indices and write Clebsch-Gordan product among input vectors as inner products:

$$\psi^{(j_c)} = \langle C^{(j_a, j_b, j_c)}, \psi^{(j_a)} \otimes \psi^{(j_b)} \rangle \quad (6)$$

¹Linear representations should not be confused with the different use of the word “representation” in representation learning.

Input: Incoming atomic activations: $\{\Psi_1^{(\text{in})}, \dots, \Psi_n^{(\text{in})}\}$ and to initialize a FusionBlock module
Output: Outgoing atomic activations: $\Psi^{(\text{out})}$

for $Q_i^{(j,k;J)} \in \mathcal{Q}$ **do**

 apply incoming activations $\psi_1^{(\text{in}, j_1)}, \dots, \psi_n^{(\text{in}, j_n)}$. with an aggregation function ϕ to enforce permutation equivariance:

$$\tilde{\psi}_i^J = \phi(\text{FusionBlock}(\psi_1^{(\text{in}, j_1)} \dots \psi_n^{(\text{in}, j_n)})) \quad (7)$$

end for

Concatenate outputs for each fusion diagram along the new channel dimension $\tilde{\tau}_c$ and reshape.

$$\tilde{\Psi}_{\tilde{\tau}_c} = \tilde{\Psi}_1 \oplus \dots \oplus \tilde{\Psi}_n \quad (8)$$

Multiply by a linear mixing matrix.

$$\Psi_{\tilde{\tau}_c}^{(\text{out})} \leftarrow \tilde{\Psi}_{\tilde{\tau}_c} W_{\tilde{\tau}_c} \quad (9)$$

Algorithm 1: Fusion Block

Methods

Here we demonstrate how fusion diagrams can be used to design equivariant components that we call “neural fusion blocks.” We present an explicit construction for transformations under SU(2) and SO(3). In each block, we apply a collection of fusion diagrams \mathcal{Q} to our input tensors. Each incoming edge of the diagram is associated with an input to the block and each outgoing edge is associated with an output of the block. We denote the collection of input tensors associated with incoming edges as $\{\Psi_1^{(\text{in})}, \dots, \Psi_n^{(\text{in})}\}$ and components corresponding to the spin label j_1, \dots, j_n are denoted as $\{\psi_{\tau_1}^{(\text{in}, j_1)}, \dots, \psi_{\tau_n}^{(\text{in}, j_n)}\}$, where τ_1, \dots, τ_n are the channel dimension indices. We omit batch indices from our treatment for brevity. The fusion block then acts according to Algorithm 1. More illustrations on the updating rule with diagrammatic examples can be found in Section I in the SM. Due to the use of fusion diagrams, the resulting algorithm is guaranteed to be equivariant to rotations. It also serves an rotationally-equivariant universal approximator where we put the proof details in Section II in the SM.

As our focus is on the construction of specific components in an SO(3)-equivariant architecture rather than on proposing an entirely new architecture, we demonstrate the potential of our formalism by incorporating it into existing neural networks. Specifically, we choose to augment the Cormorant architecture proposed in³⁵ and the recent state-of-art model⁴⁷ with one additional three-body fusion block that replaces the conventional node-edge two-body interaction, with the aim of capturing inter-atomic interactions in a more faithful manner. Capturing three-body interactions in a SO(3) equivariant way with edge features could lead to a large overhead on computational resources. Applying fusion blocks to point clouds also requires ensuring that the resulting neural network obeys permutation symmetry. Since each fusion diagram has a single output, we can reinforce the permutation equivariance by passing these outputs through an aggregation function and incorporate them into existing message-passing-like mechanisms. It is worth mentioning that except employing Clebsch-Gordan products, there are other efficient architectures like using spherical coordinates of neighboring atoms and leveraging spherical harmonics to encode angular momentum information into higher dimensional representation of SO(3) and filtering through the spherical representations⁵.

Cormorant with Fusion Diagrams (CoFD)

Cormorant is one of the first equivariant neural networks that utilize the group equivariance and designed to learn molecular dynamics simulations and ground-state molecular properties. A neuron in Cormorant layer s operates as follows:

$$F_i^{s-1} = \left[\underbrace{F_i^s \oplus (F_i^{s-1} \otimes_{\text{cg}} F_i^{s-1})}_{\text{one-body part}} \oplus \underbrace{\left(\sum_j (Y(\mathbf{x}_{ij}) \otimes_{\text{cg}} F_j^{s-1}) g_{ij} \right)}_{\text{two-body part}} \right] \cdot W_{s,\ell}^{\text{vertex}}. \quad (10)$$

Here j sums over atom i ’s local neighborhood, and g_{ij} is a learned rotationally invariant function that takes F_i^s and F_j^s as input in addition to other rotationally invariant features. \otimes_{cg} denotes the channel-wise Clebsch-Gordan products and $Y(\mathbf{x}_{ij})$ the

spherical harmonics with input the relative displacement vectors between i th and j th atoms (we refer the reader to³⁵ for the precise functional form of g_{ij}). Each of these terms corresponds to the two-body diagram on the left below, while the product with g_{ij} is in a three-way product, it never has a covariant or contravariant component.

In particular, we observe that this layer has no equivariant interaction between equivariant parts of the activation atom i and the activation for atom j . Instead, their activations only interact through the rotationally invariant function g . Instead, in the present paper we employ our fusion diagrams to add an additional term to (10) that fully integrates all of the information between atom i and atom j . This corresponds to the fusion diagram on the right. The resulting fusion block has three inputs: the input activation for atom i , the input activation for atom j , and the collection of spherical harmonics evaluated on their relative displacements, and one output: a new feature for atom i . Consequently, we require a fusion diagram with three ingoing edges and one outgoing edge. Going from left to right, we input the representation of atom i , the representation of atom j , and the spherical harmonic representation of the edge connecting the two. We then incorporate this as a new term in (10), giving the following functional form for our modified Cormorant layer.

$$F_i^{s-1} = \left[F_i^s \oplus (F_i^{s-1} \otimes_{\text{cg}} F_i^{s-1}) \oplus \sum_i \left(\sum_j \left(Y(\mathbf{x}_{ij}) \otimes_{\text{cg}} F_j^{s-1} \right) g_{ij} \oplus F_{ij}^{\text{fusion}} \right) \right] \cdot W_{s,\ell}^{\text{vertex}}, \quad (11)$$

where F_{ij}^{fusion} is the output of the fusion block where inputs came from atom i and atom j within the coming legs chosen to be i th atom, j th atom, and their connecting edge. In other words, we use fusion diagrams to efficiently combine the atom-level messaging passing and edge-level message passing.

MACE with Fusion Diagrams (MoFD)

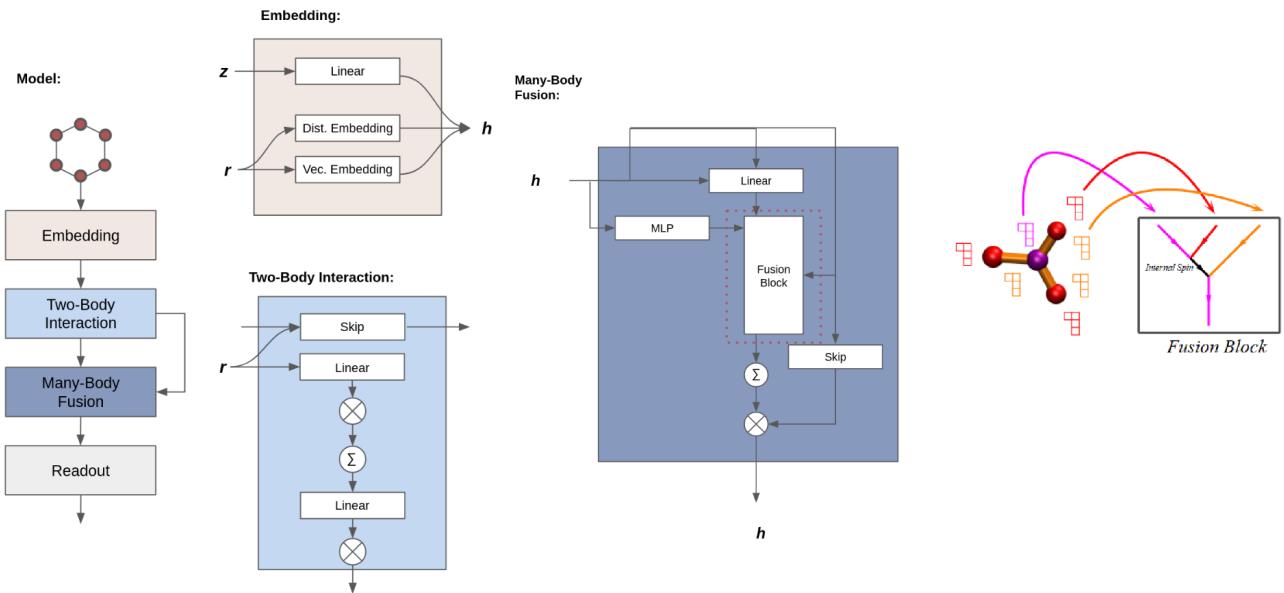


Figure 1. Schematic illustration of the implementation of fusion blocks in the MACE architecture. For each atom the fusion block first fuses all the neighboring atoms for a given radius cut-off by pre-selected fusion diagram templates. Specifically, for each neighboring atom, we fuse the information from the root, neighbor atom, and their connecting edge. Then the fusion block applies an aggregation method: in the present work, we simply sum all the neighbors.

Implementation of 3-body fusion blocks

In our modified MACE architecture we use fusion diagrams (Figure 1), a local neighborhood is defined by a cut-off radius r , the information on the central particle is $\Psi_o^{(\text{in},j)}$, adjacent particle $\Psi_i^{(\text{in},j)}$; $i \in \mathcal{N}(o)$, and the incident edges $\Phi_{(o,i)}^{(\text{in},j)}$. In particular, this information is passed to a l -element sequence of linearly independent 3-body fusion interactions given by a sequence of different internal spin configurations $\{k_1, k_2, \dots, k_l\}$, with the l -element sequence of outgoing activation on the center particle:

$$\Psi_{o,k_\alpha}^{(\text{out},j_d)} := \sum_{i \in \mathcal{N}(o)} \text{FusionBlock} \left(\Psi_o^{(\text{in},j_a)} \Phi_i^{(\text{in},j_b)} \Psi_i^{(\text{in},j_c)} \right) \quad (12)$$

The final permutation invariant update to the center node information is obtained by concatenating $\{\Psi_{o,k_\alpha}^{(\text{out},j)}\}_{\alpha=1,\dots,l}$, followed by a linear mixing layer along the new concatenated axis. Note that in the sparse implementation, the feature dimension for each incoming activation never gets updated. Each time the internal spin configuration is only specified by a single internal spin label k_α , thus sparsifying the three-body information flow. For each internal spin value k , the 3-body interaction fuses into a single SO(3)-equivariant tensor (this fusion corresponds to the Fig.1 in the SM), while the final messaging passing aggregates neighboring edges and nodes information to the center node. In this implementation based on MACE architectures, we found our fusion block would only marginally increase the number of trainable parameters given the same channel width.

Fusion Block can also be initialized with significantly more trainable parameters than the original Mace does, which we denote as the *dense* implementation. The key difference to the sparse implementation is the inclusion of multi-partite internal spins to create a nested FusionBlock module. More specifically, given a l -sequence of internal spins $\{k_\alpha\}_\alpha$, we can choose a tuple $(k_{\alpha_1}, k_{\alpha_2})$, a triple $(k_{\alpha_1}, k_{\alpha_2}, k_{\alpha_3})$, and beyond instead of specifying a single choice of the internal spin in the sparse implementation. Hence, a total of $l!$ selections can be made resulting in a significant boost to the model size and number of trainable parameters. In our explicit implementation, we feed all choices of internal spins at once, resulting in a typical 10X boost of the trainable parameter size. As a result, we do not need to additionally pass a linear layer to reshape the channel width. The dense model could often outperform or be on par with the sparse implementation only with half the channel width. As an overall observation, the choices of internal spins are vital to our numerical performance. In our practice, the internal spins are chosen to range from $j = 0, 1, 2$, and sometimes with both parties.

Results

We describe three well-rounded benchmarks to test CoFD and MoFD, including QM-9³⁶ molecular property prediction, MD-17⁶ small molecular dynamics, non-adiabatic molecular dynamics of stilbene. Our results are summarized in Figure 2, Table 1 and Table 2.

QM-9 Molecular properties and MD-17 molecular dynamics datasets

We first implement the fusion diagram on Cormorant architecture³⁵. The standard QM-9 benchmark dataset³⁶ is used to test the performance of the CoFD model to predict molecular quantum properties of roughly 130,000 molecules in equilibrium, which contains multiple tasks of scalar value regression including atomization enthalpy, free energy, etc. In contrast, the MD-17 dataset⁶ involves learning the ground-state PES and its gradient, for eight small organic molecules at room temperature from reference DFT calculations.

We compare the CoFD model and the original Cormorant model. The fusion diagram reduces the number of parameters in our networks, ensuring that we are not simply improving performance by adding additional parameters: for MD17, the networks with fusion diagrams have 135393 parameters compared to 154241 in the original Cormorant³⁵, and our QM9 neural network has 121872 parameters compared to 299808 in the original³⁵. We report that the total time of training QM9 (resp. MD17) use 20 (resp. 12) hours with 256 Epoches, each with a mini-batch size of 64. Hence each epoch costs 281 (resp. 169) seconds. Code for our modified network can be found at https://github.com/ehthiede/diagram_corm. To be noted, the fusion block used in the CoFD to predict QM-9 and MD-17 is a sparse implementation. We did not use the dense implementation in predicting the QM-9 and MD-17 properties due to the large computational expense. However, it would be an interesting future direction to reduce the recourse overhead in the dense implementation, which would enable more subsequent experiments.

Stilbene Non-adiabatic molecular dynamics

Non-adiabatic MD (NAMD)⁴⁸ is a powerful approach for predicting photo-induced chemical processes, including photocatalytic reactivity⁴⁹, photo-induced DNA damage⁵⁰, and the performance of sun-screening products⁵¹. Unlike ground-state dynamics, NAMD involves evaluating multiple PESs and their gradients simultaneously. However, studying excited-state dynamics requires higher accuracy electronic structure methods than DFT⁵², resulting in significantly higher computational costs. Thus, there is motivation to test our model's ability to study multiple PESs that are not generated by DFT.

In this study, we explore the photo-induced cis-trans isomerization process of stilbene, a phenomenon first reported by Syage⁵⁴. Our approach utilizes the Complete Active Space Self-Consistent Field (CASSCF) theory⁵⁵, specifically targeting the conjugated π orbital localized on the carbon-carbon double bond and its anti-bonding counterpart. This selection forms our active space, characterized as two electrons in two orbitals (2e,2o), and all calculations are conducted using the 6-31G* basis set. To accurately capture the quantum effects inherent in photoisomerization, we adopt a quantum-classical approximation through trajectory surface hopping (TSH), as implemented in the SHARC package⁵⁶. This method integrates both quantum and classical dynamics, crucial for studying processes like isomerization. Wigner sampling⁴⁵ is employed to generate a variety of initial configurations, initiating the molecular trajectories under study.

Table 1. Mean absolute error of various prediction targets on QM-9 (left) and conformational energies (in units of kcal/mol) on MD-17 (right), for both the original Cormorant architecture and our modified version that incorporates a fusion block. It should be noted that the CoFD models have significantly fewer parameters than the original Cormorant. We report the mean and standard deviation from multiple runs. In comparison, the model with lower predictive error has been bolded.

	Cormorant	CoFD		Cormorant	CoFD
α (bohr ³)	0.085 (0.001)	0.088 (0.003)			
$\Delta\epsilon$ (eV)	0.061 (0.005)	0.062 (0.001)			
ϵ_{HOMO} (eV)	0.034 (0.002)	0.0391 (0.0008)			
ϵ_{LUMO} (eV)	0.038 (0.008)	0.0347 (0.0006)			
μ (D)	0.038 (0.009)	0.035 (0.001)			
C_v (cal/mol K)	0.026 (0.000)	0.0272 (0.0002)			
G (eV)	0.020 (0.000)	0.0135 (0.0002)			
H (eV)	0.021 (0.001)	0.0132 (0.0004)			
R^2 (bohr ²)	0.961 (0.019)	0.50 (0.02)			
U (eV)	0.021 (0.000)	0.0130 (0.0004)			
U_0 (eV)	0.022 (0.003)	0.0133 (0.0003)			
ZPVE (meV)	2.027 (0.042)	1.43 (0.04)			

Model	Feature Dimension	Num. of Param.	Train Size	Ground State (Energy, Forces)	First Excited State (Energy, Forces)	Second Excited State (Energy, Forces)
MACE	64	330320	285	(19.15 , 0.70)	(9.88 , 1.25)	(21.80 , 1.03)
MoFD-sparse	16	66784	285	(19.74, 0.62)	(13.43, 1.12)	(23.42, 1.07)
MACE	128	979088	950	(26.44 , 1.30)	(29.07 , 3.56)	(48.77 , 3.05)
MoFD-sparse	32	141168	950	(28.84, 1.40)	(36.69, 3.55)	(55.08, 3.11)
MoFD-dense	16	690976	950	(27.59, 1.14)	(32.64, 3.31)	(54.76, 2.65)

Table 2. Comparative analysis of MACE and MoFD models in dense and sparse implementations, evaluated on single and multiple independent non-adiabatic trajectories of cis-stilbene. The table presents the feature dimension, number of parameters (Num. of Param.), training set size (Train Size), and results for the ground state, first excited state, and second excited state. Results include energy values in milli-Hartree (mHartree) and forces in milli-Hartree per Angstrom (mHartree/A). The bold figures represent the best performance in each category.

A stringent criterion is applied to ensure the quality of the data: only trajectories maintaining total energy conservation within 0.2 eV were considered valid and included in the dataset. This threshold ensures the physical relevance of the trajectories by excluding those that do not adhere to energy conservation principles. The resultant dataset, therefore, comprises multiple molecular trajectories of stilbene, predominantly initiated in an excited state. These trajectories provide a comprehensive view of the isomerization process, offering valuable insights into the dynamics of this photochemical reaction. Detailed computational specifications and a more thorough introduction to the methods employed are available in the SM.

The widely-adopted MD17 dataset⁵⁷ comprises adiabatic dynamic trajectories using the PBE functional, though the spin polarization, basis set, and computational grid information are absent from the literature, near equilibrium, where molecular movements are trivial. As a result, MD17 is heavily biased towards sampling the reactant region of the PES without considering the driven non-equilibrium forces.⁵⁸ However, a meaningful chemical reaction typically involves three parts on the PES: reactant, product, and transition state. It is important to note that the accuracy of common density functionals is usually a few kcal/mol when compared to higher levels of theory. For example, the PBE functional used in the MD17 dataset has an average error of more than 9 kcal/mol (roughly 0.4 eV) when predicting reaction barriers⁵⁹. In contrast, the trajectories we sampled visited the reactant, product, and transition state regions of multiple PESs, as illustrated in Figure 2.

To compare the performance of MACE and MoFD with sparse implementation, we selected one reactive trajectory and employed the MACE model with a feature channel dimension of 64 and high-order equivariant features with $l = 0, 1, 2, 3$. For MoFD, we maintain the same feature angular momentum and set the feature channel dimension to 16, resulting in a model with only 66,784 parameters, an order of magnitude smaller than that of MACE. Given the increased difficulty in predicting atomic forces, we adjust the training loss on energy and forces with a ratio of 1:1000, as recommended in previous literature^{11,12}. As the loss is disproportionately weighted towards the force, we concentrate on the force regression performance. The models are trained on 285 samples and tested on a separate hold-out test set of 428 samples. The models were trained in a state-specific fashion, which means each model regresses single state's PES and forces for comparison purposes. Our findings indicate that MoFD with sparse implementation has a decent performance in force prediction for the first two states, while MACE fits better when predicting the energy.

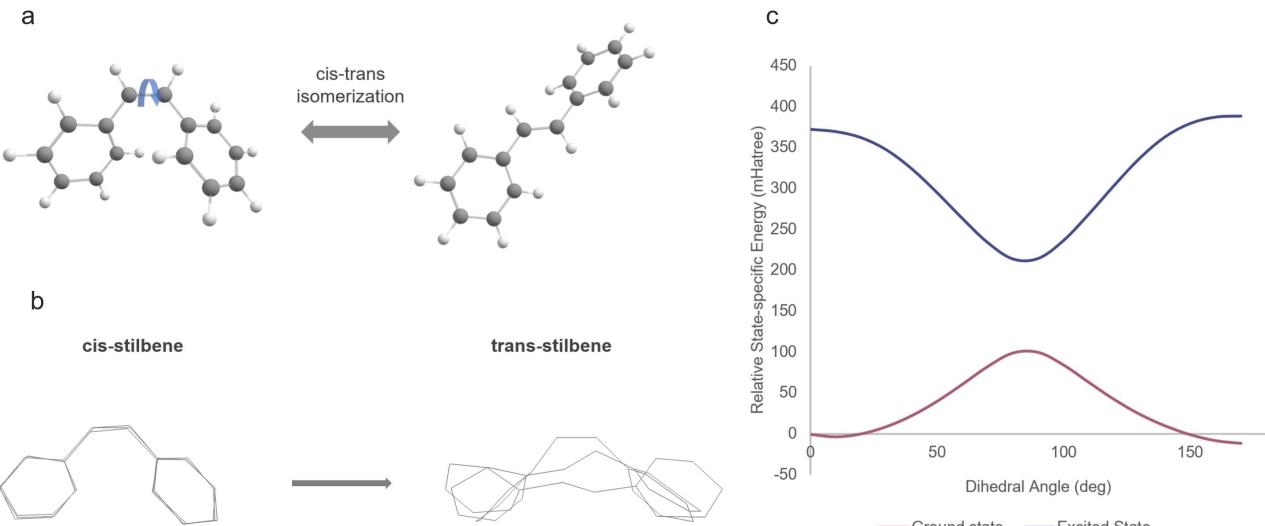


Figure 2. (a) Illustration of photo-induced cis-trans isomerization of stilbene (b) Initial and end configurations of three representative trajectories, which are Wigner-sampled.⁵³ (c) the one-dimensional cut of stilbene ground/ excited-state PESs by rotating the carbon-carbon bond as illustrated in (a), which illustrates the energetic diagram of stilbene isomerization process.

We further assess the generalization ability of our models across different trajectories by incorporating two additional independent trajectories into the dataset, resulting in a total of 950 training samples and 1,395 hold-out testing samples. We increase the complexity of MACE by expanding its feature channel width to 128, leading to a total of 979,088 parameters. Concurrently, we double the feature dimension of MoFD to 32, making it only as large as MACE. Additionally, we implement MoFD with a dense feature dimension of 16, with equivariant features $l = 0, 1, 2$, resulting in a total of 690,976 parameters (29.4% fewer than the original MACE). In terms of runtime, each epoch requires 52 seconds in the MACE model compared to 31 seconds in the MoFD model, attributed to the utilization of lower-dimensional angular momentum features as inputs. The MoFD model with the dense implementation surpass MACE in the force prediction tasks, while the MoFD model with the sparse implementation remains comparable to MACE's accuracy as indicated in Table 2. Nonetheless, it is crucial to note that the performance of all models decrease when learning excited states due to the less well-defined topologies of excited-state PESs⁵².

Discussion

In this work, we have introduced a new method for constructing equivariant blocks for rotation-equivariant layers based on fusion diagrams. Previous work has shown that tensor products can be used to construct neurons for rotation-equivariant neural networks. Moreover, prior research has observed that neural network ansatzes for the quantum system can be unified with spin network ansatzes. Our work is the first to employ these connections in the opposite direction: by employing diagrammatic methods used in physics, we construct new components that can be incorporated into equivariant neural networks.

Using classic results from invariant theory, we show that neural networks built from using fusion blocks are capable of approximating any continuous SU(2)-equivariant functions. To demonstrate the practical utility of fusion blocks, we perturb existing SO(3) equivariant neural network architectures, such as Cormorant³⁵ and MACE¹², by incorporating a fusion block in each layer. The modified architectures generally achieves better performance for a smaller number of parameters. Indeed, the idea of using equivariance and symmetry to prune neural networks has been applied⁶⁰ in the quantum setting. We believe this indicates that fusion blocks can be a useful addition to group-equivariant neural networks.

To test the performance of the fusion block approach, we apply the revised CoFD and MoFD models not only to the standard benchmark datasets QM-9³⁶ and MD-17⁶, but also novel applications such as non-adiabatic molecular dynamics. We find that the addition of the fusion blocks improved the performance of the models.

In future work, we hope to use fusion blocks to improve the interpretability of equivariant neural networks. In theoretical physics, fusion diagrams represent physical processes that correspond to many-body interactions. Furthermore, physicists often manipulate fusion diagrams through internal permutations through a process known as *recoupling*. Recouplings relate

to the physical properties of different fusion diagrams and can show symmetries present in the products that may not be immediately apparent by inspection. Employing the formalism of recoupling may highlight hidden symmetries in the network architecture, indicating new ways to save computational effort. Employing the language of fusion diagrams in these settings could help unify our physical picture of fusion diagrams with computational realities. Finally, fusion diagrams are graphical representations of ways in which local atoms are being fused. It is of interest to consider the effect of the local subgraph topology on the corresponding fusion blocks; in particular, whether fusion diagrams serve as a general principle towards building more expressive graph neural nets with 3D equivariance specific to chemical applications. We leave addressing these questions as future research opportunities.

References

1. Wang, H., Zhang, L., Han, J. & Weinan, E. Deepmd-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Comput. Phys. Commun.* **228**, 178–184 (2018).
2. Schoenholz, S. & Cubuk, E. D. Jax md: a framework for differentiable physics. *Adv. Neural Inf. Process. Syst.* **33**, 11428–11441 (2020).
3. Atz, K., Grisoni, F. & Schneider, G. Geometric deep learning on molecular representations. *Nat. Mach. Intell.* **3**, 1023–1032 (2021).
4. Walters, W. P. & Barzilay, R. Applications of deep learning in molecule generation and molecular property prediction. *Accounts chemical research* **54**, 263–270 (2020).
5. Shuaibi, M. *et al.* Rotation invariant graph neural networks using spin convolutions. *arXiv preprint arXiv:2106.09575* (2021).
6. Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, DOI: [10.1126/sciadv.1603015](https://doi.org/10.1126/sciadv.1603015) (2017).
7. Esteves, C. Theoretical aspects of group equivariant neural networks. *arXiv preprint arXiv:2004.05154* (2020).
8. Cohen, T. S. *et al.* Equivariant convolutional networks (2021).
9. Garcia Satorras, V., Hoogeboom, E. & Welling, M. E(n) Equivariant Graph Neural Networks. *arXiv e-prints arXiv:2102.09844*, DOI: [10.48550/arXiv.2102.09844](https://doi.org/10.48550/arXiv.2102.09844) (2021). [2102.09844](https://doi.org/10.48550/arXiv.2102.09844).
10. Du, W. *et al.* SE(3) Equivariant Graph Neural Networks with Complete Local Frames. *arXiv e-prints arXiv:2110.14811*, DOI: [10.48550/arXiv.2110.14811](https://doi.org/10.48550/arXiv.2110.14811) (2021). [2110.14811](https://doi.org/10.48550/arXiv.2110.14811).
11. Batzner, S. *et al.* E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials (2021). [2101.03164](https://doi.org/10.48550/arXiv.2101.03164).
12. Batatia, I., Kovács, D. P., Simm, G. N., Ortner, C. & Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *arXiv preprint arXiv:2206.07697* (2022).
13. Kondor, R., Lin, Z. & Trivedi, S. Clebsch-Gordan Nets: a fully Fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems (NeurIPS)* (2018).
14. Kondor, R. & Trivedi, S. On the generalization of equivariance and convolution in neural networks to the action of compact groups (2018). [1802.03690](https://doi.org/10.48550/arXiv.1802.03690).
15. Olver, P. J. *Classical Invariant Theory*. London Mathematical Society Student Texts (Cambridge University Press, 1999).
16. Procesi, C. *Lie groups: an approach through invariants and representations*. Universitext (Springer, New York, 2007). OCLC: ocm61309419.
17. Goodman, R. & Wallach, N. R. *Symmetry, Representations, and Invariants* (Springer New York, 2009).
18. White, S. R. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **69**, 2863 (1992).
19. Rommer, S. & Östlund, S. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B* **55**, 2164 (1997).
20. Orús, R. Tensor networks for complex quantum systems. *Nat. Rev. Phys.* **1**, 538–550, DOI: [10.1038/s42254-019-0086-7](https://doi.org/10.1038/s42254-019-0086-7) (2019).
21. Fannes, M., Nachtergael, B. & Werner, R. F. Finitely correlated states on quantum spin chains. *Commun. Math. Phys.* **144**, 443–490 (1992).

22. Verstraete, F. & Cirac, J. I. Renormalization algorithms for quantum-many body systems in two and higher dimensions (2004).
23. Vidal, G. Entanglement renormalization. *Phys. Rev. Lett.* **99**, 220405 (2007).
24. Huang, C. *et al.* Efficient parallelization of tensor network contraction for simulating quantum computation. *Nat. Comput. Sci.* **1**, 578–587 (2021).
25. Pan, F. & Zhang, P. Simulating the sycamore quantum supremacy circuits (2021).
26. Stoudenmire, E. & Schwab, D. J. Supervised learning with tensor networks. *Adv. Neural Inf. Process. Syst.* **29** (2016).
27. Efthymiou, S., Hidary, J. & Leichenauer, S. Tensor network for machine learning (2019).
28. Roberts, C. *et al.* Tensornetwork: A library for physics and machine learning. *arXiv preprint arXiv:1905.01330* (2019).
29. Swingle, B. Entanglement renormalization and holography. *Phys. Rev. D* **86**, 065007 (2012).
30. Pastawski, F., Yoshida, B., Harlow, D. & Preskill, J. Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *J. High Energy Phys.* **2015**, 1–55 (2015).
31. Mäkinen, I. Introduction to SU(2) recoupling theory and graphical methods for loop quantum gravity (2019). [1910.06821](#).
32. Singh, S. & Vidal, G. Tensor network states and algorithms in the presence of a global su(2) symmetry. *Phys. Rev. B* **86**, DOI: [10.1103/physrevb.86.195114](#) (2012).
33. Schmoll, P., Singh, S., Rizzi, M. & Orús, R. A programming guide for tensor networks with global su(2) symmetry. *Annals Phys.* **419**, 168232, DOI: [10.1016/j.aop.2020.168232](#) (2020).
34. Schmoll, P. & Orús, R. Benchmarking global $su(2)$ symmetry in two-dimensional tensor network algorithms. *Phys. Rev. B* **102**, 241101, DOI: [10.1103/PhysRevB.102.241101](#) (2020).
35. Anderson, B., Hy, T.-S. & Kondor, R. Cormorant: Covariant molecular neural networks (2019). [1906.04015](#).
36. Ramakrishnan, R., Dral, P. O., Rupp, M. & von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1** (2014).
37. Zaheer, M. *et al.* Deep Sets. *arXiv e-prints* arXiv:1703.06114 (2017). [1703.06114](#).
38. Segol, N. & Lipman, Y. On Universal Equivariant Set Networks (2019). [1910.02421](#).
39. Maron, H., Litany, O., Chechik, G. & Fetaya, E. On learning sets of symmetric elements. In *International Conference on Machine Learning*, 6734–6744 (PMLR, 2020).
40. Cohen, T. S. & Welling, M. Group equivariant convolutional networks (2016). [1602.07576](#).
41. Maron, H., Ben-Hamu, H., Shamir, N. & Lipman, Y. Invariant and equivariant graph networks. In *International Conference on Learning Representations* (2019).
42. Thiede, E. H., Hy, T. S. & Kondor, R. The general theory of permutation equivariant neural networks and higher order graph variational encoders (2020). [2004.03990](#).
43. Varshalovich, D. A., Moskalev, A. N. & Khersonskii, V. K. *Quantum Theory of Angular Momentum* (WORLD SCIENTIFIC, 1988).
44. Lindgren, I. & Morrison, J. *Atomic Many-Body Theory* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1986).
45. Biedenharn, L. C., Giovannini, A. & Louck, J. D. Canonical definition of wigner coefficients in un. *J. Math. Phys.* **8**, 691–700, DOI: [10.1063/1.1705266](#) (1967). <https://doi.org/10.1063/1.1705266>.
46. Gao, M.-J. & Chen, J.-Q. THE CLEBSCH-GORDAN COEFFICIENTS OF PERMUTATION GROUPS S(2) - S(6). *J. Phys. A* **18**, 189–213, DOI: [10.1088/0305-4470/18/2/009](#) (1985).
47. Batatia, I. *et al.* The design space of E(3)-equivariant atom-centered interatomic potentials, DOI: [10.48550/ARXIV.2205.06643](#) (2022).
48. Tully, J. C. Perspective: Nonadiabatic dynamics theory. *The J. chemical physics* **137**, 22A301 (2012).
49. Tong, H. *et al.* Nano-photocatalytic materials: possibilities and challenges. *Adv. materials* **24**, 229–251 (2012).
50. Dougherty, T. J. *et al.* Photodynamic therapy. *JNCI: J. national cancer institute* **90**, 889–905 (1998).
51. Baker, L. A., Marchetti, B., Karsili, T. N., Stavros, V. G. & Ashfold, M. N. Photoprotection: extending lessons learned from studying natural sunscreens to the design of artificial sunscreen constituents. *Chem. Soc. Rev.* **46**, 3770–3791 (2017).

52. Mai, S., Atkins, A. J., Plasser, F. & González, L. The influence of the electronic structure method on intersystem crossing dynamics. the case of thioformaldehyde. *J. Chem. Theory Comput.* **15**, 3470–3480 (2019).
53. Dahl, J. P. & Springborg, M. The morse oscillator in position space, momentum space, and phase space. *The J. chemical physics* **88**, 4535–4547 (1988).
54. Syage, J., Lambert, W. R., Felker, P., Zewail, A. & Hochstrasser, R. Picosecond excitation and trans-cis isomerization of stilbene in a supersonic jet: Dynamics and spectra. *Chem. Phys. Lett.* **88**, 266–270 (1982).
55. Roos, B. O., Taylor, P. R. & Sigbahn, P. E. A complete active space scf method (casscf) using a density matrix formulated super-ci approach. *Chem. Phys.* **48**, 157–173 (1980).
56. Mai, S., Marquetand, P. & González, L. Nonadiabatic dynamics: The sharc approach. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **8**, e1370 (2018).
57. Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Sci. advances* **3**, e1603015 (2017).
58. Pengmei, Z., Liu, J. & Shu, Y. xxmd: Benchmarking neural force fields using extended dynamics beyond equilibrium (2023). [2308.11155](https://arxiv.org/abs/2308.11155).
59. Mardirossian, N. & Head-Gordon, M. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 200 density functionals. *Mol. physics* **115**, 2315–2372 (2017).
60. Wang, X. *et al.* Symmetric pruning in quantum neural networks. *arXiv preprint arXiv:2208.14057* (2022).

Acknowledgements

J.L. is supported in part by International Business Machines (IBM) Quantum through the Chicago Quantum Exchange, and the Pritzker School of Molecular Engineering at the University of Chicago through AFOSR MURI (FA9550-21-1-0209).

Supplementary Material

Zimu Li, Zihan Pengmei, Han Zheng, Erik Thiede, Junyu Liu, Risi Kondor

Contents

1 Overview of Fusion Diagrams	1
2 Proof of the Universality Theorem	3
2.1 Equivariant Polynomials and Clebsch-Gordan Product	4
2.2 Spin Diagram and Spin Recoupling	6
2.3 Equivariant Universality with Fusion Blocks	8
3 Architecture, hyper-parameter, and computational details	11
3.1 Electronic structure method	11
3.2 Dynamics details	12
4 Additional Numerical Results	13

1 Overview of Fusion Diagrams

Clebsch-Gordan products (CG-products) appear naturally in quantum mechanics out of necessity to describe the interaction of multiple particles under global $SU(2)$ symmetry. For even moderately complex quantum mechanical systems, equations keeping track of successive products will become formidably complicated. Consequently, physicists have developed a diagrammatic representation of successive Clebsch-Gordan decompositions. We refer to these diagrams as *fusion diagrams*. The space of fusion diagrams that fuse n incoming $SU(2)$ -contravariant features $\{\psi^{(j_i)}\}_{i=1}^n$ (defined in Background in the main text) into a single outgoing $SU(2)$ -contravariant feature $\Psi^{(J)}$ comprised of successive Clebsch-Gordan products is denoted by $\text{Hom}_{SU(2)}(V_{j_1} \otimes \cdots \otimes V_{j_n}, V_J)$ (Fig.1). By choosing a fusion diagram $Q^{(j_1, \dots, j_n; J)} \in$

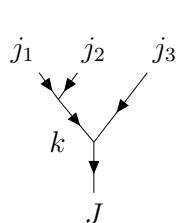


Figure 1: A fusion diagram as a standard basis element in $\text{Hom}_{SU(2)}(V_{j_1} \otimes V_{j_2} \otimes V_{j_3}, V_J)$. Fusion diagrams of this type with all possible values k are also called Wigner 4-jm elements.

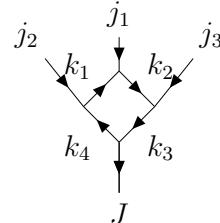


Figure 2: Another fusion diagram as some element in $\text{Hom}_{SU(2)}(V_{j_1} \otimes V_{j_2} \otimes V_{j_3}, V_J)$.

$\text{Hom}_{\text{SU}(2)}(V_{j_1} \otimes \cdots \otimes V_{j_n}, V_J)$ the fusion process can be computed by the following form like inner products:

$$\psi^{(J)} = \langle Q^{(j_1, \dots, j_n; J)}, \psi^{(j_1)} \otimes \cdots \otimes \psi^{(j_n)} \rangle. \quad (1)$$

To be more specific, in a fusion diagram, e.g., Fig.1 & 2, each edge is associated with a spin label, which is a non-negative integer or half-integer standing for an SU(2) irrep. While only integers appears in the case of SO(3) irreps. Each vertex represents a common CG-product $C^{(j_a, j_b, j_c)}$ and is therefore connected to exactly three edges¹. We adopt the following convention in labeling fusion diagrams, which have n external incoming edges and one external outgoing edge:

1. Each fusion diagram has a list of external spin labels $\{j_1, j_2, \dots, j_n\} \equiv \mathbf{j}$, and a list of internal spin labels $\{k_1, \dots, k_m\} \equiv \mathbf{k}$. These lists are referred to as external and internal spin configurations respectively.
2. For each vertex, input vectors are always assumed to be transformed contravariantly, i.e., with upper entry indices m so that they are fused together via Eq.(1) and yield a contravariant output which would be used for subsequent updates of CG-products like in Figure 1.
3. The spins associated with each vertex must obey the Clebsch-Gordan rule: denoting the two incoming edges j_a and j_b and the outgoing edge j_c , we require that $|j_a - j_b| \leq j_c \leq j_a + j_b$ [3, 25].
4. We label the final outgoing edge with J so that incoming n SU(2)-contravariant features are fused into one SU(2)-contravariant feature to be processed in the next layer.

For instance, the fusion diagram in Figure 1 corresponds to

$$Q^{(\mathbf{j}, \mathbf{k}; J)}_{m_{j_1} m_{j_2} m_{j_3}}{}^{m_J} \equiv M C^{(j_1 j_2 k)}_{m_{k_1} m_{j_2}}{}^{m_k} C^{(k j_3 J)}_{m_k m_{j_3}}{}^{m_J}, \quad (2)$$

where M is a normalization coefficient equal to $\sqrt{(2k_1 + 1)(2k_2 + 1)}$.

In the following discussion, we will refer to any fusion diagrams that form a basis in the invariant Hom set as *Wigner n-jm symbol* where n counts the total outgoing and incoming edges. Wigner n -jm elements such as Fig.1 are prototypes for fusion blocks and essential in proving our theoretical results later. For a fixed set of external spin labels \mathbf{j} , varying all possible assignments of \mathbf{k} of (2) exhausts all possible orthogonal basis elements in $\text{Hom}_{\text{SU}(2)}(V_1 \otimes V_2 \otimes \cdots \otimes V_n, U)$ [25, 16, 4].

There are also more general spin diagrams that we can draw. For instance, with three incoming external edges and one outgoing edge (Figure 2)

$$Q^{(\mathbf{j}, \mathbf{k}; J)}_{m_{j_1} m_{j_2} m_{j_3}}{}^{m_J} = C^{(j_1, k_2, k_1)}_{m_{j_1} m_{k_2} m_{k_1}} C^{(j_2 k_1 k_4)}_{m_{j_2} m_{k_1} m_{k_4}} C^{(j_3, k_3, k_2)}_{m_{j_3} m_{k_3} m_{k_2}} \\ C^{(J, k_4, k_3)}_{m_J m_{k_4} m_{k_3}}, \quad (3)$$

can be expressed as a linear combination of Wigner-4jm tensors in (2) with different internal spin configuration \mathbf{k} because they form a basis for all such high-order CG-products.² Note that the dimension of the invariant Hom set grows exponentially with the number of outgoing and incoming edges so it remains an intriguing question to determine which fusion diagrams could span larger subspaces that help build practically useful universal neural network architectures. One motivation for using more general fusion diagrams such as Fig.2 is that not only does it spans multiple Wigner symbols at once, it can encode non-trivial local graphical invariance

¹One can also assume that vertices connected with more than three edges are sup-fusion diagrams with vertices having only three edges. We refer the interested audience to [23, 22, 21] for more details. Note for general tensor networks without prescribed symmetry, there is no such restriction.

²For readers familiar with matrix product states, we observe that (3) is a matrix product state with boundary conditions used in tensor networks [23, 22, 21].

such as the C_4 group invariance when choosing appropriate \mathbf{j} and \mathbf{k} .

2 Proof of the Universality Theorem

Equivariant Universal Approximation

Rotationally equivariant neural architectures affording universal approximation on point clouds are analyzed in [15]. Here we show that the fusion block activation is an universal approximator subject to $SU(2)$ or $SO(3)$ equivariance but using different methods, borrowing techniques from classical invariant theory, which studies invariant and equivariant polynomials of classical groups [18, 20, 8]. Similar universality statements to the ones we present have also been given in [4]. We pursue a different proof strategy, using iterated Clebsch-Gordan products instead of constructing all basis elements for the infinite-dimensional space of equivariant functions. To be specific, it was proved by Gordan, Hilbert, and others in Classical Invariant Theory that Clebsch-Gordan products are *the only* essential ingredients to construct any $SU(2)$ -equivariant function [10, 18, 20]. A formal statement of this significant result is provided in Theorem 2.2 later while we present the following adapted form for our case.

Theorem 2.1. *Let $V_j, W_{j'}$ be two irreps of $SU(2)$ and let $\Psi_1, \dots, \Psi_n \in V_j$ be spin- j vectors as inputs. For simplify, we temporarily denote by (Ψ_r, \dots, Ψ_s) an iterated CG-product. Given a multi-degree $\mathbf{d} = (d_1, \dots, d_n)$, a CG-product for multi-homogeneous monomial of degree \mathbf{d} is written as*

$$\left(\underbrace{(\Psi_1, \dots, \Psi_1)}_{d_1 \text{ copies}}, \dots, \underbrace{(\Psi_n, \dots, \Psi_n)}_{d_n \text{ copies}} \right) \quad (4)$$

Any $SU(2)$ -equivariant polynomial $f : V_j^{\oplus n} \rightarrow W_{j'}$ can be written as a linear combination of these multi-homogeneous monomials.

The above theorem says that any $SU(2)$ -equivariant polynomial can be constructed from CG-products. Using an equivariant version of the Stone-Weierstrass theorem [5], that any continuous equivariant function can be approximated by equivariant polynomials. As a result, they can be approximated by iterated CG-products which are graphically represented as fusion diagrams in our setting.

We now explore both rotation and permutation equivariant universality of neural networks using fusion blocks. Graph neural networks (GNNs) in practice perform local aggregation on the graph; in our case, this corresponds to the case where the fusion blocks may be applied only to particles in a local atomic neighborhood. For this reason, we prove universality for functions computed in the vicinity of an atom in the points cloud. This is a standard and reasonable assumption since the the graph structure of points clouds are defined based on interactions of atoms within a radial distance for which the induced local subgraph is fully connected. If receptive field is set to be global, we obtain the universality when taking interactions of any pair of atoms from the points cloud into account. Let us assume that the learning task is defined on $V_j^{\oplus n}$ with n representing the number of atoms in a local neighborhood or simply the entire system. Intuitively, as fusion blocks fuse inputs on each atom with the center and then aggregate (see Fig.1 in the main text), iterated CG-products will be built with permutation equivariance after updating several layers.

The universality theorem is proved by results from Classical Invariant Theory [9, 18, 20] and Quantum Angular Momentum Recoupling Theory [14, 25, 16]. Classical Invariant Theory studies polynomial maps from V to W equivariant under classical group actions, like these of $SO(n)$, $SU(n)$, $SL(n, \mathbb{C})$. For short, they are called *G-equivariant polynomials* or *covariants*.

For instance, linear equivariant maps commonly used in designing equivariant neural networks are simply equivariant polynomials of degree one. Clebsch-Gordan products like $C^{j_1 j_2 k} \psi^{j_1} \psi^{j_2}$ are equivariant polynomials of degree two. Invariant polynomials are simply special cases of covariants when the target space W admits the trivial G -representation. Quantum Angular Momentum Recoupling Theory specializes the case for $SU(2)$ tensor product representations. Different ways of taking tensor products are referred as different *recoupling schemes* and produces different results in computing Clebsch-Gordan coefficients (CGC) which has real significance in studying quantum system like the $jj - LS$ recoupling for electron configuration [14]. Spin diagrams used in this paper are graphical interpretation of various recoupling schemes. They are further generalized in Topological Quantum Computation [11] and Quantum Gravity [19]. Classical Invariant Theory and Angular Momentum Recoupling Theory provides different insights on Clebsch-Gordan products. We will introduce gradually how we benefit from these insights and obtain the universality theorem in the following.

2.1 Equivariant Polynomials and Clebsch-Gordan Product

Equivariant layers are central part in equivariant neural networks. Formally, they are functions L that commutes with the group action

$$T'_g \circ L = L \circ T_g, \forall g \in G.$$

Most previous researches only focus on *linear equivariant layer* which is a linear map commuting with the G -action (e.g., a convolution). However, one can in principle design *nonlinear equivariant layer* from the very beginning by equivariant polynomials. As a standard notation from Classical Invariant Theory, we denote by $P(V, W)^G$ the collection of all equivariant polynomials from V to W and $P(V)^G$ the collection of all invariant polynomials on V .

To determine $SU(2)$ -equivariant functions from V and W , we decompose V, W into irreps and consider the problem within each irrep. As a well-known result from representation theory, *Schur's lemma* states that linear equivariant maps between isomorphic irreps are simply scalars. For non-isomorphic irreps, the only possible linear equivariant map is the trivial zero map [8]. Therefore, in our setting with $V = V_j^{\oplus n}$, linear equivariant layers can only be defined as learnable weights combining n inputs (ψ_1, \dots, ψ_n) linearly [12]. To update inputs in a nonlinear manner, we can employ the *Clebsch-Gordan decomposition* or *Clebsch-Gordan product* which is formally defined from the tensor product $V_{j_1} \otimes V_{j_2}$ of two irreps to a third irrep V_J which can be decomposed from the tensor product:

$$C^J_{j_1 j_2} : V_{j_1} \otimes V_{j_2} \rightarrow V_J.$$

The Clebsch-Gordan non-linearity was first used in [13, 1]. Explicitly, for any $(\psi_1, \psi_2) \in V_{j_1} \oplus V_{j_2}$ expanded under basis (spherical harmonics basis/quantum angular momentum basis),

$$(C^J_{j_1 j_2}(\psi_1, \psi_2))^M = (C^J_{j_1 j_2})^M_{m_1 m_2} \psi_1^{m_1} \psi_2^{m_2},$$

where m_i denotes the magnetic number, is a *homogeneous polynomial of degree two* because we have the quadratic term $\psi_1^{m_1} \psi_2^{m_2}$ of inputs.

On the other hand, one may argue that the CG-product is linear on each input. That is correct and a CG-product should be formally called a *bilinear function*. We denote the collection of $SU(2)$ bilinear functions by $\text{Hom}_{SU(2)}(V_{j_1}, V_{j_2}; W)$. Another equivalent notation $\text{Hom}_{SU(2)}(V_{j_1} \otimes V_{j_2}; W)$ and its generalization $\text{Hom}_{SU(2)}(V_{j_1} \otimes \dots \otimes V_{j_n}; W)$ for multilinear functions are used more often in the main article. Mathematically, bilinear map on the direct sum $V_{j_1} \oplus V_{j_2}$ corresponds uniquely to a linear map on the tensor product $V_{j_1} \otimes V_{j_2}$ and hence we

use the above notation. This is also the reason that even originally defined on tensor products, we are free to use CG-products on direct sums without any ambiguity. As a reminder, assume $j_1 = j_2 = J$, the following map

$$V_j \mapsto V_j \oplus V_j \rightarrow V_J; \quad \psi \mapsto (\psi, \psi) \mapsto C_{jj}^J(\psi, \psi).$$

is non-linear and equivariant from V_j to V_J . It circumvents the restriction of Schur lemma on linear equivariant maps.

To discuss more nontrivial examples, we now present two ways to generalize CG-products to high order as the central notions used for the proof:

1. *Iterated CG-series* is a canonical high order CG-product by contracting common CG-products sequentially, e.g.,

$$\begin{aligned} C_{kj_3}^J C_{j_1 j_2}^k : V_{j_1} \oplus V_{j_2} \oplus V_{j_3} &\rightarrow V_k \oplus V_{j_3} \rightarrow V_J; \\ (\psi_1, \psi_2, \psi_3) &\mapsto (C_{j_1 j_2}^k(\psi_1, \psi_2), \psi_3) \mapsto C_{kj_3}^J(C_{j_1 j_2}^k(\psi_1, \psi_2), \psi_3) \end{aligned}$$

An n th iterated CG-product is an equivariant polynomial of degree $n + 1$. It is also called a *multi-linear function* with n inputs living in $\text{Hom}_{\text{SL}(2, \mathbb{C})}(V_{j_1} \otimes \cdots \otimes V_{j_n}; W)$.

2. Given any iterated CG-product Q , suppose (j_1, \dots, j_k) are all distinct spin labels of Q but each of which appears d_i times (like C_{jj}^J from above). Inputting the same vector for all repeated spin labels defines a *multi-homogeneous polynomial* of degree $d = (d_1, \dots, d_k)$. For instance

$$C_{j_1 k}^J(C_{j_2 j_1}^k(\psi_1, \psi_2), \psi_1)$$

is a multi-homogeneous polynomial of degree $(2, 1)$. One variable polynomial can be decomposed into homogeneous polynomials, as a generalization, multi-homogeneous polynomials are building blocks for multivariate polynomials [20]. At a result, one cannot always expect a polynomial to be linear on each of its variable.

The above examples shows some hints that CG-products can be assembled together to construct any complicated $\text{SU}(2)$ -equivariant polynomial. This is true due to Gordan Theorem [9, 18, 17]:

Theorem 2.2 (Gordan Theorem). *Given a complex $SL(2, \mathbb{C})$ representation $V = \bigoplus V_j^{\oplus m_j}$ (m_j denotes possible multiplicity of V_j), any equivariant polynomial f on V can be written as a linear combination of iterated CG-products $((\dots((\psi_1, \psi_2)_{k_1}, \psi_3)_{k_2}, \dots, \psi_n)_{k_{n-1}})$. Note that each input may occur more than once in the iteration.*

We use the simplified notation for Iterated CG-products in the above theorem. That is the most standard of notation used in classical invariant theory where CG-product is called *transvectant*. Transvectants are expressed by polynomials and differential operators by Gordan, Hilbert and others [10, 18], while the matrix/tensor notation of CG-products was adapted by later researchers like Wigner and Racah to study quantum physics and representation theory [14, 25]. We follow the tensor notation from physics and representation theory in this paper.

Gordan theorem was originally formulated to find all *generators* of $\text{SL}(2, \mathbb{C})$ -equivariant polynomials, which can be further restricted to $\text{SU}(2)$ -equivariant polynomials by definition. It was then proved by Hilbert in his celebrated Basis Theorem that one just needs finitely many generators to generate the infinite dimensional collection of equivariant polynomials (see [18, 8] for more details). Even though it is extremely difficult to write down these generators explicitly. For small spin labels ($j \sim 10$), a few explicit generators of $\text{SL}(2, \mathbb{C})$ irreps are summarized in [18, 17]. The so-called First Fundamental Theorem says that generators of

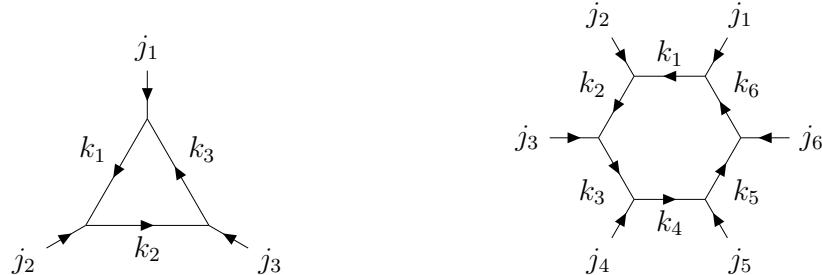
fundamental representation of $GL(n)$, $SO(n)$ and $Sp(n)$ are, loosely speaking, inner products [8] (inner products are fundamental cases of CG-products). For this reason, we do not bother ourselves to compute generators but simply use Gordan theorem which is enough to claim that CG-products are *the only* essential ingredient to construct any $SU(2)$ -equivariant polynomial.

2.2 Spin Diagram and Spin Recoupling

Except the above two algebraic expressions of CG-products: tensor notation and transvectant, we present more details about *spin diagrams/fusion diagrams* as the main topic of this paper. In the context of physics, iterated CG-products are special cases of tensor networks with $SU(2)$ -symmetry [23, 22, 21]. While general tensor networks are obtained by contracting arbitrary tensors which may not be $SU(2)$ -equivariant³. Graphically, we draw



as a common CG-product (LHS) and a third order iterated CG-products (RHS) with k being an internal spin. Outputs of fusion diagrams are generally spin- J vectors. To construct *invariant polynomials* with scalar output however, one can set $J = 0$ or consider spin diagrams like



To build even complicated fusion diagrams, we just need to keep in mind that each vertex in the diagram is connected with *three and only three legs* which represents a simple CG-product (see [23, 22, 21] for more examples). Based on the graphical expression, Gordan theorem 2.2 says that no matter how complicated a fusion diagram would be, it can be written as a linear combination of fusion diagrams of iterated CG-products.

As we mentioned above, multi-homogeneous polynomials are building blocks for multivariate polynomials. Because of repeated inputs, a general multi-homogeneous polynomial of degree $d = (d_1, \dots, d_k)$ has $\frac{d!}{d_1! \cdots d_k!}$ possible different forms even written as a iterated CG-product like

$$C_{j_2k}^J(C_{j_1j_1}^k(\psi_1, \psi_1), \psi_2), \quad C_{j_1k}^J(C_{j_2j_1}^k(\psi_2, \psi_1), \psi_1), \quad C_{j_1k}^J(C_{j_1j_2}^k(\psi_1, \psi_2), \psi_1).$$

The notion of *spin recoupling* accompanied spin diagrams from angular momentum recoupling theory can help to resolve this problem: like an extension of Gordan theorem, it says that for instance, the second and third iterated CG-products from above can be written as a linear combination of the first one with internal spins varying. This result is desirable because it facilitates the proof of universality theorem later. To be precise, recoupling can be summarized

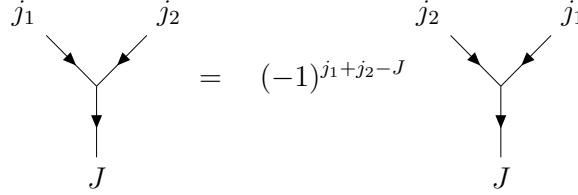
³It is more common to use the word "symmetry" to indicate that a physics system is symmetric under certain transformation while the word "equivariance" is more formal in group theory

as two symmetric properties when computing Clebsch-Gordan coefficients [14, 25, 11]:

1. The *R-move*:

$$(C_{j_1 j_2}^J)^M{}_{m_1 m_2} = (-1)^{j_1 + j_2 - J} (C_{j_2 j_1}^J)^M{}_{m_2 m_1}$$

which says if we exchange the coupling order from $V_{j_1} \otimes V_{j_2}$ to $V_{j_2} \otimes V_{j_1}$, Clebsch-Gordan coefficients will differ by a phase factor $(-1)^{j_1 + j_2 - J}$. It is symmetric in some cases (like inner products), but alternating in other cases. Graphically, we draw



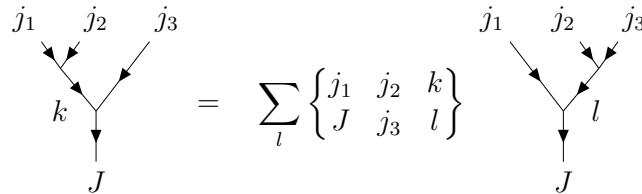
2. The so-called *F-move* accounts for higher order cases: consider the following third order iterated CG-products:

$$\begin{aligned} (\psi_1, \psi_2, \psi_3) &\mapsto (C_{j_1 j_2}^k(\psi_1, \psi_2), \psi_3) \mapsto C_{kj_3}^J(C_{j_1 j_2}^k(\psi_1, \psi_2), \psi_3), \\ (\psi_1, \psi_2, \psi_3) &\mapsto (\psi_1, C_{j_2 j_3}^l(\psi_2, \psi_3)) \mapsto C_{j_1 l}^J(\psi_1, C_{j_2 j_3}^l(\psi_2, \psi_3)), \end{aligned}$$

where the first operator couples j_1, j_2 firstly and then takes j_3 while the second couples j_2, j_3 firstly and then takes j_1 . These two coupling schemes can be transformed by *Wigner 6j-symbols*:

$$(C_{j_1 l}^J C_{j_2 j_3}^l)_{m_1 m_2 m_3 M} = \sum_k (2k+1)(-1)^{j_2 + j_3 + k + l} \left\{ \begin{matrix} j_1 & j_2 & k \\ J & j_3 & l \end{matrix} \right\} (C_{kj_3}^J C_{j_1 j_2}^k)_{m_1 m_2 m_3 M}.$$

It can be thought as a generalized associative rule when coupling more than two spins. Graphically, we draw

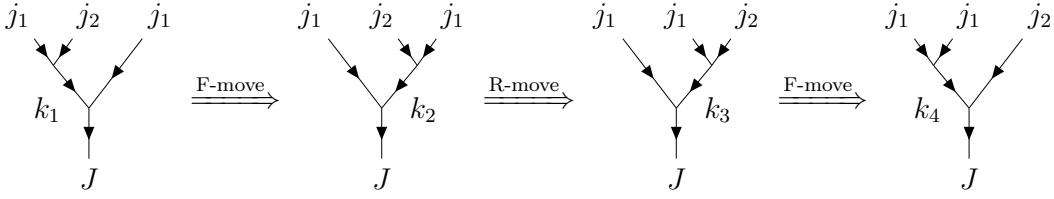


Back to the above question on multi-homogeneous polynomials, we see that

$$C_{j_1 k}^J (C_{j_1 j_2}^k(\psi_1, \psi_2), \psi_1) = \sum_k (2k+1)(-1)^{j_2 + j_3 + k + l} \left\{ \begin{matrix} j_1 & j_2 & k \\ J & j_1 & l \end{matrix} \right\} C_{j_2 l}^J (C_{j_1 j_1}^l(\psi_1, \psi_1), \psi_2).$$

Converting the third iterated CG-product to the first from above is more tricky, we illustrate by the following manipulation of fusion diagrams with R-move and F-move (summations are

omitted):



There is no need to compute 6j-symbols. Our aim is just verifying theoretically that any complicated multi-homogeneous polynomial written by iterated CG-products can be standardized as above. A precise standardization scheme is given in the next subsection (Definition 2.3).

2.3 Equivariant Universality with Fusion Blocks

We now prove the universality of fusion blocks. Let us first make a formal definition:

Definition 2.3. Let $V_j, W_{j'}$ be two irreps of $SU(2)$ and let ψ_1, \dots, ψ_n be spin- j vectors as inputs. Given a multi-degree $d = (d_1, \dots, d_n)$, we take fusion diagrams coupling each ψ_i by d_i times. Then we map the processed n vectors into a iterated CG-product. The whole fusion diagram is called a *standardized Clebsch-Gordan product for multi-homogeneous polynomial* of degree d and denoted by $C_{\mathbf{k}}^d(\psi_1, \dots, \psi_n)$ with \mathbf{k} being a collection of internal spins.

Remark. The above definition appoints a way to construct multi-homogeneous polynomials. It has two steps: (a) fusing/coupling each input repeatedly. Then (b) fusing the processed vectors together (without repetition). As a result of spin recoupling theory, any complicated multi-homogeneous polynomial can be written as a linear combination of these standardized forms with 6j-symbols. In the following context, when we talk about multi-homogeneous polynomials, we always refer the ones described in Definition 2.3. For further use, we would also employ the notation $(d_1, \dots, d_n) \vdash d$: (d_1, \dots, d_n) is called a *composition* of d as $d_1 + \dots + d_n = d$.

As an immediate result from Gordan theorem 2.2 and spin recoupling theory, we have:

Theorem 2.4. Any $SU(2)$ -equivariant polynomial $f : V_j^{\oplus n} \rightarrow W_{j'}$ can be written as a linear combination of standardized Clebsch-Gordan products $C_{\mathbf{k}}^d(\psi_1, \dots, \psi_n)$.

Note that in the above theorem, equivariant polynomials are defined on the direct sum of a single $SU(2)$ irrep V_j . Even we can consider more general cases as in Theorem 2.2, the special case always happens in a real learning task, e.g., n would represent the number of concerned atoms moving in the space and each of which is assigned a spin- j vector standing for its dynamical/chemical states. To introduce permutation equivariance when relabeling these atoms, we make the following a simple observation:

Lemma 2.5. Let $f : V_j^{\oplus n} \rightarrow W_{j'}$ be an S_{n-1} -invariant on its last $n-1$ variables. Then the function F defined by

$$F(\psi_1, \psi_2, \dots, \psi_n) = (f(\psi_1, \psi_2, \dots, \psi_n), f(\psi_2, \psi_1, \dots, \psi_n), \dots, f(\psi_n, \psi_1, \dots, \psi_{n-1})). \quad (5)$$

is S_n -equivariant from $V_j^{\oplus n}$ to $W_{j'}^{\oplus n}$. Moreover, any S_n -equivariant function is written in this form.

Proof. For first statement, let $\sigma \in S_n$, then by definition

$$F(\sigma \cdot (\psi_1, \psi_2, \dots, \psi_n)) = (f(\psi_{\sigma^{-1}(1)}, \psi_{\sigma^{-1}(2)}, \dots, \psi_{\sigma^{-1}(n)}), \dots, f(\psi_{\sigma^{-1}(n)}, \psi_{\sigma^{-1}(1)}, \dots, \psi_{\sigma^{-1}(n-1)})) \quad (6)$$

Since f is invariant on its last $n - 1$ variables,

$$f(\psi_{\sigma^{-1}(k)}, \psi_{\sigma^{-1}(2)}, \dots, \psi_{\sigma^{-1}(n)}) = f(\psi_{\sigma^{-1}(k)}, \psi_1, \dots, \psi_{\sigma^{-1}(k)-1}, \psi_{\sigma^{-1}(k)+1}, \dots, \psi_n) \quad (7)$$

for $k = 1, \dots, n$. Note that this identity should be revised when $\sigma^{-1}(k) = 1$ or n , but we omit these trivial details. On the other hand,

$$\sigma \cdot (F(\psi_1, \psi_2, \dots, \psi_n)) = \sigma \cdot (f(\psi_1, \psi_2, \dots, \psi_n), f(\psi_2, \psi_1, \dots, \psi_n), \dots, f(\psi_n, \psi_1, \dots, \psi_{n-1})) \quad (8)$$

and permuted vectors can be written as Eq.(7), which establishes the S_n -equivariance.

Now let $F = (f_1, \dots, f_n)$ be an arbitrary S_n -equivariant function. To prove the second part, we simply need to check the actions of transpositions $\sigma = (i, j) \in S_n$. For instance, acting on F by $\sigma = (1, 2)$, we have

$$\begin{aligned} F(\sigma \cdot (\psi_1, \psi_2, \dots, \psi_n)) &= (f_1(\psi_2, \psi_1, \dots, \psi_n), f_2(\psi_2, \psi_1, \dots, \psi_n), \dots, f_n(\psi_2, \psi_1, \dots, \psi_n)) \\ &= (f_2(\psi_1, \psi_2, \dots, \psi_n), f_1(\psi_1, \psi_2, \dots, \psi_n), \dots, f_n(\psi_1, \psi_2, \dots, \psi_n)) \\ &= (\sigma \cdot F)(\psi_1, \psi_2, \dots, \psi_n). \end{aligned} \quad (9)$$

This indicates that

$$f_1(\psi_2, \psi_1, \dots, \psi_n) = f_2(\psi_1, \psi_2, \dots, \psi_n), \quad \forall \phi \in V_j. \quad (10)$$

Applying (1, 3), we have

$$f_1(\psi_1, \psi_2, \psi_3, \dots, \psi_n) = f_3(\psi_3, \psi_2, \psi_1, \dots, \psi_n), \quad \forall \phi \in V_j. \quad (11)$$

Continuing by induction, we can conclude that

$$F(\psi_1, \psi_2, \dots, \psi_n) = (f(\psi_1, \psi_2, \dots, \psi_n), f(\psi_2, \psi_1, \dots, \psi_n), \dots, f(\psi_n, \psi_1, \dots, \psi_{n-1})). \quad (12)$$

To verify that f is S_{n-1} -invariant, we check Eq.(9) again which shows that $f_i(\psi_1, \dots, \psi_n) = f(\psi_i, \psi_1, \dots, \psi_n)$ is invariant when exchanging ψ_1 and ψ_2 for $i \geq 3$. Similar results hold for other transpositions and can be assembled to prove the invariance of f . \square

Remark. In the language from [24]), we are talking about first-order S_n -equivariant function defined from $V_j^{\oplus n}$ to $W_{j'}^{\oplus n}$ where S_n permutes input vectors. There are high order cases in which S_n would permute tensor with more than one indices. However, higher order equivariance is generally not applicable here as V_j, W_j are SU(2) irreps and do not carry a manifest S_n action.

We now prove the main theorem:

Theorem 2.6. *A neural network equipped with fusion blocks can be used to build any $SU(2) \times S_n$ equivariant polynomial from $V_j^{\oplus n}$ to $W_{j'}^{\oplus n}$.*

Proof. With Algorithm 1 presented in the main text, we label atoms by $1, \dots, n$ with atomic activations $(\psi_1, \dots, \psi_n) \in V_j^{\oplus n}$. Updating with fusion blocks, we are able to create $SU(2) \times S_n$ equivariant polynomials $F : V_j^{\oplus n} \rightarrow W_{j'}^{\oplus n}$ due to the employment of CG-products and aggregating at each layer. Any such function F can be expanded as Eq.(5) by Lemma 2.5 in which each component $f(\psi_i, \psi_1, \dots, \psi_n)$ is recorded with the i th atom being chosen as the center.

Since $f : V_j^{\oplus n} \rightarrow W_{j'}$ is both S_{n-1} -invariant on its last $n-1$ variables and $SU(2)$ -equivariant, by Theorem 2.4,

$$f(\psi_1, \dots, \psi_n) = \sum_{d, \mathbf{k}} c_{\mathbf{k}}^d C_{\mathbf{k}}^d(\psi_1, \dots, \psi_n), \quad (13)$$

where $c_{\mathbf{k}}^d$ denotes the expansion coefficients. Furthermore, $f(\psi_1, \dots, \psi_n)$ equals its symmetrization on the last $n - 1$ variables:

$$\begin{aligned} f(\psi_1, \dots, \psi_n) &= \frac{1}{(n-1)!} \sum_{\sigma \in S_{n-1}} f(\psi_1, \psi_{\sigma^{-1}(2)}, \dots, \psi_{\sigma^{-1}(n)}) \\ &= \frac{1}{(n-1)!} \sum_{d, \mathbf{k}, \sigma \in S_{n-1}} c_{\mathbf{k}}^d C_{\mathbf{k}}^d(\psi_1, \psi_{\sigma^{-1}(2)}, \dots, \psi_{\sigma^{-1}(n)}) \\ &= \sum_{\mathbf{k}} c_{\mathbf{k}}^d \bar{C}_{\mathbf{k}}^d(\psi_1, \psi_2, \dots, \psi_n), \end{aligned} \quad (14)$$

where we denote by $\bar{C}_{\mathbf{k}}^d(\psi_1, \psi_2, \dots, \psi_n)$ the symmetrization of $C_{\mathbf{k}}^d(\psi_1, \psi_2, \dots, \psi_n)$. To prove the universality, we just need to show that updating with fusion blocks produces symmetrized CG-products of the above kind, which can be done with the following procedure

1. For a given multi-degree $d = (d_1, \dots, d_n)$, we first process ψ_i by self-coupling (iterated CG-products) without aggregation as in Definition 2.3. We denote these pre-processed vectors by $C_{\mathbf{k}^j}^{d_j}(\psi_i)$ for $1 \leq i, j \leq n$.
2. We instantiate with the first atom being taken the center for reference. As an elementary example, assume $d = (d_1, d_2)$ with $d_i = 0$ for $i \geq 3$, then we apply the common CG-product, which corresponds to the simplest fusion diagram, on the central atom and each of its neighbor:

$$\sum_{i \geq 2} C^{k_{d_1}^1 k_{d_2}^2 J}(C_{\mathbf{k}^1}^{d_1}(\psi_1), C_{\mathbf{k}^2}^{d_2}(\psi_i)). \quad (15)$$

Note that the summation appears now as we aggregate all center-neighbor interaction/coupling. It is exactly the symmetrization of standardized CG-products.

3. Assume $d_i = 0$ for $i \geq 4$, we update the central atom one more time:

$$\sum_{j \geq 2} C^{J' k_{d_3}^3 J} \left(\sum_{i \geq 2} (C^{k_{d_1}^1 k_{d_2}^2 J'}(C_{\mathbf{k}^1}^{d_1}(\psi_1), C_{\mathbf{k}^2}^{d_2}(\psi_i))), C_{\mathbf{k}^3}^{d_3}(\psi_j) \right). \quad (16)$$

General multi-homogeneous polynomials with required equivariance are constructed sequentially and we complete the proof. \square

There is still one more subtle point need being clarify in the previous proof. It should be helpful to illustrate with the following example.

Example. Suppose $j = 0$, that is, atomic activations x_1, \dots, x_n are simply scalars and hence the SU(2)-equivariance holds trivially. We are left with the permutation equivariance. We still exemplify when the first atom is chosen as a center. Updating scalars with fusion blocks, we build polynomials as

$$M_{\lambda}(x_1, \dots, x_n) = \sum_{i_1, \dots, i_{n-1} \geq 2} x_1^{d_1} x_{i_1}^{d_2} \cdots x_{i_{n-1}}^{d_n}, \quad (17)$$

where λ denotes a composition (d_1, \dots, d_n) of the multi-degree d (see the previous Remark). As i_1, \dots, i_{n-1} can be arbitrary numbers greater than one, there are cases when $i_r = i_s$ for $r \neq s$. On the other hand, there are **monomial symmetric polynomials** on the last $n - 1$ variables:

$$m_{\lambda}(x_1, \dots, x_n) := \sum_{\sigma \in S_{n-1}} x_1^{d_1} x_{i_{\sigma^{-1}(1)}}^{d_2} \cdots x_{i_{\sigma^{-1}(n-1)}}^{d_n}, \quad (18)$$

which has no repetition on indices by definition. We note that

$$M_{\lambda=(d_1, d_2)} = m_{\lambda=(d_1, d_2)}, \quad M_{\lambda=(d_1, d_2, d_3)} = m_{\lambda=(d_1, d_2)} + m_{\lambda=(d_1, d_2, d_3)}, \dots \dots \quad (19)$$

It is well-known that monomial symmetric polynomials m_λ can be used to build any permutation-invariant polynomial [8]. Even with repeated indices, Eq.19 shows that our polynomials M_λ produced by fusion blocks also fulfill the task. This argument holds generally to the case when $j \neq 0$.

As a final step, we apply the so-called *Equivariant Stone-Weierstrass Theorem* which says that any continuous equivariant function can be approximated by equivariant polynomials [5] we conclude that:

Corollary 2.7. *The Cormorant architecture updated with fusion blocks permutation-equivariant and universal to approximate any continuous $SO(3)$ -equivariant map.*

3 Architecture, hyper-parameter, and computational details

Here, we provide additional details on the architecture used, hyperparameter choices, and computational details for our experiments.

Cormorant over fusion diagram neural network was modified from the publically available cormorant package available at <https://github.com/risilab/cormorant/>, with the sole difference being the addition of the fusion block to the Atomlevel class in the network. To compensate for the additional cost of the network we changed the number of layers in the Cormorant_CG class to 2 and set the number of channels to 8 or the QM9 experiments and 16 for the MD17 experiments. (The number of channels was chosen primarily to ensure the model fit in the memory of the GPUs used.) The training was performed on a single NVIDIA V100 GPUs at single precision, at 256 Epochs each with a mini-batch size of 64. All other training hyperparameters were chosen to be identical to the ones used in [1]. This resulted in a total training time of 12 hours for the MD17 calculations and 20 hours for the QM9 calculations.

MACE over fusion diagram neural network was modified from the publically available MACE package available at <https://github.com/ACESuit/mace>. While the MACE neural network was directly cloned for comparison. In non-adiabatic dynamics of stilbene simulations, we set the feature channel width of MoFD-sparse to 16 with $l = 0, 1, 2, 3$ without parity in the small-sample training, and we set the channel width to 32 in the training with more samples. For MACE, we set the feature channel width to 64 with $l = 0, 1, 2, 3$ without parity in the small-sample training, and we set the channel width to 128 in the training with more samples. For MoFD-dense, we set the feature channel width to 16 with $l = 0, 1, 2, 3$ without parity. In the training process of both MACE and MoFD, we set the maximal number of epochs to 500 with stopping patience of 30 epochs without improvement. The energy-to-force loss balance was enforced to 1:1000. The order of spherical harmonics is set to 3, while the size of the receptive field is set to 5 angstroms.

3.1 Electronic structure method

The fundamental electronic structure calculations in this study are performed using the Complete Active Space Self-Consistent Field (CASSCF) theory. CASSCF represents an advanced post-Hartree-Fock wavefunction-based method, which is particularly effective for systems with strong electron correlation effects. This method is distinguished by its approach to classifying

molecular orbitals into three distinct categories: inactive orbitals, active orbitals, and virtual orbitals.

Inactive orbitals are orbitals that are doubly occupied in all configurations and are thus excluded from the correlation treatment. The active orbitals form the 'active space' for the CASSCF calculations, which are a set of orbitals that can be occupied by a predefined number of electrons. The electrons in these orbitals are allowed to be distributed in all possible ways, leading to the generation of many electron configurations, thus capturing missing electron correlation effects in the Hartree-Fock calculation. Virtual orbitals are unoccupied in the reference state and lie above the active orbitals in energy. CASSCF iteratively optimizes both the molecular orbitals and the configuration interaction coefficients to achieve self-consistency.

CASSCF is particularly useful for studying excited states, transition states, and systems where traditional methods like Density Functional Theory (DFT) might fail due to strong electron correlation effects.

3.2 Dynamics details

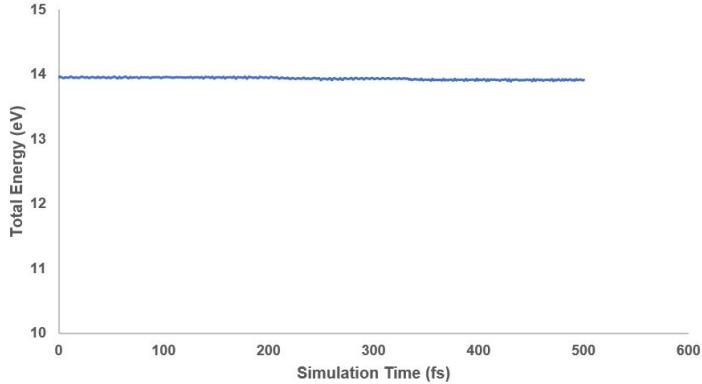


Figure 3: Illustration of conservation of energy over the dynamic simulation.

We used SHARC-MD package to run non-adiabatic dynamic simulations, the software is available at <https://github.com/sharc-md/sharc>. The compilation used SHARC-MD 2.1.2 with Intel One Compilers 2022 and Inter Math Kernel Library 2022. For SA-CASSCF calculations, OpenMolcas was used, which is available at <https://gitlab.com/Molcas/OpenMolcas>. The compilation used OpenMolcas 22.06 with Intel One Compilers 2022 and Inter Math Kernel Library 2022. Initial conformations are generated by Wigner-Sampling of the optimized ground-state structure with the same level of electronic structure method. For each conformation, a single-point calculation is performed to acquire the energy of states without spin-orbit calculations. To select initial excited-states, the MCH representation of the Hamiltonian is used to simulate delta-pulse excitation based on excitation energies and oscillators strengths with an excitation window of 0.0 to 10.0 eV. The threshold of total energy was set to 0.2 eV to account for inevitable active space rotation during the dynamic simulations. We initiated each trajectory with 0.5 fs as a single timestep and 500 fs as total length.

Table 1: Mean absolute error of various prediction targets on QM-9 (left) and conformational energies (in units of kcal/mol).

Target	Unit	Cormorant	Ours(CoFD)	
α	a_0^3	0.085 (0.001)	0.088 (0.003)	
$\Delta\epsilon$	eV	0.061 (0.005)	0.062 (0.001)	
ϵ_{HOMO}	eV	0.034 (0.002)	0.0391 (0.0008)	
ϵ_{LUMO}	eV	0.038 (0.008)	0.0347 (0.0006)	
μ	D	0.038 (0.009)	0.035 (0.001)	
C_v	$\frac{\text{cal}}{\text{mol K}}$	0.026 (0.000)	0.0272 (0.0006)	
G	eV	0.020 (0.000)	0.0135 (0.0002)	
H	eV	0.021 (0.001)	0.0132 (0.0004)	
R^2	a_0^2	0.961 (0.019)	0.50 (0.02)	
U	eV	0.021 (0.000)	0.0130 (0.0004)	
U_0	eV	0.022 (0.003)	0.0133 (0.0003)	
ZPVE	meV	2.027 (0.042)	1.43 (0.04)	

Table 2: QM9 results on the Dimenet split. Benchmarks taken from [6].

Target	Unit	SchNet	MGCN	DeepMoleNet	DimeNet	DimeNet ⁺⁺	Ours(CoFD)
μ	D	0.0330	0.0560	0.0253	0.0286	0.0297	0.028
α	a_0^3	0.235	0.0300	0.0681	0.0469	0.0435	0.0682
ϵ_{HOMO}	meV	41.0	42.1	23.9	27.8	24.6	37.8
ϵ_{LUMO}	meV	34.0	57.4	22.7	19.7	19.5	26.6
$\Delta\epsilon$	meV	63.0	64.2	33.2	34.8	32.6	53.0
$\langle R^2 \rangle$	a_0^2	0.073	0.110	0.680	0.331	0.331	0.4128
ZPVE	meV	1.70	1.12	1.90	1.29	1.21	1.304
U_0	meV	14.0	12.9	7.70	8.02	6.32	10.4
U	meV	19.0	14.4	7.80	7.89	6.28	10.8
H	meV	14.0	14.6	7.80	8.11	6.53	10.6
G	meV	14.0	16.2	8.60	8.98	7.56	11.1
c_v	$\frac{\text{cal}}{\text{mol K}}$	0.0330	0.0380	0.0290	0.0249	0.0230	0.0242

4 Additional Numerical Results

Here, we give additional numerical results, complementing the ones given in the main text. First, give a more comprehensive treatment of our results on QM9. Whereas in the main text we gave results for QM9 on only a single target, in 1 we give the mean and standard deviation of our results over multiple replicates. Specifically, we give the mean and standard deviation over 3 replicates for $\Delta\epsilon$, ϵ_{HOMO} , ϵ_{LUMO} , Cv , G , R^2 , U and over 4 replicates for all other targets. To compare our work against other architectures we then trained this same model against the split of QM-9 given in the [7]; results are given in Table 2. We find that our model performs reasonably well, although it does not achieve state-of-the-art performance. However, we note that our ability to perform hyperparameter tuning was limited due to the expense of running the model and our comparative compute budget. We hope that subsequent hyperparameter tuning would improve the accuracy of our models.

In addition, we trained a variant of a model with the fusion block against the MD17 split introduced in [7]. This required making architectural changes compared with the original cormorant architecture: as implemented, the cormorant code evaluates radial functions on the displacements between an atom and itself. However, the original radial functions used are not

Table 3: Mean absolute testing error for the energies and forces for cormorant with a fusion block and various comparative models in the literature. Results taken from [2]

Molecule	Target	Unit	SchNet	DimeNet	NequIP (l=3)	Ours
Aspirin	Energy	meV	16.0	8.8	5.7	11.1
	Force	meV/ \AA^2	58.5	21.6	8.0	23.72
Ethanol	Energy	meV	3.5	2.8	2.2	2.6
	Force	meV/ \AA^2	16.9	10.0	3.1	9.4
Malonaldehyde	Energy	meV	5.6	4.5	3.3	4.7
	Force	meV/ \AA^2	28.6	16.6	5.6	20.4
Naphthalene	Energy	meV	6.9	5.3	4.9	8.1
	Force	meV/ \AA^2	25.2	9.3	1.7	21.4
Salicylic Acid	Energy	meV	8.7	5.8	4.6	4.2
	Force	meV/ \AA^2	36.9	16.2	3.9	8.9
Toluene	Energy	meV	5.2	4.4	4.0	6.0
	Force	meV/ \AA^2	24.7	9.4	2.0	16.6
Uracil	Energy	meV	6.1	5.0	4.5	3.7
	Force	meV/ \AA^2	24.3	13.1	3.3	6.9

differentiable at 0. Consequently, we instead used the radial functions

$$R(x) = \frac{x^2}{a} e^{-x^2/a} \quad (20)$$

where a is a collection of learned parameters, initially evenly spaced on the interval $(0, 3]$. As before, each network was made of two neural network layers, this time with 64 channels for each layer. The network was trained on a linear combination of the mean square error in the energy and the force, with the total force error averaged over atoms and multiplied by a constant factor. The constant factor was chosen from 1, 10, and 100 by taking the value with the minimal validation error. Training proceeded on an NVIDIA A100 GPU for 2024 epochs, with a batch size of 64. We give the results in table 3.

References

- [1] Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks, 2019.
- [2] 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, 2021.
- [3] L. C. Biedenharn, A. Giovannini, and J. D. Louck. Canonical definition of wigner coefficients in un. *Journal of Mathematical Physics*, 8(4):691–700, 1967.
- [4] Genevieve Dusson, Markus Bachmayr, Gabor Csanyi, Ralf Drautz, Simon Etter, Cas van der Oord, and Christoph Ortner. Atomic cluster expansion: Completeness, efficiency and stability, 2019.
- [5] Michael J Field. *Dynamics and Symmetry*, volume 3 of *ICP Advanced Texts in Mathematics*. Imperial College Press, September 2007.

- [6] Johannes Gasteiger, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.
- [7] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs, 2020.
- [8] Roe Goodman and Nolan R. Wallach. *Symmetry, Representations, and Invariants*. Springer New York, 2009.
- [9] Paul Gordan. Beweis, dass jede Covariante und Invariante einer binären Form eine ganze Function mit numerischen Coefficienten einer endlichen Anzahl solcher Formen ist. *Journal für die Reine und Angewandte Mathematik*, 1868(69):323–354, January 1868.
- [10] David Hilbert and Bernd Sturmfels. *Algebraic invariants*. Cambridge mathematical library. Cambridge University Press, Cambridge [England] ; New York, NY, USA, 1993.
- [11] Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, January 2006.
- [12] Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials, 2018.
- [13] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-Gordan Nets: a fully Fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [14] Ingvar Lindgren and John Morrison. *Atomic Many-Body Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.
- [15] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.
- [16] Pierre Martin-Dussaud. A primer of group theory for Loop Quantum Gravity and spinfoams. *General Relativity and Gravitation*, 51(9):110, September 2019.
- [17] Marc Olive. About Gordan’s algorithm for binary forms, March 2014.
- [18] Peter J. Olver. *Classical Invariant Theory*. London Mathematical Society Student Texts. Cambridge University Press, 1999.
- [19] Roger Penrose. Angular momentum: an approach to combinatorial space-time. In *Quantum Theory and Beyond*, pages 151–180. Cambridge University Press, Cambridge, 1971.
- [20] Claudio Procesi. *Lie groups: an approach through invariants and representations*. Universitext. Springer, New York, 2007. OCLC: ocm61309419.
- [21] Philipp Schmoll and Román Orús. Benchmarking global $su(2)$ symmetry in two-dimensional tensor network algorithms. *Phys. Rev. B*, 102:241101, Dec 2020.
- [22] Philipp Schmoll, Sukhbinder Singh, Matteo Rizzi, and Román Orús. A programming guide for tensor networks with global $su(2)$ symmetry. *Annals of Physics*, 419:168232, Aug 2020.
- [23] Sukhwinder Singh and Guifre Vidal. Tensor network states and algorithms in the presence of a global $su(2)$ symmetry. *Physical Review B*, 86(19), Nov 2012.

- [24] Erik Henning Thiede, Truong Son Hy, and Risi Kondor. The general theory of permutation equivariant neural networks and higher order graph variational encoders, 2020.
- [25] D A Varshalovich, A N Moskalev, and V K Khersonskii. *Quantum Theory of Angular Momentum*. WORLD SCIENTIFIC, October 1988.