

# Symmetry Group Equivariant Neural Networks for Physical Systems

Alexander J. Heilman

July 22, 2025

# 1 Literature Review & Background

The fundamental assumption in any mathematical model is that some initial set of properties describing the state of a system fully determines its future state. And at a high level, it could be argued that both the fields of physics and machine learning are most fundamentally applications of mathematical modeling.

Physics is essentially the pursuit of rigorous mathematical models that accurately describe physical phenomena, mapping some set of physical observables at one point in time to other sets of physical observables at the same or different times. Often these models of physics have some (hopefully intuitive) rationale behind them from which they are derived, but such models must always describe the data collected empirically.

Machine Learning (ML) describes a set of techniques which also assume there exists a map between some input space and some target space (though, perhaps, in an extremely complex manner). However, supervised ML approaches typically start with some sufficiently large, and initially randomly-chosen, model which is then iteratively tuned or ‘trained’ to better fit the given target data directly. In this way, supervised ML methods are a general tool that allows one to build mathematical models for arbitrary mappings, given sufficient data, from some set of tunable ansatz. Their possible application to physical systems (the same as those modeled by ‘pure’ physics) should then be obvious.

Indeed, many machine learning techniques have proven to be effective and efficient in many applications to the natural sciences [1–7]. Of particular interest here are supervised learning techniques applied to the prediction of material properties from atomistic systems (usually DFT-relaxed or idealized), typically crystalline materials [8–13] or molecules [14–17].

## 1.1 Graph Neural Networks Applied to Atomic Systems

One set of machine learning techniques utilizes a general set of mathematical functions referred to as neural networks. Neural networks are a class of universal function approximators, composed layer-wise by functions  $\mathcal{L}^1 \circ \mathcal{L}^2 \circ \dots \circ \mathcal{L}^n$ , where each layer-to-layer transition map  $\mathcal{L}^i$  is a trainable function from some feature space associated with layer  $L$  to a feature space associated with layer  $L + 1$  [18]. While hard proofs of the suitability of machine learning techniques are elusive due to the wide range of both techniques and target spaces, some proofs have been given for suitably wide and deep networks as universal function approximators [19, 20].

Many modern approaches use a specific type of neural network, referred to as a Graph Neural Network (GNN), which act on data encoded in features associated with some representative graph (i.e. a collection of nodes and connections between them) [21]. In applications to physical systems, we encode atomic arrangements (such as crystals and molecules) as graphs with associated features and then update their associated features, initially derived from physical properties, through such a graph neural network.

Convolution on graphs (or "message passing" on graphs [22]) is a generalization of typical convolutional networks applied to image-related tasks, such as image classification. In typical convolutional networks, feature maps describing each pixel are updated by the features of their neighborhood of pixels (up to some set size) [23]. Through layers of convolution on graph features, features are updated according to their neighboring node's features, with neighbors here defined as nodes sharing edges [8, 24].

In typical convolutional networks however, pixels have fixed distances between them, allowing for discretized 'filters' that expect fixed distances between neighborhoods of pixels. In atomic systems though, distances between atoms are continuous and thus require filters sensitive to continuously-distributed features. Such 'continuous filters', useful for applications to atomic systems, were first introduced in SchNet [14]. The widely-cited Crystal Graph Convolutional Neural Network (CGCNN [8]) and following works [9, 12, 25] then used continuous filters to great effect on a wide variety of material property prediction tasks.

### 1.1.1 Crystal Graphs

Of course to apply graph neural networks to atomistic systems, such as crystals and molecules, we first require some way to encode these systems in corresponding graphs. We may define a crystal graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  as a set of vertices  $v_i \in \mathcal{V}$ , corresponding to each atom  $i$ , and edges  $e_{ij} \in \mathcal{E}$ , where edges are determined by some physical criteria. Physical information is then associated with the objects in these graphs by way of feature vectors. These are vectors with components describing the physical characteristics of their corresponding graph component, and are further 'learned' or updated throughout the layers of a graph neural network.

The nodes' feature vectors encode the atomic information of the sites they describe. Two usual techniques include: explicitly engineered feature vectors (as in [8]); and the learning of encodings for atomic sites based only on their atomic number (as in [12]), beginning with some random initialization. Edge features are often derived exclusively from their corresponding interatomic distance, and expressed as a multidimensional feature vector by way of a Radial Basis Functions (RBF). Common radial basis functions include Gaussian and Bessel expansions, with an example Gaussian RBF given in section 1.1.1.

A commonly applied criterion for the formation of edges between atoms is a combination of a maximum distance cutoff  $r_{max}$  and a maximum number of neighbors for each node  $N_{max}$ . That is, for each atom, edges are constructed between its node and its  $\leq N_{max}$ -th closest neighbors in the crystalline structure within a shell of radius  $r_{max}$ . While other criteria may also be utilized, such as face-sharing in Voronoi Tesselations [26], and scaled maximum distance cutoffs based on nearest neighbor distances, these more advanced criteria are generally more computationally expensive in the data processing stage and often provide negligible differences in performance. Alternatively, most models [8, 9, 12] adopt the more basic criteria of maximum numbers of neighbors within cutoff shells,

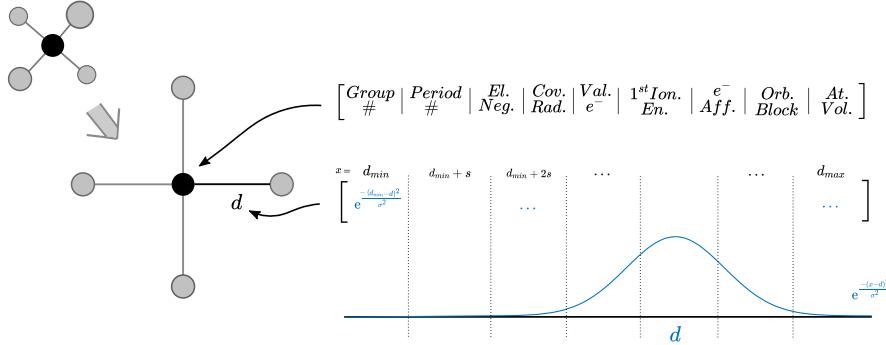


Figure 1: Schematic for a simple CGCNN-like graph encoding of some basic atomic structure. Note that the atomic properties are generally expanded by some one-hot encoding or RBF expansion and then concatenated. The edge feature here then is given, as an example, in terms of a Gaussian expansion of  $d$ , where  $d$  is the relevant interatomic spacing.

and adopt some continuous-filter convolutional function that can learn which connections are more important for the task at hand. Often, this maximum number of neighbors considered is chosen to be 12, however recent tests in coNGN [24] suggest even this may be too restrictive, and show that a maximum set as large as 24 may improve performance.

### 1.1.2 Crystal Graph Convolution

Crystal graphs are usually constructed solely for use in graph convolutional neural networks. Perhaps the most general framework in which we may define graph convolution is the message passing framework defined by Gilmore [22]. A message passing network updates nodes based on ‘messages’ generated by the features of, and passed through, neighboring nodes (that is, nodes sharing an edge). Gilmore defines the message passing framework in terms of three functions:  $M$ , the message forming function;  $U$ , the node update function; and  $R$ , the readout function. These three functions act on graph representations as follows:

$$\begin{aligned} m_i^{t+1} &= \sum_{n_j \in \mathcal{N}(i)} M_t(n_i^t, e_{ij}, n_j^t) \\ n_i^{t+1} &= U_t(n_i^t, m_i^{t+1}) \\ \hat{y} &= R(\{n_v^T\}) \end{aligned}$$

Where for each pair of connected nodes (or edge), a message is formed from each node’s representation  $n_i^t, n_j^t$ , of layer  $t$ , and the connecting edge’s feature

$e_{ij}$ . These node representations of layer  $t$  are updated to their  $t + 1$ -th layer representation by update function  $U_t$  of layer  $t$ , which takes as input the set of messages for each node, and the nodes current representation. At the final layer  $T$ , the readout function  $R$  then predicts the target value  $\hat{y}$  from the updated node representations.

Message passing networks acting on crystal graphs have several advantages: message passing networks enjoy invariance under permutation of node indices; and, further, if the encoded features are coordinate system invariant, the output of the network is itself coordinate system invariant.

However, the graph construction above lacks the inclusion higher-order geometrical information within the crystals. That is, distinct local geometrical environments of atoms (coordination environments/motifs) may be mapped degenerately to the same crystal graph [24, 27]. One could include atomic position in the node features or a vector direction in edge features, however, this generally requires a unique treatment of such coordinate-system dependent information through convolution. As such, often only coordinate system invariant features are included in crystal graph representations, such as distance and atomic properties. Modern approaches that do incorporate coordinate-system dependent features are often *equivariant* by construction, and such equivariant networks are introduced in the following section.

## 1.2 $O(3)$ -Equivariant Networks

When a function is claimed to be equivariant, it always is with respect to some group that acts on its input and output space. For physical space, the most general symmetry group is  $O(3)$ , the orthogonal group of  $\mathbb{R}^3$ , containing all rotations and reflections about some central point.  $O(3)$ -Equivariant networks provide a framework in which direction quantities are encoded and, further, respected through convolution; in this way, outputs themselves may be treated as directional quantities. In essence, equivariant functions commute with their domain and codomain's group representations. More explicitly, for a function  $f : X \rightarrow Y$ , where  $X$  and  $Y$  are vector spaces with some set of group representations  $\mathcal{D}^X(G)$  and  $\mathcal{D}^Y(G)$  acting on these spaces respectively,  $f$  is equivariant with respect to  $G$  if it satisfies:

$$\mathcal{D}^Y(g)f(x) = f(\mathcal{D}^X(g)x)$$

Compositions of equivariant functions are themselves equivariant functions [28]. As such, we may form an equivariant network by composing it layer-wise from a set of equivariant functions. Common choices for such building blocks include:  $O(3)$ -feature convolutions,  $\ell$ -wise self-interactions and non-linearities, and pooling across nodes. The most important of these is  $O(3)$ -equivariant convolution.

Through convolution, convolutional filters and node features interact through tensor products, since tensor products of two representation spaces are equivariant under transformation of the two subspaces, i.e.:

$$\mathcal{D}^V \otimes \mathcal{D}^W = \mathcal{D}^{V \otimes W}.$$

By way of Clebsch-Gordan coefficients  $c_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3}$ , these tensor products of  $SO(3)$  irreducible representations may be related to a third set of irreducible representations as [29]:

$$(u \otimes v)_{\ell_o}^{m_o} = c_{\ell_1 m_1 \ell_2 m_2}^{\ell_o m_o} u_{\ell_1}^{m_1} v_{\ell_2}^{m_2}$$

where  $u$  and  $v$  are harmonic vectors of order  $\ell_1$  and  $\ell_2$ , respectively. Since tensor products of representations are naturally equivariant,  $SO(3)$ -equivariant convolution is often defined as the scaled tensor product of the two representation spaces (i.e. that of the input feature space, and the filter space). Specifically, layer to layer convolutional maps  $\mathcal{L}$  may be defined component-wise as [30]:

$$\mathcal{L}_{acm_o}^{\ell_o}(\vec{r}_a, V_{acm_i}^{\ell_i}) = \sum_{m_f, m_i} c_{\ell_i m_i \ell_f m_f}^{\ell_o m_o} \sum_b F_{cm_f}^{\ell_f \ell_i}(r_{ab}) V_{bcm_i}^{\ell_i}$$

where the filter function  $F_{cm_f}^{\ell_f \ell_i}(r_{ab})$  depends only on the distance between point  $a$  and  $b$  (as opposed to directional dependence, to maintain equivariance). Instances of such functions then generally have independent, trainable parameters for different rotational orders  $\ell_f, \ell_i$ , azimuthal orders  $m$ , and channels  $c$ . It should be noted that this requires the node and edge features of the graphs to also have these additional indexes in such equivariant networks, as well as the outputs of these networks.

## 2 Current/Previous Work

My previous work pertains to both invariant and equivariant graph neural networks in their application to materials science. Particularly, these works focus on improving the expressiveness of invariant graph neural networks by generalizing descriptors to hypergraphs; and utilizing  $O(3)$ -equivariant networks to predict tensorial material properties. Below, the techniques of both of these previous works are briefly introduced to help further motivate the proposed problem, the statement of which follows this section.

### 2.1 Crystal Hypergraph Networks

The primary problem with the representations of material systems in invariant graph neural networks is that they often lack 'geometrical resolution' beyond interatomic distance. For example, in the proto-typical techniques of CGCNN, distinct coordination environments of the same interatomic-distance and coordination number will be mapped onto equivalent graphs, as displayed in fig. 2.

Several approaches have been taken to increase the geometric resolution of invariant graph neural networks: namely, the inclusion of bond-angles as additional triplet features, dihedral angles as quartic features, etc.; and, alternatively, the explicit inclusion of higher-order (than strictly pair-wise) tailored neighborhood features describing common motifs in crystal systems, or more generally, common coordination environments.

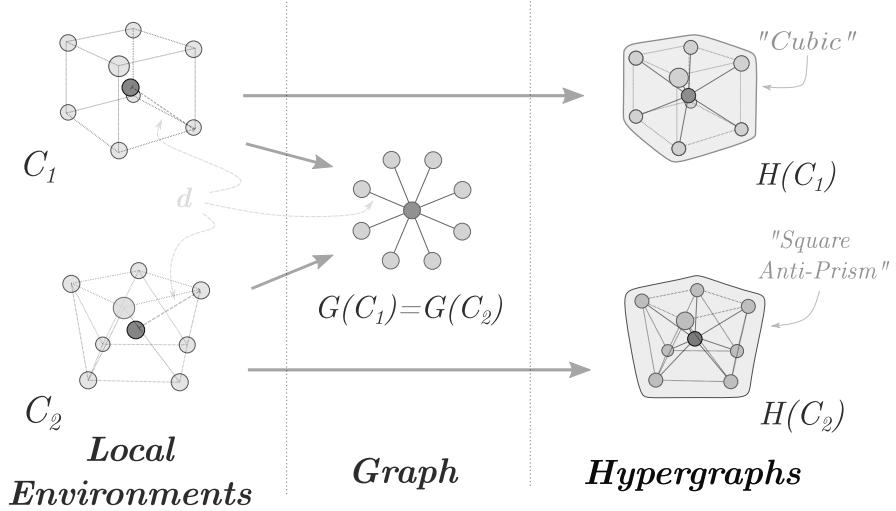


Figure 2: An example of two distinct geometries that are mapped to the same distance-based crystal graph. With inclusion of a first-shell feature vector encoding local geometry however, these structures are mapped to two distinct crystal hypergraphs. Note these are two possible coordination environments in oxides, determined statistically in [31].

My approach to increasing geometric resolution, while maintaining invariance, was to extend the graphical descriptions of material systems to hypergraph descriptions, in which edges (now hyperedges) are then allowed to contain more than two atoms. This allows for all invariant features to be naturally associated with arbitrary order structures in materials, simultaneously on equal footing. For example, pairs, triplets, and coordination environments may all be described in the same crystal hypergraph as hyperedges of differing orders.

### 2.1.1 Crystal Hypergraphs

Hypergraphs are a very general framework describing relations between an abstract set  $\mathcal{V}$  of nodes or vertices, defined by a set of hyperedges  $\mathcal{H}$  containing arbitrary subsets of  $\mathcal{V}$ . Consequently, the set of all hypergraphs on a set of nodes is more general than, and contains all possible, topological sets and simplicial complexes on  $\mathcal{V}$  [32]. The method proposed here reduces the intrinsic limitations of invariant crystal graph features by allowing the explicit incorporation of higher-order geometrical information in the form of hyperedges, which can be used to directly represent these higher-order structures.

A crystal hypergraph  $\mathcal{H} = \{\mathcal{V}, \mathcal{H}\}$  is a collection of nodes  $v_i \in \mathcal{V}$  and hyperedges  $h_j \in \mathcal{H}$  (containing an arbitrary number of nodes, see Supplementary Note S5), where the hyperedges are most generally heterogeneous. That is, we generally wish to describe different types of hyperedges (e.g. bonds, triplets,

and motifs) in the same crystal hypergraph. These objects then have associated feature vectors encoding relevant physical information, which we also refer to as  $v_i$  and  $h_j$ , since the indices specify all the relevant information for their association to particular nodes and hyperedges, respectively.

For the purpose of modeling material systems, we need to identify what different order structures are most important in their representation. Of course, atomic and bond level information is particularly important. However, higher-order structures may also be of interest, such as triplets of atoms and local environments of atoms, which we refer to as motifs in crystals.

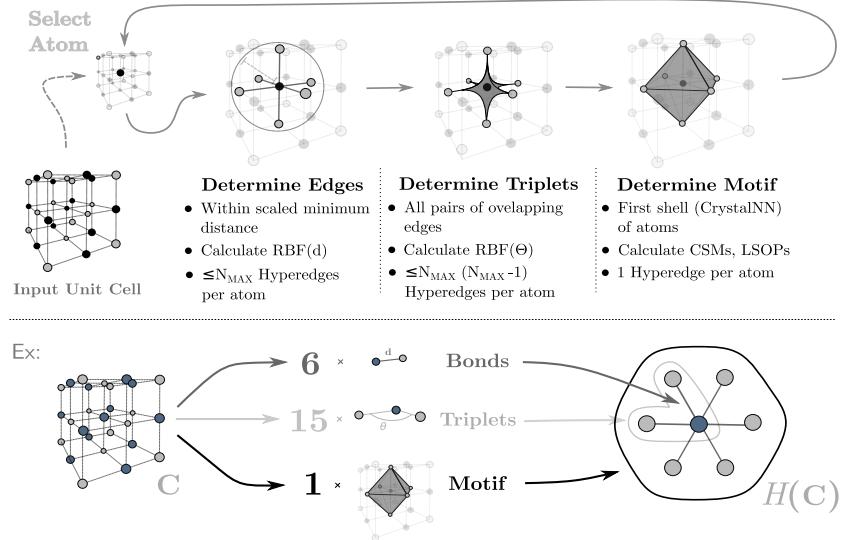


Figure 3: Typical construction loop for a crystal hypergraph. First, pair-wise bonds/edges are determined, then triplets are derived from overlapping pairs of bonds, and finally motifs are determined as first-shells of neighbors by some (generally more restrictive) criteria. Features for each and upper bounds on numbers of hyperedges for each type are also listed. Note that RBF abbreviates radial basis function (such as Gaussian, Bessel, etc.); CSM, continuous symmetry measure; and, LSOP, local structure order parameter.

Each of the aforementioned structures also has a natural set of distinct, coordinate-system invariant features that may be associated with them. At the triplet level (where two bonds share some common node), there is always a corresponding angle. While at the motif level, order parameters [33, 34] or continuous symmetry measures [35, 36] may be used to describe 3-dimensional coordination environments quantitatively. These different order structures may all be represented in a single crystal hypergraph, and is accounted for by treating the data as a heterogeneous graph, with different hyperedge types.

Of course, these different structures require different sets of descriptors, and thus inequivalent feature vectors, suggesting these crystal hypergraphs are naturally heterogeneous in their hyperedges. That is, not all hyperedges in crystal hypergraphs are of the same type. This simply means we need different convolutional filters for each type of hyperedge.

### 2.1.2 Crystal Hypergraph Convolution

To accomodate these extended representations, we must also consider a new convolutional structure, analogous to *Gilmore, et al* [22] but applying to hypergraph structures. That is, we now have:

$$\begin{aligned} m_v^{t+1} &= \sum_{h_j \in \mathcal{N}(v)} M_t(n_v^t, h_j^t, \{n_w^t | n_w \in h_j\}), \\ n_v^{t+1} &= U_t(n_v^t, m_v^{t+1}), \\ \hat{y} &= R(\{n_v^T\}), \end{aligned}$$

so that each node is still updated according to some layer-wise update function  $U_t$ , aggregating messages  $m^{t+1}$  formed from origin node features, hyperedge features  $h_j$ , and hyperedge neighborhood features  $n_w \in h_j$ . This update occurs node-wise and then after  $T$  layers, some readout function  $R$  is used to output the corresponding predicted value  $\hat{y}$ , which utilizes the set of learned node features.

The biggest difference here is that we now need a message forming function  $M_t$  that accounts for a set of node features  $\{n_w^t | n_w \in h_j\}$  which may vary in size between different hyperedges (even of the same type). This stands in opposition to the case of regular edges, where we are assured a fixed size of two nodes per edge.

One approach would be to fix the dimensionality of each type of hyperedge, or have a different convolutional operator for each different size hyperedge (as is effectively the approach taken with line graph networks [37]). Here, however, we wish to maintain generality in edge size so we need not fix hyperedge sizes for hyperedge type, since structures of different sizes may be described by similar metrics. For example, we may wish that motifs resembling polyhedra with different numbers of vertices are described by common sets of features. We considered three strategies that allow us to apply our convolutional operator to hyperedges of arbitrary size, with these approaches detailed schematically in fig. 4.

## 2.2 Equivariant Tensor Predictions

Spherical harmonics, the basis vectors of  $O(3)$ -invariant subspaces, naturally correspond to sets of basis tensors as well [38]. This correspondence allows for the direct prediction of tensorial properties from the output of  $O(3)$  networks in a way that respects transformations on the input coordinate system, acting essentially as similar transformations on the output tensors. In this way,

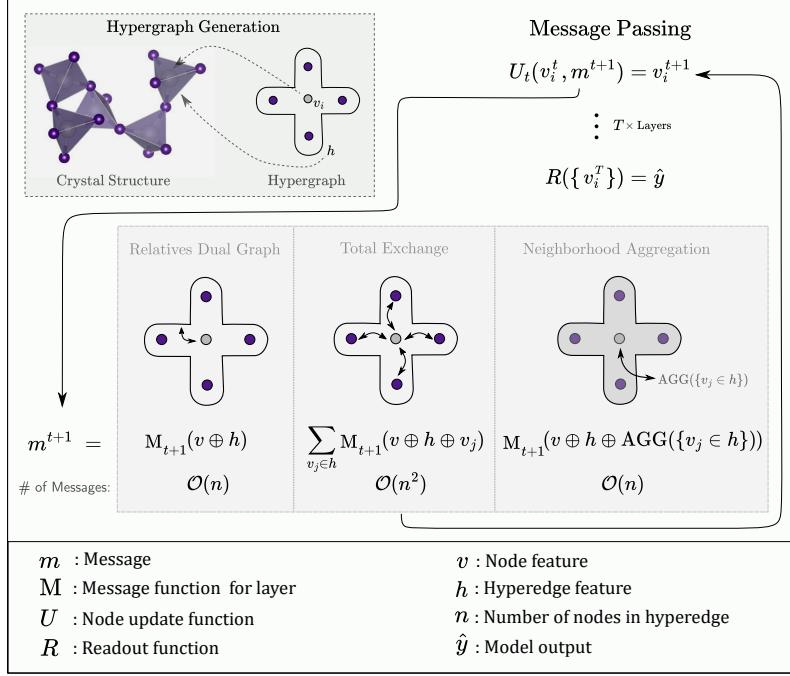


Figure 4: Overview of three possible message functions  $M$  for nodes that generalize the message function in [22] to hyperedges  $h$  with more (or less) than two nodes. AGG specifies some aggregation function acting on the set of node features in the hyperedge, such as max, min, or average.

$SO(3)$  equivariant networks are naturally well suited for the prediction of tensorial properties. Arbitrary tensors may generally be decomposed into a set of  $SO(3)$  invariant subspaces, which can each be associated with a set of spherical harmonic tensors themselves. Since the outputs of  $SO(3)$  networks naturally transform like spherical harmonic tensor coefficients indexed by  $\ell$  and  $m$ , we may then essentially read them off as such and convert these into Cartesian tensor components. In this work [39], we predicted three common material properties described by tensors, namely: the dielectric tensor for anisotropic materials; the piezoelectric tensor, relating strain to internal electric fields; and the elasticity tensor, relating strain to stress.

The key to such applications is the decomposition of tensor spaces into  $\ell$  and  $m$  indexed subspaces (that is, into irreducible  $SO(3)$  tensor subspaces). Since  $SO(3) = (SL(3) \cap O(3)) \subset GL(3)$ , to construct invariants of  $SO$ , we generally form invariant subspaces under  $GL$  by way of Young symmetrizers first, and then construct invariants under  $O$  and/or  $SL$  of these  $GL$  invariant subspaces. This often results in several different harmonic subspaces of the same order  $\ell$  in the harmonic decomposition of a tensor. Also, note that the  $GL$  decompositions for tensors of order greater than two are not, in general, unique and thus  $SO(3)$

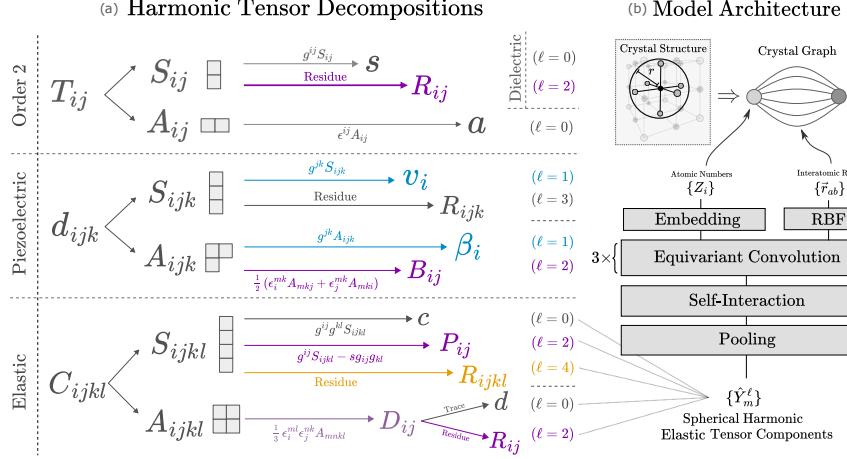


Figure 5: Overview of  $SO(3)$  decomposition of order two tensors, piezoelectric (order 3) tensors, and elasticity (order 4) tensors. Grey corresponds to scalar subspaces of the related tensor space, blue to order 1 subspaces, purple to order 2, and yellow to order 4. A schematic for the general model architecture used in this work is also provided, where the output coefficients correspond directly to harmonic tensor components resulting from these decompositions.

decompositions of tensors are also often not unique. While not the focus of the present work, it should be noted that these three methods correspond to the decomposition of a tensor with respect to the general linear group (by way of Young symmetrizers [40]), the orthogonal group (contractions with  $g_{ij}$ ), and the special linear group (contractions with  $\epsilon_{ijk}$ ), respectively [41, 42]. These techniques allow one to predict arbitrary tensors of a set rank. However, since these sets form complete basis for arbitrary tensors, symmetries of the material are often not captured through convolution, and thus the model predicts non-zero tensor components that are known to be zero a priori from symmetry constraints.

### 3 Proposed Project

$O(3)$ -Equivariant networks have proven effective in applications within the physical sciences, suggesting that physical symmetries are a useful inductive bias in the design of machine learning techniques. While  $O(3)$  networks, in principle, can fully incorporate all possible symmetries of a system, in some applications specific symmetry subgroups may be more relevant. One case of this is applications of machine learning to condensed matter systems, where atoms are arranged in periodic structures where each atomic position has some point symmetry group.

In  $O(3)$  Equivariant networks, initial edge features  $\vec{e}_{ij}^{L=0}$  are often spherical harmonic functions  $Y_m^\ell(\vec{r}_{ab})$  evaluated for interatomic radii  $\vec{r}_{ab}$  of atom pairs  $(a, b)$ . Point group features may be similarly derived from interatomic radii by way of projection operators  $\hat{P}_{ij}^\mu$  which project arbitrary vector spaces onto subspaces  $V_\mu$  transforming as irreducible representations  $\mu$  of point group  $G$ , and defined as [43]:

$$\hat{P}_\alpha^{kk} = \frac{d_\alpha}{N} \sum_g [\Gamma_{kk}^\alpha(g)]^* O(g)$$

where  $d_\alpha$  is the dimension of irrep  $\Gamma^\alpha$ ,  $N$  is the order of the group, and  $O(g)$  is the operator representation acting on the feature space of relevance. While  $O(3)$  networks rely on Clebsch-Gordan expansions of tensor products of neighboring sites and filters, point group networks instead require generalized CG coefficients or ‘coupling coefficients’  $U_{\alpha i \beta j}^{\gamma n}$  for the direct sum decompositions of point group-irreducible features:

$$\psi_n^\gamma = \bigoplus_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} (u_i^\alpha \otimes v_j^\beta)$$

With these coefficients, a similar convolutional structure may be adopted, as in  $O(3)$  networks, though now instead of harmonic indices  $\ell, m$  for features, we have point group irreducible representation indices  $\gamma, n$ :

$$(v_{nc}^{L+1})_n^\gamma = (v_{nc}^L)_n^\gamma + \sum_{b \in \mathcal{N}(n)} \sum_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} (F_c^L(r_{nb}))_j^\beta (v_{bc}^L)_i^\alpha$$

The implementation of these models requires the availability of the analogous Clebsch-Gordan coefficients,  $U_{\alpha i \beta j}^{\gamma n}$ , which we refer to more generally as *coupling coefficients*, for all point groups of interest. These may be defined, from particular forms of representations  $\Gamma^\alpha, \Gamma^\beta, \Gamma^\gamma$  for all point groups of interest from Dirl’s formula [44]:

$$(U_{\alpha i \beta j}^{\gamma n})^m = \sqrt{\frac{d_\gamma}{N_G}} \left( \sum_{g \in G} \Gamma_{qq}^\alpha(g) \Gamma_{ss}^\beta(g) \Gamma_{aa}^{\gamma\dagger}(g) \right)^{-\frac{1}{2}} \cdot \sum_{g \in G} \Gamma_{iq}^\alpha(g) \Gamma_{js}^\beta(g) \Gamma_{na}^{\gamma\dagger}(g)$$

It should be noted here that the definition of these coefficients thus requires explicit (fixed) forms for each irrep of each point group of interest.

We propose to incorporate the above techniques into a novel framework for point-group-equivariant graph networks, suited for applications where systems display particular symmetries reduced from the full rotation symmetry of  $O(3)$ . One particularly relevant application is the prediction of condensed matter Hamiltonian elements. In solid-state systems based upon an underlying periodic lattice, each atomic position displays a certain point group symmetry. Thus the Hamiltonian of the system must share a simultaneous set of eigenvalues with the symmetry operations since in this case they necessarily commute. Essentially then, the Hamiltonians solutions must be indexed by the same set as the irreducible subspaces  $V_\mu$ , each of dimension  $d_\mu$ . The basis set of one-particle wavefunctions then must have a separable form:

$$\psi_{nm}^\mu(\vec{r}) = R_{\mu nm}(r) \Omega_n^\mu(\theta, \phi)$$

where  $1 \leq n \leq d_\mu$ . The radial function  $R_{\mu nm}(r)$  then depends on the specific form of the Hamiltonian and may have a very complicated structure. The prediction of this functional dependence is thus well suited for the machine learning applications discussed here. Such models provide a method by which we may predict Hamiltonian elements naturally associated with these point-group indexes, allowing for complete descriptions of interactions between neighboring sites irreducible subspaces without truncations of rotational order (such as  $\ell_{max}$  in  $O(3)$  networks).

Furthermore, these models allow for the prediction of properties that respect the known symmetries of physical systems, by reading out features that transform as the trivial representation, allowing for more intuitive predictions of scalar (totally invariant) quantities from input directional quantities, as well as tensorial properties with desired physical symmetries.

A natural starting point for such applications is to material systems with high symmetry, the largest crystalline point group being  $O_h$  with 48 operations (including rotations, inversion, and mirror planes). The Materials Project database [45] contains approximately 15,000 crystalline materials with such  $O_h$  symmetry, constituting a suitably large training set for some basic machine learning tasks.

However, while scalar targets, such as band gap, formation energy, and metallicity are generally available from the Materials Project along with the corresponding crystal structures, one large difficulty with this project will be the availability of data for Hamiltonian matrix elements in a properly symmetrized basis (which is also required to be localized in real space). This may possibly be approached by generating such data using a symmetry-adapted Wannier basis with DFT [46].

## References

- [1] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, *InfoMat* **1**, 338–358 (2019).
- [2] S. Chibani and F.-X. Coudert, *APL Mater.* **8** (2020).
- [3] Y. Liu, C. Niu, Z. Wang, Y. Gan, Y. Zhu, S. Sun, and T. Shen, *J. Mater. Sci. Tech.* **57**, 113–122 (2020).
- [4] C. H. Chan, M. Sun, and B. Huang, *EcoMat* **4**, e12194 (2022).
- [5] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, et al., *Commun. Mater.* **3**, 93 (2022).
- [6] S. Kadulkar, Z. M. Sherman, V. Ganesan, and T. M. Truskett, *Annu. Rev. Chem. Biomol. Eng.* **13**, 235–254 (2022).
- [7] M. H. Mobarak, M. A. Mimona, M. A. Islam, N. Hossain, F. T. Zohura, I. Imtiaz, and M. I. H. Rimon, *Appl. Surf. Sci. Adv.* **18**, 100523 (2023).
- [8] T. Xie and J. C. Grossman, *PRL* **120**, 145301 (2018).

- [9] K. Choudhary and B. DeCost, npj Comput. Mater. **7**, 1–8 (2021).
- [10] H. R. Banjade, S. Hauri, S. Zhang, F. Ricci, W. Gong, G. Hautier, S. Vucetic, and Q. Yan, Sci. Adv. **7**, eabf1754 (2021).
- [11] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, arXiv:1802.08219 (2018).
- [12] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, Chem. Mater. **31**, 3564–3572 (2019).
- [13] C. Chen and S. P. Ong, Nat. Comput. Sci. **2**, 718–728 (2022).
- [14] K. T. Schütt, H. E. Saucedo, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).
- [15] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, New J. Phys. **15**, 095003 (2013).
- [16] G. A. Pinheiro, J. Mucelini, M. D. Soares, R. C. Prati, J. L. Da Silva, and M. G. Quiles, J. Phys. Chem. A **124**, 9854–9866 (2020).
- [17] C. R. Collins, G. J. Gordon, O. A. Von Lilienfeld, and D. J. Yaron, J. Chem. Phys. **148**, 24 (2018).
- [18] M. Islam, G. Chen, and S. Jin, American Journal of Neural Networks and Applications **5**, 7–11 (2019).
- [19] K. Hornik, M. Stinchcombe, and H. White, Neural networks **2**, 359–366 (1989).
- [20] B. C. Csáji et al., Faculty of Sciences, Etvs Lornd University, Hungary **24**, 7 (2001).
- [21] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, Distill **6**, e33 (2021).
- [22] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, 34th Int. Conf. Mach. Learn. **70**, 1263–1272 (2017).
- [23] K. O’Shea and R. Nash, *An introduction to convolutional neural networks*, 2015.
- [24] R. Ruff, P. Reiser, J. Stühmer, and P. Friederich, Digit. Discov. **3**, 594–601 (2024).
- [25] C. W. Park and C. Wolverton, Phys. Rev. Mater. **4**, 063801 (2020).
- [26] Q. Du, V. Faber, and M. Gunzburger, SIAM review **41**, 637–676 (1999).
- [27] A. J. Heilman, W. Gong, and Q. Yan, *Crystal hypergraph convolutional networks*, 2024.
- [28] T. Cohen and M. Welling, in Proceedings of the 33rd international conference on machine learning, Vol. 48, edited by M. F. Balcan and K. Q. Weinberger, Proceedings of Machine Learning Research (June 2016), pp. 2990–2999.

- [29] M. Geiger and T. Smidt, arXiv:2207.09453 (2022).
- [30] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, (2018).
- [31] D. Waroquiers, X. Gonze, G.-M. Rignanese, C. Welker-Nieuwoudt, F. Rosowski, M. Gobel, S. Schenk, P. Degelmann, R. André, R. Glaum, et al., Chem. Mater. **29**, 8346–8360 (2017).
- [32] R. Mulas, D. Horak, and J. Jost, *Graphs, simplicial complexes and hypergraphs: spectral theory and topology*, edited by F. Battiston and G. Petri (Springer International Publishing, Cham, 2022), pp. 1–58.
- [33] N. E. R. Zimmermann, M. K. Horton, A. Jain, and M. Haranczyk, Front. Mater. **4**, 34 (2017).
- [34] N. E. Zimmermann and A. Jain, RSC Adv. **10**, 6063–6081 (2020).
- [35] M. Pinsky and D. Avnir, Inorg. Chem. **37**, 5575–5582 (1998).
- [36] D. Waroquiers, J. George, M. Horton, S. Schenk, K. A. Persson, G.-M. Rignanese, X. Gonze, and G. Hautier, Acta Crystallogr. B **76**, 683–695 (2020).
- [37] Z. Chen, X. Li, and J. Bruna, arXiv:1705.08415 (2017).
- [38] F. G. Ledesma and M. Mewes, (2020).
- [39] A. Heilman, C. Schlesinger, and Q. Yan, *Equivariant graph neural networks for prediction of tensor material properties of crystals*, 2024.
- [40] “Irreducible tensor representations and young tableau”, in *Lie groups and lie algebras for physicists* () Chap. Chapter 5, pp. 125–156.
- [41] Y. Itin, Mathematics and Mechanics of Solids **25**, 1873–1895 (2020).
- [42] Y. Itin and S. Reches, Mathematics and Mechanics of Solids **27**, 222–249 (2022).
- [43] A. F. Izmaylov, The Journal of Physical Chemistry A **123**, 3429–3433 (2019).
- [44] R Dirl, Journal of Mathematical Physics **20**, 659–663 (1979).
- [45] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, APL Mater. **1**, 011002 (2013).
- [46] R. Sakuma, Physical Review B—Condensed Matter and Materials Physics **87**, 235109 (2013).