Group-Equivariant Neural Networks with Fusion Diagrams

Zimu Li*
DAMTP, Center for Mathematical Sciences, University of Cambridge, Cambridge CB30WA, UK

Han Zheng*
Department of Computer Science, The University of Chicago, Chicago, IL 60637, USA
Department of Statistics, The University of Chicago, Chicago, IL 60637, USA

Erik Thiede*
Center for Computational Mathematics, Flatiron Institute, New York NY 10010

Junyu Liu
Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA
Kadanoff Center for Theoretical Physics, The University of Chicago, Chicago, IL 60637, USA
qBraid Co., Harper Court 5235, Chicago, IL 60615, USA

Risi Kondor
Department of Statistics, The University of Chicago, Chicago, IL 60637, USA
Department of Computer Science, The University of Chicago, Chicago, IL 60637, USA
Flatiron Institute, New York City, NY 10010, USA

Abstract

Many learning tasks in physics and chemistry involve global spatial symmetries as well as permutational symmetry between particles. The standard approach to such problems is equivariant neural networks, which employ 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 increases, the bookkeeping associated with ensuring parsimony as well as equivariance quickly becomes nontrivial. In this paper, we propose to use fusion diagrams, a technique widely used in simulating SU(2)-symmetric quantum many-body problems, to design new equivariant components for use in equivariant neural networks. This yields a diagrammatic approach to constructing new neural network architectures. We show that when applied to particles in a given local neighborhood, the resulting components, which we call *fusion blocks*, are universal approximators of any continuous equivariant function defined on the neighborhood. As a practical demonstration, we incorporate a fusion block into a pre-existing equivariant architecture (Cormorant) and show that it improves performance on benchmark molecular learning tasks.

$*$: Equal Contributions

1 Introduction

For many machine learning tasks in the natural sciences, it is paramount that learning algorithms respect fundamental physical symmetries. For instance, important learning targets for molecular data such as the chemical system's total potential energy are invariant to both global rotations of the

system as well as a permutation of the system's internal components [1–7]. Any machine learning algorithm used to predict these quantities must give the same answer independent of datapoints' rotational pose or ordering [8].

A popular approach to addressing this type of problem is the use of *group-equivariant neural networks* [9, 10]. In a group-equivariant network, symmetry operations on the data, such as rotations or permutations commute with the network's layers. Colloquially, this means that if we apply a rotation to the inputs of a rotation-equivariant neural network, all of the interior activations in the neural network must rotate accordingly. Similarly, if we were to shuffle the inputs to a permutation-equivariant neural network, the same permutation would act on the neural network's activations. When learning on particle-based systems such as molecular configurations, we must often account for both spatial *and* permutational symmetries, since we can not only rotate and translate the system, but also permute the particle labels. In order to address this, most group-equivariant networks account for translations by working only with the relative displacements between particles. Rotational degrees of freedom are encoded in activations that transform under an action of the three-dimensional rotation group. Conditions for constructing rotation-equivariant linear operations on these activations are given in e.g., [11], and such neural networks often use nonlinearities derived by taking various tensor products between parts of the representation. However, the additional permutational degrees of freedom considerably complicate the information flow. For instance, each layer of our neural network may have several sets of activations, corresponding to individual particles, pairs of particles, or many-body interactions [2, 5, 12, 13]. Taking all possible products quickly becomes unfeasibly expensive. Consequently, there is a need to develop new principles to organize the information flow in the construction of such neural networks.

In this paper, we address this combinatorial complexity by borrowing techniques from theoretical physics, where physicists have developed a system for organizing tensor products using diagrams known as *fusion diagrams*. First developed by Penrose to study quantum gravity [14–16] and later adopted in tensor networks with symmetries [17–19], fusion diagrams concisely summarize the space of possible tensor products, allowing us to classify higher-order composition schemes between tensor products.

Consequently, they can be used not only to classify existing neural network architectures, but also to inspire new equivariant blocks. Using classical results from invariant theory, we demonstrate that the resulting blocks are expressive. Specifically, we show that under certain conditions, these blocks achieve universality. As a specific example, we use a fusion diagram to construct a new SO(3)-equivariant layer, which we incorporate into Cormorant [20], a neural network architecture designed for molecular learning problems. We show that employing the new layer noticeably improves the performance of Cormorant for a comparable number of parameters.

2 Related work

Our approach to constructing neural networks using fusion diagrams draws together several independent lines of research. Here, we review related work in both machine learning and physics.

2.1 Machine Learning for Particle-based Systems

Recent years have seen an explosion of interest in the development of machine learning approaches to learning on point clouds, with a particular focus on learning the properties of molecular systems based on their three-dimensional configurations. It is often crucial that the learned properties are strictly invariant to the rotation of the point cloud. One popular, and arguably the first, approach is to construct rotationally invariant descriptors of particle neighborhoods. These can be incorporated into off-the-shelf machine learning approaches such as linear regression [21], Gaussian processes [22–24], feed-forward neural networks [8, 25], or graph neural networks [4, 26–29].

At the same time, the success of convolutional neural networks in image processing, coupled with the observation that convolutional neurons are equivariant to the translation group, has inspired considerable interest in constructing neural networks that are equivariant to other groups. While initial work focused on applications related to image recognition [30–33], it was soon realized that these approaches could be extended to point clouds [12, 13, 20, 34]. Equivariant neural networks are some of the leading architectures for learning on molecular data [3], and have been successfully applied to the Lorentz group for applications in particle physics [35]. In particular, in the application of point clouds, rotational equivariance has been applied to graph neural networks. There has also been extensive work on the construction and expressive power of equivariant graph neural networks [36, 37],

which would serve as general conditions for group equivariance of the linear transformations were discussed in [11, 38] and it was proven that universal approximation for sets of elements transforming under a global group could be obtained by composing $\mathbb{S}_n$-equivariant universal approximators on rotation- and group-equivariant approximators [37, 39].

2.2 Tensor networks with global symmetry

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 of a larger tensors by 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 [40, 41] and introducing low-rank tensors to represent part of the whole quantum states [42]. Applications of tensor networks include quantum simulation of quantum physics problems [43–45], quantum computing and quantum supremacy experiments [46, 47], machine learning and data science [48–50], and quantum gravity [51, 52]. The so-called *fusion diagrams* used in this paper are special types of tensor networks with global SU(2) symmetry. They are originally used to simulate SU(2)-symmetric quantum system [17–19, 42] and we will show their potential in constructing expressive equivariant neural networks.

3 Background

Before presenting the main contributions of our work, we give a brief review of equivariant neural networks and the use of fusion diagrams in physics.

3.1 Equivariant neural networks

Neural networks transform their inputs by repeatedly applying learned affine functions followed by fixed nonlinear functions between layers. When the learning problem is subject to symmetry, the choice of these functions is subject to some constraints. This is formalized by the concept of *group equivariance*. Let X and Y be the input and output spaces of a layer L, and let T and T' be the action of a group $\mathbb{G}$ encoding the symmetry on X resp. Y. The layer is said to be equivariant to $\mathbb{G}$ if

$$T'_g \circ L = L \circ T_g \qquad \forall g \in \mathbb{G}, \tag{3.1}$$

In general, the action of the group on the input and output space need not be the same. As a specific example, consider a layer that is equivariant to the group of three-dimensional rotations $\mathrm{SO}(3)$. If we were to apply a rotation to the input, the output must rotate the same way. If the group action on the output space is the identity transformation, or when $T'_g y = y$ for all elements of $\mathbb{G}$, we reduce to the case of *invariant layer*:

$$L = L \circ T_g \qquad \forall g \in \mathbb{G}. \tag{3.2}$$

Many key learning targets have group symmetries. For instance, chemistry properties such as the energy of a molecular system is invariant to rotation or permutation of the atomic labels. In these cases, it is common to enforce the weaker condition of equivariance in all but the last layer, only enforcing invariance in the final layer.

Constructing an equivariant neural network requires that both the learned affine function and the fixed nonlinear function obey equivariance. [11] 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 representation[1] of a compact or finite group G is pair (V, ρ) with $\rho : G \to \mathrm{End(V)}$ such that $\forall g_1, g_2 \in G, \rho(g_1)\rho(g_2) = \rho(g_1 g_2)$. In this work, we will exclusively study cases where $V = \mathbb{C}^k$ for some finite k. In this case, the range of ρ is a subset of the space of complex $k \times k$ matrices. As an example consider the case where G is the three-dimensional rotation group $\mathrm{SO}(3)$ and V is the space of three-dimensional vectors in Cartesian space. Then, for every rotation $g \in \mathrm{SO}(3)$, $\rho(g)$ is corresponding rotation matrix.

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 [11, 30, 36, 53–55]. One approach that has seen success in scientific application is to make explicit use of irreducible linear representations. An irreducible linear representation (irrep) is a representation where V has no proper subspaces that are preserved under ρ. Using well-known results in group theory, we can

[1] Linear representations should not be confused with the different use of the word "representation" in representation learning.

apply a transformation that decomposes the inputs, outputs, and activations of a neural network into components that transform according to the group's irrep. Then, one can show that the most general possible equivariant linear transformation can be written as matrix multiplication against each individual component.

On the other hand, constructing an equivariant nonlinearity has its own difficulties. Some of the most commonly used nonlinearities, such as hyperbolic tangent or ReLU functions, would break equivariance. Instead, a common technique used to construct equivariant nonlinearities is to employ tensor products of activations, exploiting the fact that tensor products of group representations are themselves group representations. The decomposition tensor product representations, in physics and chemistry literature, is often referred as angular momentum coupling such as spin-orbital coupling. We will refer to this equivariant non-linearity as the *CG-product* [33], which we briefly introduce in the following.

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

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 $g \in \mathrm{SU}(2)$ 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. 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 the transformation manner, 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 these basic notions being clarified, let us define CG-product formally. 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 and can be again 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* $g \in \mathrm{SU}(2)$. Formally,

$$C^{(j_a,j_b,j_c)\dagger}\big(W^{j_a}(g) \otimes W^{j_b}(g)\big)C^{(j_a,j_b,j_c)} = W^{j_c}(g) \;\text{ for all } g \in \mathrm{SU}(2)$$

By definition, a CG-decomposition is equivariant with the action of SU(2) as well as SO(3).

There are well-established methods to compute $C^{(j_a,j_b,j_c)}$ both theoretically and algorithmically, e.g., [56–58]. Summing in Einstein notation with lower and upper indices, we would like to call it a CG-product for input vectors

$$\psi^{(j_c)m_c} = C^{(j_a,j_b,j_c)}{}_{m_a,m_b}{}^{m_c}\psi^{(j_a)m_a}\psi^{(j_b)m_b} \tag{3.3}$$

We will also leave m as entries indices and write

$$\psi^{(j_c)} = \langle C^{(j_a,j_b,j_c)}, \psi^{(j_a)} \otimes \psi^{(j_b)} \rangle \tag{3.4}$$

3.3 Fusion diagrams

Clebsch-Gordan products appear naturally in quantum mechanics out of necessity to describe the interaction of multiple particles. For even moderately complex quantum mechanical systems, equations keeping track of a successive products will become formidably complicated. Consequently, physicists employ diagrams that represent the succession of tensor products and Clebsch-Gordan decompositions in pictorial form. We refer to these diagrams as *fusion diagrams*[2].

The space of fusion diagrams that fuse n incoming SU(2)-contravariant features $\{\psi^{(j_i)}\}_{i=1}^n$ into a single outgoing SU(2)-contravariant feature $\Psi^{(J)}$, comprised of successive Clebsch-Gordan products, is denoted by $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes \cdots \otimes V_{j_n}, V_J)$ as depicted in Fig.1. By choosing a fusion

[2]We note that the quantum gravity literature also has a related concept of a *spin network*, which is a certain "integration" on *fusion* and *split* diagrams[14, 15, 59]. These should not be confused with the notion of neural networks that operate on spins, or use spin-like concepts..

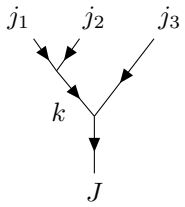


Figure 1: A fusion diagram as a standard basis element in $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes V_{j_2} \otimes V_{j_3}, V_J)$. Such fusion diagrams with all possible values k are also called Wigner 4-jm elements.

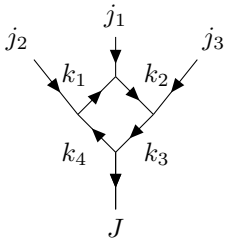


Figure 2: A fusion diagram as some element in $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes V_{j_2} \otimes V_{j_3}, V_J)$.

diagram $Q^{(j_1,\dots,j_n;J)} \in \mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes \cdots \otimes V_{j_n}, V_J)$ the fusion process can be computed by the following inner products:

$$\psi^{(J)} = \langle Q^{(j_1,\cdots,j_n;J)}, \psi^{(j_1)} \otimes \cdots \otimes \psi^{(j_n)} \rangle. \tag{3.5}$$

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. Each vertex represents a CG-product $C^{(j_a,j_b,j_c)}$ and is hence connected to no more than three edges[3]. We adopt the following convention in labeling fusion diagrams which have n external incoming edges and one external outgoing edge (also see Section 4).

1. Each fusion diagram has a list of external spin labels $\{j_1, j_2, ..., j_n\} \equiv \mathbf{j}$, and a list of internal spin labels $\{k_1, ..., k_m\} \equiv \mathbf{k}$. These are referred to as external and internal spin configurations respectively
2. For each vertex, we employ a CG-product reading clockwise around the node. That is, the top left vertex in Figure 1 denotes $C^{(j_1 j_2 k_1)}{}_{m_{j_1}, m_{j_2}}{}^{m_{k_3}}$. Incoming indices are lowered covariantly while outgoing indices are raised contravariantly.
3. The spins associated with each vertex must obey the Clebsch-Gordan rule: denoting the two incoming edges as j_a and j_b and the outgoing edge as j_c, we require that $|j_a - j_b| \leq j_c \leq j_a + j_c$. [56, 58].
4. We label the final outgoing edge with J so that incoming n SU(2)-equivariant features are fused into a SU(2)-equivariant feature to be processed in the next layer.

For instance, for the fusion diagram in Figure 1 is

$$Q^{(\mathbf{j},\mathbf{k};J)}{}_{m_{j_1} m_{j_2} m_{j_3}}{}^{m_J} \equiv M C^{(j_1 j_2 k)}{}_{m_{k_1} m_{j_2}}{}^{m_k} C^{(k j_3 J)}{}_{m_k m_{j_3}}{}^{m_J}, \tag{3.6}$$

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

In the discussion that follows, we will refer to any fusion diagrams that form a basis in the invariant Hom set as *Wigner n-jm symbol* where n counting 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 result in Section 4. For a fixed set of external spin labels $\mathbf{j}$, varying all possible assignments of $\mathbf{k}$ of (3.6) enumerates all possible orthogonal basis elements in $\mathrm{Hom}_{\mathrm{SU}(2)}(V_1 \otimes V_2 \otimes \cdots \otimes V_n, U)$ [15, 21, 58]. In neural network language, different configurations of $\mathbf{k}$ give rise to mutually orthogonal ways to perform higher-order CG-products.

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

$$Q^{(\mathbf{j},\mathbf{k};J)}{}_{m_{j_1} m_{j_2} m_{j_3}}{}^{m_J} = C^{(j_1,k_2,k_1)}{}_{m_{j_1}}{}^{m_{k_2}}{}_{m_{k_1}} C^{(j_2 k_1 k_4)}{}_{m_{j_2}}{}^{m_{k_1}}{}_{m_{k_4}} C^{(j3,k_3,k_2)}{}_{m_{j_3}}{}^{m_{k_3}}{}_{m_{k_2}} C^{(J,k_4,k_3)}{}_{m_J}{}^{m_{k_4}}{}_{m_{k_3}}, \tag{3.7}$$

[3] One can also assume that vertices connected with more than three edges are sub-fusion diagrams with vertices having only no more than three edges. We refer the interested audience to [17–19] for more details. Note for general tensor networks without prescribed symmetry, there is no such restriction

can be expressed as a linear combination of Wigner-4jm tensors in (3.6) with different internal spin configuration $\mathbf{k}$ because they form a basis for all such high-order CG-products. [4] The fusion diagram in (3.7) can be expressed as linear combinations of Wigner 4-jm symbols in (3.6) by varying the internal spin configuration k. 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 helps build practically useful universal neural network architectures (See in Theorem 4.2). To motivate the use of more general fusion diagrams such as Fig.2, not only does it spans multiple Wigner symbols at once, it can encode non-trivial local graphical invariance such as the $C4$ group invariance, when choosing appropriate $\mathbf{j}$ and $\mathbf{k}$. This connection between fusion diagrams and local graphical invariance naturally leads to the interesting idea to generalize the equivariant graph neural nets beyond the message-passing paradigm and may be suitable to other more sophisticated yet chemically interesting data such as the metal-organic (MOF) framework [60, 61]. Exploring the use of fusion diagrams as a pathway to achieve higher graphical expressivity for more specific chemical molecules is anticipated for our forthcoming publication.

4 Fusion Diagrams in Neural Networks

Despite numerous applications of fusion diagrams in various domains in physics and neural quantum states, its importance as diagrammatic guidance to construct neural network components has not been explored. The main contribution of this paper is to exploit fusion diagrams as blueprints to construct new, expressive components for group-equivariant networks. Under appropriate conditions, the resulting neural network components admit universal approximation.

4.1 Neural Fusion Blocks

Here, we demonstrate how fusion diagrams can be used to design equivariant layers for point cloud data. 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. With the Einstein summation convention, inputs and outputs both transform contravariantly with raised indices. We denote the collection of input tensors associated with incoming edge a as $\Psi^{(\mathrm{in})}$ and components corresponding to the spin label j_a are denoted as $\psi^{(\mathrm{in},j_a)}$. The tensor's τ'th channel index is denoted by $\Psi^{(\mathrm{in}),j_a}_{\tau}$; we omit batch indices from our treatment for brevity. The fusion block then acts according to Algorithm 1. Due to the use of fusion diagrams, the resulting algorithm is guaranteed to be equivariant to rotations.

Fusing n atoms naively in practice would entail exponential scaling with the number of products needed to perform. For instance, if each atom stores information on $j = 0, 1, 2$, then a naive fusion that coupled every single combination of spin configurations would require 3^n many Clebsch-Gordan products, quickly becoming inaccessible. However, this computational dilemma can be efficiently and elegantly salvaged by drawing some fusion diagrams Q by supplying specific sets of internal spin configurations $\mathbf{k}$, for which we pass into hyper-parameters. Nevertheless, we caution that the injudicious use of fusion blocks can drastically increase computational costs.

4.2 Equivariant universal approximation theorems

Rotational equivariant neural architectures affording universal approximation on point clouds are analyzed in [37]. We here 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 aims at studying invariant and equivariant polynomials of classical groups [62–64]. We introduce this viewpoint for interested reader in the Appendix and in the technical proofs for the above theorems. Similar universality statements to the ones we present have also been given in [21]. We pursue a different proof strategy, using iterated CG-products (Theorem 4.1) mentioned in Section 3 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 CG-products are *the only* essential ingredients to construct any SU(2)-equivariant function [62, 63, 65]. To illustrate this point rigorously, we adapt one of the most relevant results due to Gordan [62, 66, 67] with the language of angular momentum recoupling theory [15, 58, 68] and representation theory.

[4] For readers familiar with matrix product states, we observe that (3.7) is a matrix product state with boundary conditions used in tensor networks.

Algorithm 1: Fusion Block

Data: Incoming atomic activations: $\{\Psi_1^{(\text{in})}, \cdots \Psi_n^{(\text{in})}\}$
Result: Outgoing atomic activation $\Psi^{(\text{out})}$
Pick a set of spin configurations and construct a corresponding collection of fusion diagrams $\mathcal{Q} = \{Q^1, ..., Q^p\}$ having n external incoming edge and 1 outgoing edge;
for $Q_i^{(\mathbf{j},\mathbf{k};J)} \in \mathcal{Q}$ **do**
Extract parts of the incoming activation that transform according to Q_i's spin labels, $\psi_1^{(\text{in},j_1)}, \dots \psi_n^{(\text{in},j_n)}$. Fuse Q with incoming activations and (optionally) apply an aggregation function ϕ to enforce permutation equivariance:

$$\widetilde{\psi}_i^J = \phi(\langle Q_i^{(\mathbf{j},\mathbf{k};J)}, \psi_1^{(\text{in},j_1)} \cdots \psi_n^{(\text{in},j_n)} \rangle) \tag{4.1}$$

end
Concatenate outputs for each fusion diagram along the new channel dimension to record the number of spin configurations we used and reshape them into the channel dimension $\tilde{\tau}_c$

$$\tilde{\Psi}_{\tilde{\tau}_c} = \tilde{\Psi}_1 \oplus \cdots \oplus \tilde{\Psi}_n \tag{4.2}$$

Multiply by a linear mixing matrix.

$$\Psi_{\tau_c}^{(\text{out})} \leftarrow \tilde{\Psi}_{\tilde{\tau}_c} W^{\tilde{\tau}_c}{}_{\tau_c} \tag{4.3}$$

Theorem 4.1. *We define the* standardized CG-product for multi-homogeneous polynomial *of size n to be the fusion diagram corresponding to the product*

$$MC^{(j_1 j_2 k_1)}{}_{m_{j_1} m_{j_2}}{}^{m_{k_1}} \left(\prod_{i=1}^{n-3} C^{(k_i j_{i+2} k_{i+1})}{}_{m_{k_i} m_{j_{i+2}}}{}^{m_{k_{i+1}}} \right) C^{(k_{n-2} j_4 J)}{}_{m_{k_{n-2}} m_{j_n}}{}^{m_J}, \tag{4.4}$$

Any SU(2)-equivariant polynomial $f : V_j^{\oplus n} \to U_{j'}$ can be written as a linear combination of fusion process over standardized CG-products for multi-homogeneous polynomials.

The above theorem says that any SU(2)-equivariant polynomial can be constructed from CG-products. In the supplementary information we show, using an equivariant version of the Stone-Weierstrass theorem [69], 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. Graphical neural networks (GNNs) in practice perform local graphical computation; 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 on a local neighborhood, assuming that local graph is fully connected. This is a standard and reasonable assumption in point-clouds where points clouds interact with each other based on the radial distance. If computation is global, this amounts to universality for the entire neural network. 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 these of the global system. Intuitively, as fusion blocks fuse inputs on each atom with the center and then aggregate (Fig.3), iterated CG-products will be built with permutation equivariance after updating several layers and hence approximate any rotation and permutation equivariant map finally (Details can be seen in the SM):

Corollary 4.2 (Global Rotation and Permutation Equivariant Universality)**.** *The Cormorant architecture (see Section 5) updated with fusion blocks is universal to approximate any continuous SO(3) $\times S_n$-equivariant map if every atomic neighborhood contains all atoms in the molecule. When summing at the output layer, it recovers $SO(3)$-equivariant and S_n-invariant universality.*

Note that the above theorem holds for both SU(2) and SO(3) equivariance. As the learning tasks exhibited in this paper draw classical data from atomic systems, we concretize the theorem with classical SO(3) rotational equivariance.

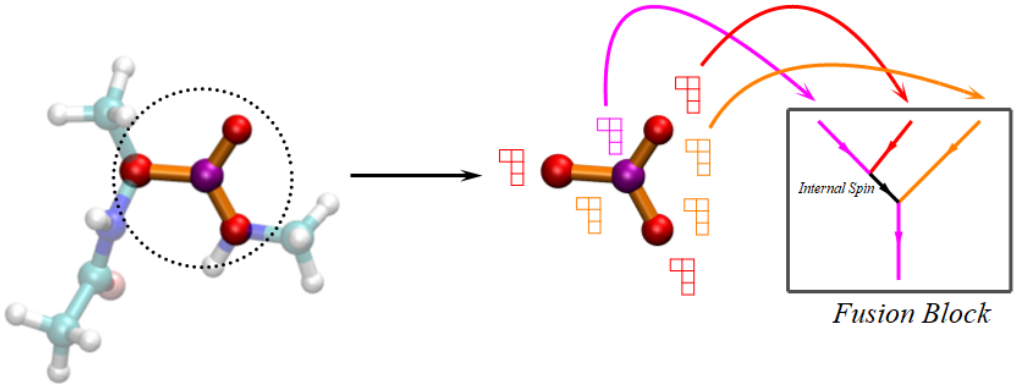


Figure 3: Schematic illustration of the fusion block. For each atom the fusion block first fuses all the neigboring atoms given radius cut-off by pre-selected fusion diagram templates. Specifically, for each neighboring atom, we fuse the information from root, neighbor atom, and their connecting edge. Then the fusion block applies an aggregation method–in this work, we simply sum all the neighbors.

5 Experimental results

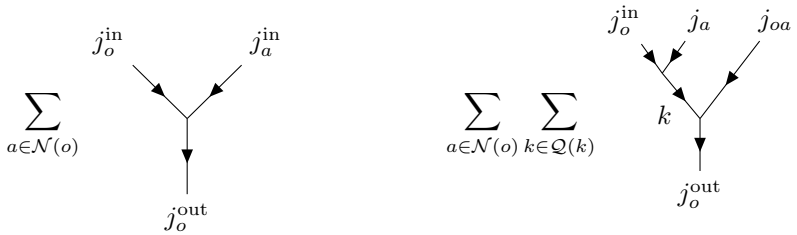
By operating via tensor diagrams, fusion blocks automatically obey equivariance to SO(3). Applying these blocks to point clouds requires ensuring that the resulting neural network obeys permutation symmetry. However, the fact that each fusion diagram has a single output means that they can easily be passed through an aggregation function and incorporated into existing message-passing-like mechanisms. For concreteness, assume that each possible input to the block was associated with either a single particle i, in which case we denote it as $\Psi^{(\text{in,i})}$ or pairs of particles.

5.1 An example application of fusion blocks.

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 an existing neural network. Specifically, we choose to augment the Cormorant architecture proposed in [20] with one additional block derived using a fusion diagram. A neuron in Cormorant layer s operates as follows:

$$F_i^{s-1} = \Big[\underbrace{F_i^s \oplus \left(F_i^{s-1} \otimes_{\text{cg}} F_i^{s-1}\right)}_{\text{one-body part}} \oplus \underbrace{\Big(\sum_j \Big(Y(\mathbf{x}_{ij}) \otimes_{\text{cg}} F_j^{s-1} \Big) g_{ij} \Big)}_{\text{two-body part}} \Big] \cdot W_{s,\ell}^{\text{vertex}}. \tag{5.1}$$

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. (We refer the reader to [20] for the precise functional form of g_{ij}.) Each of these terms corresponds to the two-body diagram above, corresponding to the 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.

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.

	Cormorant		Ours
α (bohr^3)	0.085	(0.001)	0.086
$\Delta\epsilon$ (eV)	0.061	(0.005)	0.061
ϵ_{HOMO} (eV)	**0.034**	**(0.002)**	0.039
ϵ_{LUMO} (eV)	0.038	(0.008)	0.034
μ (D)	0.038	(0.009)	0.036
C_v (cal/mol K)	**0.026**	**(0.000)**	0.0270
G (eV)	0.020	(0.000)	**0.0134**
H (eV)	0.021	(0.001)	**0.013**
R^2 (bohr^2)	0.961	(0.019)	**0.5028**
U (eV)	0.021	(0.000)	**0.0131**
U_0 (eV)	0.022	(0.003)	**0.0131**
ZPVE (meV)	2.027	(0.042)	**1.42**

	Cormorant	Ours
Aspirin	0.098	**0.0951**
Ethanol	0.027	**0.0241**
Malonaldehyde	0.041	**0.0380**
Naphthalene	**0.029**	0.0321
Salicylic Acid	0.066	**0.0608**
Toluene	0.034	**0.0316**
Uracil	**0.023**	0.0297

Consequently, we employ our fusion diagrams to add an additional term to (5.1) 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. We, therefore, choose to employ the fusion diagram depicted in Section 3 & 4. 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 (5.1), giving the following functional form for our modified Cormorant layer.

$$F_i^{s-1} = \left[F_i^s \oplus \left(F_i^{s-1} \otimes_{\text{cg}} F_i^{s-1}\right) \oplus \sum_i \left(\sum_j \left(Y(\mathbf{x}_{ij}) \otimes_{\text{cg}} F_j^{s-1}\right) g_{ij} \oplus F_{ij}^{fusion}\right)\right] \cdot W_{s,\ell}^{\text{vertex}}, \quad (5.2)$$

where F_{ij}^{fusion} is the output of the fusion block where inputs came from atom i and atom j in (3.6) within the coming legs chosen to be ith atom, jth atom, and their connecting edge. In other words, we use the fusion diagram to efficiently combine the atom-level messaging passing and edge-level messaging passing, with significantly shallow layers and less amounts of parameters.

e then compare the performance of a network with the resulting layers to the original Cormorant architecture by comparing them to the two learning tasks described in the Cormorant paper: regressing various predictions on the QM-9 dataset [70] and predicting the energies of the MD-17 dataset [23]. We leave almost all network hyperparameters the same as those in the original experiments. However, we reduce the number of neural network layers to two and set the number of channels to 8 for our QM9 experiments and 16 for MD17. This 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 Cormorant [20], and our QM9 neural network has 121872 parameters compared to 299808 in the original [20]. Code for our modified network can be found at https://github.com/ehthiede/diagram_corm. In table 1 we compare the mean absolute error of the two models. We find that our modified architecture outperforms the original Cormorant architecture on a majority of the learning tasks. Consequently, we believe that selectively incorporating neural network blocks derived from fusion diagrams can improve the expressiveness of rotation-equivariant neural networks.

6 Conclusion

In this work, we have introduced a new method for constructing equivariant blocks for rotation-equivariant layers based on the use of 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 common to physics, we construct new components that can be incorporated into equivariant neural networks that we call "fusion blocks".

Using classic results from invariant theory, we show that neural networks using fusion blocks are capable of approximating any continuous SU(2) $\times$ S_n-equivariant functions. To demonstrate their practical utility, we perturb an existing SO(3) equivariant neural network architecture, the Cormorant

architecture, by incorporating a fusion block in the Cormorant layer. The modified architecture generally achieves better performance for a smaller number of parameters. Indeed, the idea using equivariance and symmetry to prune the neural networks has been applied [71] in the quantum setting. We believe this indicates that neural networks built using fusion diagrams can be useful building blocks in the design of group-equivariant neural networks.

In future work, we intend to implement parity-preserving O(3) equivariant fusion blocks and employ more advanced fusion diagrams. We also hope to use them 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. In the future, employing the formalism of recoupling may highlight hidden symmetries in the network architecture, indicating new ways to save computational effort.

Indeed, these efforts would complement existing approaches that attempt to build many-body effects into neural networks for three-dimensional point clouds: For instance, concurrent work has explored employing successive tensor products over multiple atoms in an atomic neighborhood[72]. 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 or not fusion diagrams serve as a leading principle towards building more expressive graph neural nets with 3D equivariance specific to chemical applications. We leave addressing this question to our forthcoming publication.

Acknowledgements

We specially thank Hy Truong Son for insightful discussion and help in setting up numerical simulation. We thank Chih-chan Tien, Ziyu Ye, Zixin Ding, Yuxin Chen, Zihan Pengmei, and Liang Jiang for insightful discussions. The Flatiron Institute is a division of the Simons Foundation. Computational Resources were provided by the Flatiron Institute Scientific Computing Core and the University of Chicago Research Computing Center. JL 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).

References

[1] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," (2021), arXiv:2102.09844 [cs.LG] .

[2] Z. Qiao, A. S. Christensen, M. Welborn, F. R. Manby, A. Anandkumar, and T. F. M. I. au2, "Unite: Unitary n-body tensor equivariant network with applications to quantum chemistry," (2021), arXiv:2105.14655 [cs.LG] .

[3] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," (2021), arXiv:2101.03164 [physics.comp-ph] .

[4] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, The Journal of Chemical Physics **148**, 241722 (2018).

[5] K. Schütt, O. Unke, and M. Gastegger, in *International Conference on Machine Learning* (PMLR, 2021) pp. 9377–9388.

[6] H. Stärk, O.-E. Ganea, L. Pattanaik, R. Barzilay, and T. Jaakkola, arXiv preprint arXiv:2202.05146 (2022).

[7] M. Shuaibi, A. Kolluru, A. Das, A. Grover, A. Sriram, Z. Ulissi, and C. L. Zitnick, arXiv preprint arXiv:2106.09575 (2021).

[8] J. Behler and M. Parrinello, Physical Review Letters **98**, 146401 (2007).

[9] C. Esteves, arXiv preprint arXiv:2004.05154 (2020).

[10] T. S. Cohen *et al.*, "Equivariant convolutional networks," (2021).

[11] R. Kondor and S. Trivedi, "On the generalization of equivariance and convolution in neural networks to the action of compact groups," (2018), arXiv:1802.03690 [stat.ML] .

[12] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds," (2018), arXiv:1802.08219 [cs.LG] .

[13] R. Kondor, "N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials," (2018).

[14] R. Penrose, in *Quantum Theory and Beyond* (Cambridge University Press, Cambridge, 1971) pp. 151–180.

[15] P. Martin-Dussaud, General Relativity and Gravitation **51**, 110 (2019).

[16] I. Mäkinen, "Introduction to SU(2) recoupling theory and graphical methods for loop quantum gravity," (2019), arXiv:1910.06821 [gr-qc] .

[17] S. Singh and G. Vidal, Physical Review B **86** (2012), 10.1103/physrevb.86.195114.

[18] P. Schmoll, S. Singh, M. Rizzi, and R. Orús, Annals of Physics **419**, 168232 (2020).

[19] P. Schmoll and R. Orús, Phys. Rev. B **102**, 241101 (2020).

[20] B. Anderson, T.-S. Hy, and R. Kondor, "Cormorant: Covariant molecular neural networks," (2019), arXiv:1906.04015 [physics.comp-ph] .

[21] G. Dusson, M. Bachmayr, G. Csanyi, R. Drautz, S. Etter, C. van der Oord, and C. Ortner, "Atomic cluster expansion: Completeness, efficiency and stability," (2019).

[22] A. P. Bartók, R. Kondor, and G. Csányi, Physical Review B **87**, 184115 (2013).

[23] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Science Advances **3** (2017), 10.1126/sciadv.1603015.

[24] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Science Advances **3**, e1701816 (2017).

[25] S. Villar, D. W. Hogg, K. Storey-Fisher, W. Yao, and B. Blum-Smith, "Scalars are universal: Equivariant machine learning, structured like classical physics," (2021).

[26] J. Han, L. Zhang, R. Car, and W. E, "Deep potential: A general representation of a many-body potential energy surface," (2017).

[27] J. Gasteiger, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," (2020).

[28] J. Gasteiger, F. Becker, and S. Günnemann, "Gemnet: Universal directional graph neural networks for molecules," (2021).

[29] P. R. Kaundinya, K. Choudhary, and S. R. Kalidindi, "Prediction of the electron density of states for crystalline compounds with atomistic line graph neural networks (alignn)," (2022).

[30] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," (2016), arXiv:1602.07576 .

[31] T. S. Cohen and M. Welling, in *International Conference on Learning Representations (ICLR)* (2017).

[32] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling, "Spherical CNNs," (2018), arXiv:1801.10130 [cs.LG] .

[33] R. Kondor, Z. Lin, and S. Trivedi, in *Advances in Neural Information Processing Systems (NeurIPS)* (2018).

[34] V. G. Satorras, E. Hoogeboom, and M. Welling, in *International Conference on Machine Learning* (PMLR, 2021) pp. 9323–9332.

[35] A. Bogatskiy, B. Anderson, J. Offermann, M. Roussi, D. Miller, and R. Kondor, in *International Conference on Machine Learning* (PMLR, 2020) pp. 992–1002.

[36] H. Maron, H. Ben-Hamu, N. Shamir, and Y. Lipman, in *International Conference on Learning Representations* (2019).

[37] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman, in *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32 (2019).

[38] T. S. Cohen, M. Geiger, and M. Weiler, in *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

[39] H. Maron, O. Litany, G. Chechik, and E. Fetaya, in *International Conference on Machine Learning* (PMLR, 2020) pp. 6734–6744.

[40] S. R. White, Physical Review Letters **69**, 2863 (1992).

[41] S. Rommer and S. Östlund, Physical Review B **55**, 2164 (1997).

[42] R. Orús, Nature Reviews Physics **1**, 538 (2019).

[43] M. Fannes, B. Nachtergaele, and R. F. Werner, Communications in Mathematical Physics **144**, 443 (1992).

[44] F. Verstraete and J. I. Cirac, "Renormalization algorithms for quantum-many body systems in two and higher dimensions," (2004).

[45] G. Vidal, Physical Review Letters **99**, 220405 (2007).

[46] C. Huang, F. Zhang, M. Newman, X. Ni, D. Ding, J. Cai, X. Gao, T. Wang, F. Wu, G. Zhang, *et al.*, Nature Computational Science **1**, 578 (2021).

[47] F. Pan and P. Zhang, "Simulating the sycamore quantum supremacy circuits," (2021).

[48] E. Stoudenmire and D. J. Schwab, Advances in Neural Information Processing Systems **29** (2016).

[49] S. Efthymiou, J. Hidary, and S. Leichenauer, "Tensor network for machine learning," (2019).

[50] C. Roberts, A. Milsted, M. Ganahl, A. Zalcman, B. Fontaine, Y. Zou, J. Hidary, G. Vidal, and S. Leichenauer, arXiv preprint arXiv:1905.01330 (2019).

[51] B. Swingle, Physical Review D **86**, 065007 (2012).

[52] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill, Journal of High Energy Physics **2015**, 1 (2015).

[53] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, arXiv e-prints , arXiv:1703.06114 (2017), arXiv:1703.06114 [cs.LG] .

[54] N. Segol and Y. Lipman, "On Universal Equivariant Set Networks," (2019), arXiv:1910.02421 [cs.LG] .

[55] E. H. Thiede, T. S. Hy, and R. Kondor, "The general theory of permutation equivarant neural networks and higher order graph variational encoders," (2020), arXiv:2004.03990 [cs.LG] .

[56] L. C. Biedenharn, A. Giovannini, and J. D. Louck, Journal of Mathematical Physics **8**, 691 (1967), https://doi.org/10.1063/1.1705266 .

[57] M.-J. Gao and J.-Q. Chen, J. Phys. A **18**, 189 (1985).

[58] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii, *Quantum Theory of Angular Momentum* (WORLD SCIENTIFIC, 1988).

[59] I. Mäkinen, "Introduction to SU(2) recoupling theory and graphical methods for loop quantum gravity," (2019), arXiv:1910.06821 [gr-qc] .

[60] F. V. Rosen AS, O. C. Huck P, and T. D. e. a. Horton MK, npj Comput Mater (2022).

[61] D. Ghanekar, P.G., S. . Greeley, and J. Adsorbate, nature communications **13**, 5788 (2022).

[62] P. J. Olver, *Classical Invariant Theory*, London Mathematical Society Student Texts (Cambridge University Press, 1999).

[63] C. Procesi, *Lie groups: an approach through invariants and representations*, Universitext (Springer, New York, 2007) oCLC: ocm61309419.

[64] R. Goodman and N. R. Wallach, *Symmetry, Representations, and Invariants* (Springer New York, 2009).

[65] D. Hilbert and B. Sturmfels, *Algebraic invariants*, Cambridge mathematical library (Cambridge University Press, Cambridge [England] ; New York, NY, USA, 1993).

[66] P. Gordan, Journal für die Reine und Angewandte Mathematik **1868**, 323 (1868).

[67] M. Olive, "About Gordan's algorithm for binary forms," (2014), arXiv:1403.2283 [math.RT] .

[68] I. Lindgren and J. Morrison, *Atomic Many-Body Theory* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1986).

[69] M. J. Field, *Dynamics and Symmetry*, ICP Advanced Texts in Mathematics, Vol. 3 (Imperial College Press, 2007).

[70] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, Scientific Data **1** (2014).

[71] X. Wang, J. Liu, T. Liu, Y. Luo, Y. Du, and D. Tao, arXiv preprint arXiv:2208.14057 (2022).

[72] I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, "The design space of E(3)-equivariant atom-centered interatomic potentials," (2022).

Supplementary Material

Contents

1 Connection to Atomic Cluster Expansion

The atomic cluster expansion (ACE) [3] is a non-neural network method to compute the inter-molecular potential energy surface:

$$E : \mathcal{R} \to \mathbb{R}$$

Where $\mathcal{R}$ is the configuration space, a set of particle position vectors. More precisely, the authors consider the set of admissible finite configurations:

$$\mathcal{R} := \bigcup_{J=0}^{\infty} \mathcal{R}_J, \quad \mathcal{R}_J := \{\{\mathbf{r}_1, \cdots, \mathbf{r}_J\}\} \tag{1}$$

which is invariant under isometries (permutation invariance is already implicit in identifying configurations as sets), that is, for an isometry $Q \in O(3)$, for any $(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_J) \in \mathbb{R}^{3J}$,

$$E\left(\{Q\boldsymbol{r}_1, \ldots, Q\boldsymbol{r}_J\}\right) = E\left(\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_J\}\right)$$

and observes a certain locality of interaction, in the sense that the energy is additive for configurations composed of far away subsystems. The potential energy surface is assumed via the *body-order* expansion: for a typical N body expansion: $V_N\left(\boldsymbol{r}_{\iota j_1}, \ldots, \boldsymbol{r}_{\iota j_N}\right)$, the potential functions involving N neighbor atoms from a given center atom.

$$\begin{aligned} E\left(\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_J\}\right) &= \sum_{\iota=1}^{J} E_\iota\left(\{\boldsymbol{r}_j - \boldsymbol{r}_\iota\}_{j\neq\iota}\right) \\ E_c\left(\{\boldsymbol{r}_{cj}\}_{j=1}^{J-1}\right) &= V_0 + \sum_{N=1}^{\mathcal{N}} \sum_{j_1<j_2<\cdots<j_N} V_N\left(\boldsymbol{r}_{cj_1}, \ldots, \boldsymbol{r}_{cj_N}\right) \end{aligned} \tag{2}$$

With $\mathbf{r}_{cj} = \mathbf{r}_j - \mathbf{r}_c$, the relative displacement vector between a neighbor atom to the center atom $\mathbf{r}_c$, and $V_0 \in \mathbb{R}$. We require some further constraints: regularity, locality, rotational and

permutational symmetry. We refer the details of the construction to the paper [3]. At the center of the construction for the rotational invariance is to compute the following projector:

$$\widetilde{\mathbf{D}}^{m_1\cdots m_N}{}_{n_1\cdots n_N} := \int dg D^{(j_1)m_1}{}_{n_1}(g)\cdots D^{(j_N)m_N}{}_{n_N}(g) = \sum_{\iota} \overline{\iota^{m_1\cdots m_N}}\iota_{n_1\cdots n_N} \tag{3}$$

Where $\{\iota_{n_1,\cdots,n_N}\}$ are the corresponding basis elements of $\mathrm{Hom}_{\mathrm{SO}(3)}(H_{c1}\otimes\cdots\otimes H_{cN})$, the space of all mutilinear functions $H_{c1}\times\cdots\times H_{cN}\to\mathbb{C}$ that are invariantly under rotation. Using Eq.(3), the authors construct the spanning set (Eq.(3.6) in [3]):

$$|\mathbf{b}_{n_1\cdots n_N}\rangle := \widetilde{\mathbf{D}}^{m_1\cdots m_N}{}_{n_1\cdots n_N}|Y^{l_1\cdots l_N}{}_{m_1\cdots m_N}\rangle \tag{4}$$

Therefore, such spanning set is constructed by all basis elements of $\iota_{n_1,\cdots,n_N}$. In our work, we only consider a selective few to construct the fusion block. It is worth noting that in Lemma 3 [3], it is also proven that Eq.(3) can be written in a sequence of Clebsch-Gordan coefficients, to which they refer as *generalized Clebsch Gordan coefficients*.

2 Proof of the Universality Theorem

The universality theorem is proved by results from Classical Invariant Theory [8, 16, 18] and Quantum Angular Momentum Recoupling Theory [13, 23, 14]. 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 $C^{j_1j_2k}\psi^{j_1}\psi^{j_2}$ 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 [13]. Spin diagrams used in this paper are graphical interpretation of various recoupling schemes. They are further generalized in Topological Quantum Computation [10] and Quantum Gravity [17]. 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 [7]. Therefore, in our setting with $V = V_j^{\oplus n}$, linear equivariant layers can only be defined as learnable weights combining n inputs $(\psi_1, ..., \psi_n)$ linearly [11]. 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:

$$C^J{}_{j_1 j_2} : V_{j_1} \otimes V_{j_2} \to V_J.$$

The Clebsch-Gordan non-linearity was first used in [12, 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),

$$\left(C^J{}_{j_1 j_2}(\psi_1, \psi_2)\right)^M = (C^J{}_{j_1 j_2})^M{}_{m_1 m_2} \psi_1^{m_1} \psi_2^{m_2},$$

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 $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1}, V_{j_2};\, W)$. Another equivalent notation $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes V_{j_2};\, W)$ and its generalization $\mathrm{Hom}_{\mathrm{SU}(2)}(V_{j_1} \otimes \cdots \otimes V_{j_n};\, W)$ 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

$$V_j \mapsto V_j \oplus V_j \to V_J; \quad \psi \mapsto (\psi, \psi) \mapsto C^J{}_{jj}(\psi, \psi).$$

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.,

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

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

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

$$C^J{}_{j_1 k}\left(C^k{}_{j_2 j_1}(\psi_1, \psi_2), \psi_1\right)$$

   is a multi-homogeneous polynomial of degree $(2, 1)$. One variable polynomial can be decomposed into homogeneous polynomials, as a generalization, multi-homogeneous poly-

nomials are building blocks for multivariate polynomials [18]. 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 SU(2)-equivariant polynomial. This is true due to Gordan Theorem [8, 16, 15]:

Theorem 2.1 (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 defined V can be written as a* linear combination *of iterated CG-products $((...((\psi_1, \psi_2)_{k_1}, \psi_3)_{k_2}, ..., \psi_n)_{k_{n-1}}$. 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 [9, 16], while the matrix/tensor notation of CG-products was adapted by later researchers like Wigner and Racah to study quantum physics and representation theory [13, 23]. We follow the tensor notation from physics and representation theory in this paper.

Gordan theorem was originally formulated to find all *generators* of SL(2, $\mathbb{C}$)-equivariant polynomials, which can be further restricted to 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 [16, 7] 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 SL(2,$\mathbb{C}$) irreps are summarized in [16, 15]. 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 [7] (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.

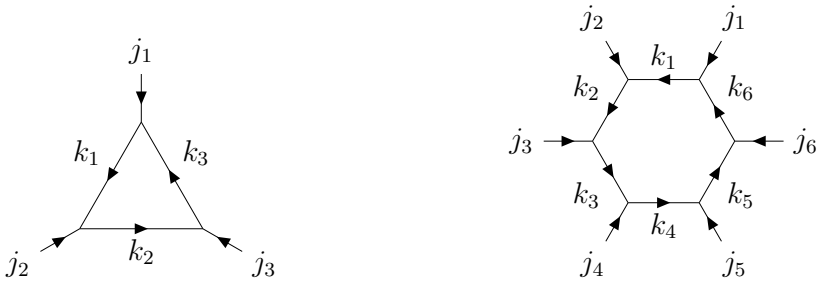
2.2 Spin Diagram and Spin Recoupling

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 [21, 20, 19]. While general tensor networks are obtained by contracting arbitrary tensors which may not be SU(2)-equivariant [1]. 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*

[1] It 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

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 [21, 20, 19] for more examples). Based on the graphical expression, Gordan theorem 2.1 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 $d = (d_1, ..., d_k)$ has $\frac{d!}{d_1!\cdots d_k!}$ possible different forms even written as a iterated CG-product like

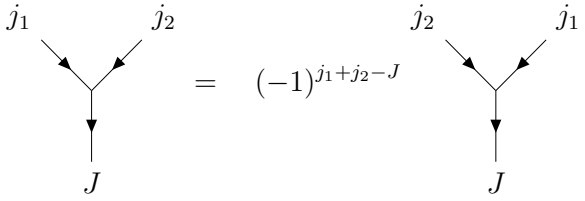
$$C^J{}_{j_2k}\big(C^k{}_{j_1j_1}(\psi_1,\psi_1),\psi_2\big),\quad C^J{}_{j_1k}\big(C^k{}_{j_2j_1}(\psi_2,\psi_1),\psi_1\big),\quad C^J{}_{j_1k}\big(C^k{}_{j_1j_2}(\psi_1,\psi_2),\psi_1\big).$$

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 as two symmetric properties when computing Clebsch-Gordan coefficients [13, 23, 10]:

1. The *R-move*:

$$(C^J{}_{j_1j_2})^M{}_{m_1m_2} = (-1)^{j_1+j_2-J}(C^J{}_{j_2j_1})^M{}_{m_2m_1}$$

   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:

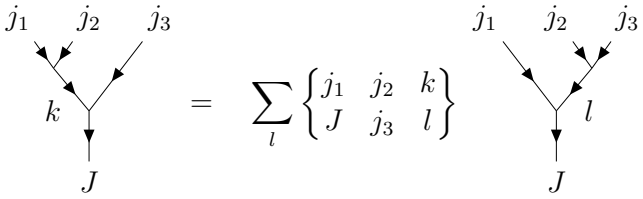
$$(\psi_1,\psi_2,\psi_3) \mapsto \big(C^k{}_{j_1j_2}(\psi_1,\psi_2),\psi_3\big) \mapsto C^J{}_{kj_3}\big(C^k{}_{j_1j_2}(\psi_1,\psi_2),\psi_3\big),$$
$$(\psi_1,\psi_2,\psi_3) \mapsto \big(\psi_1, C^l{}_{j_2j_3}(\psi_2,\psi_3)\big) \mapsto C^J{}_{j_1l}\big(\psi_1, C^l{}_{j_2j_3}(\psi_2,\psi_3)\big),$$

   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 *Winger*

6j-symbols:

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

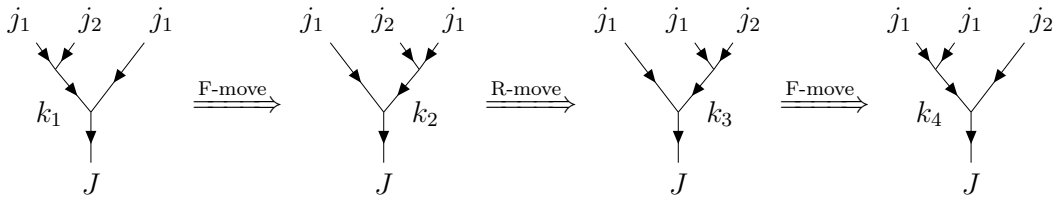
It can be though as a generalized associative rule when coupling more than two spins. Graphically, we draw



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

$$C^{J}{}_{j_1 k}\big(C^{k}{}_{j_1 j_2}(\psi_1, \psi_2), \psi_1\big) = \sum_k (2k+1)(-1)^{j_2+j_3+k+l} \begin{Bmatrix} j_1 & j_2 & k \\ J & j_1 & l \end{Bmatrix} C^{J}{}_{j_2 l}\big(C^{l}{}_{j_1 j_1}(\psi_1, \psi_1), \psi_2\big).$$

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 2.2).

2.3 Equivariant Universality with Fusion Blocks

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

Definition 2.2. Let $V_j, W_{j'}$ be two irreps of SU(2) and let $\psi_1, ..., \psi_n$ be spin-j vectors as inputs. Given a multi-degree $d = (d_1, ..., d_n)$, 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 $C^{d}_{\mathbf{k}}(\psi_1, ..., \psi_n)$ 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 2.2. For further use, we would also employ the notation $(d_1, ..., d_n) \vdash d$: $(d_1, ..., d_n)$ is called a *composition* of d as $d_1 + \cdots + d_n = d$.

As an immediate result from Gordan theorem 2.1 and spin recoupling theory, we have:

Theorem 2.3. *Any SU(2)-equivariant polynomial $f : V_j^{\oplus n} \to W_{j'}$ can be written as a linear combination of standardized Clebsch-Gordan products $C^d_{\mathbf{k}}(\psi_1, ..., \psi_n)$.*

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 2.1, 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 2.4. *Let $f : V_j^{\oplus n} \to W_{j'}$ be an S_{n-1}-invariant on its last $n-1$ variables. Then the function F defined by*

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

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

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

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

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

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

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

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

Now let $F = (f_1, ..., f_n)$ 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

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

This indicates that

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

Applying $(1, 3)$, we have

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

Continuing by induction, we can conclude that

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

To verify that f is S_{n-1}-invariant, we check Eq.(9) again which shows that $f_i(\psi_1, ..., \psi_n) = f(\psi_i, \psi_1, ..., \psi_n)$ 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. □

Remark. In the language from [22]), 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 2.5. *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 $1, ..., n$ with atomic activations $(\psi_1, ..., \psi_n) \in V_j^{\oplus n}$. Updating with fusion blocks, we are able to create SU(2) $\times S_n$ equivariant polynomials $F : V_j^{\oplus n} \to W_j^{\oplus n}$ due to the employment of CG-products and aggregating at each layer. Any such function F can be expanded as Eq.(5) by Lemma 2.4 in which each component $f(\psi_i, \psi_1, ..., \psi_n)$ is recorded with the ith atom being chosen as the center.

Since $f : V_j^{\oplus n} \to W_{j'}$ is both S_{n-1}-invariant on its last $n-1$ variables and $SU(2)$-equivariant, by Theorem 2.3,

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

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

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

where we denote by $\overline{C}_{\mathbf{k}}^d(\psi_1, \psi_2, ..., \psi_n)$ the symmetrization of $C_{\mathbf{k}}^d(\psi_1, \psi_2, ..., \psi_n)$. 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 $d = (d_1, ..., d_n)$, we first process ψ_i by self-coupling (iterated CG-products) without aggregation as in Definition 2.2. 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:

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

   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:

$$\sum_{j\geq 2} C^{J' k^3_{d_3} J} \Big(\sum_{i\geq 2} \big(C^{k^1_{d_1} k^2_{d_2} J'} (C^{d_1}_{\mathbf{k}^1}(\psi_1), C^{d_2}_{\mathbf{k}^2}(\psi_i)) \big), C^{d_3}_{\mathbf{k}^3}(\psi_j) \Big). \tag{16}$$

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

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 $x_1, ..., x_n$ 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

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

where λ denotes a composition $(d_1, ..., d_n)$ of the multi-degree d (see the previous Remark). As $i_1, ..., i_{n-1}$ 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:

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

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

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

It is well-known that monomial symmetric polynomials m_λ can be used to build any permutation-invariant polynomial [7]. 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 [4] we conclude that:

Corollary 2.6 (Global Rotation and Permutation Equivariant Universality)**.** *The Cormorant architecture updated with fusion blocks is universal to approximate any continuous SO(3) $\times S_n$-equivariant map. When summing at the output layer, it recovers $SO(3)$-equivariant and S_n-invariant universality.*

3 Architecture, hyper-parameter, and computational details

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

Our 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

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	
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)**

of the network we changed the number of layers in the Cormorant_CG class to 2 and set the number of channes 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.

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 [6]; 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 [6]. 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 differentiable at 0. Consequently, we instead used the radial functions

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

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

Table 2: QM9 results on the Dimenet split. Benchmarks taken from [5].

Target	Unit	SchNet	MGCN	DeepMoleNet	DimeNet	DimeNet^{++}	Ours
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

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/Å^2	58.5	21.6	**8.0**	23.72
Ethanol	Energy	meV	3.5	2.8	**2.2**	2.6
	Force	meV/Å^2	16.9	10.0	**3.1**	9.4
Malonaldehyde	Energy	meV	5.6	4.5	**3.3**	4.7
	Force	meV/Å^2	28.6	16.6	**5.6**	20.4
Naphthalene	Energy	meV	6.9	5.3	**4.9**	8.1
	Force	meV/Å^2	25.2	9.3	**1.7**	21.4
Salicylic Acid	Energy	meV	8.7	5.8	4.6	**4.2**
	Force	meV/Å^2	36.9	16.2	**3.9**	8.9
Toluene	Energy	meV	5.2	4.4	**4.0**	6.0
	Force	meV/Å^2	24.7	9.4	**2.0**	16.6
Uracil	Energy	meV	6.1	5.0	4.5	**3.7**
	Force	meV/Å^2	24.3	13.1	**3.3**	6.9

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] 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.

[4] Michael J Field. *Dynamics and Symmetry*, volume 3 of *ICP Advanced Texts in Mathematics*. Imperial College Press, September 2007.

[5] 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.

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

[7] Roe Goodman and Nolan R. Wallach. *Symmetry, Representations, and Invariants*. Springer New York, 2009.

[8] 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.

[9] David Hilbert and Bernd Sturmfels. *Algebraic invariants*. Cambridge mathematical library. Cambridge University Press, Cambridge [England] ; New York, NY, USA, 1993.

[10] Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, January 2006.

[11] Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials, 2018.

[12] 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.

[13] Ingvar Lindgren and John Morrison. *Atomic Many-Body Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[14] Pierre Martin-Dussaud. A primer of group theory for Loop Quantum Gravity and spinfoams. *General Relativity and Gravitation*, 51(9):110, September 2019.

[15] Marc Olive. About Gordan's algorithm for binary forms, March 2014.

[16] Peter J. Olver. *Classical Invariant Theory*. London Mathematical Society Student Texts. Cambridge University Press, 1999.

[17] Roger Penrose. Angular momentum: an approach to combinatorial space-time. In *Quantum Theory and Beyond*, pages 151–180. Cambridge University Press, Cambridge, 1971.

[18] Claudio Procesi. *Lie groups: an approach through invariants and representations*. Universitext. Springer, New York, 2007. OCLC: ocm61309419.

[19] Philipp Schmoll and Román Orús. Benchmarking global $su(2)$ symmetry in two-dimensional tensor network algorithms. *Phys. Rev. B*, 102:241101, Dec 2020.

[20] 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.

[21] 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.

[22] Erik Henning Thiede, Truong Son Hy, and Risi Kondor. The general theory of permutation equivarant neural networks and higher order graph variational encoders, 2020.

[23] D A Varshalovich, A N Moskalev, and V K Khersonskii. *Quantum Theory of Angular Momentum*. WORLD SCIENTIFIC, October 1988.