Equivariant Networks for Crystal Structures

Anonymous Author(s)
Affiliation
Address
email

Abstract

Supervised learning with deep models has tremendous potential for applications in materials science. Recently, graph neural networks have been used in this context, drawing direct inspiration from models for molecules. However, materials are typically much more structured than molecules, which is a feature that these models do not leverage. In this work, we introduce a class of models that are equivariant with respect to crystalline symmetry groups. We do this by defining a generalization of the message passing operations that can be used with more general permutation groups, or that can alternatively be seen as defining an expressive convolution operation on the crystal graph. Empirically, these models achieve competitive results with state-of-the-art on the Materials Project dataset.

1 Introduction

Deep learning has seen remarkable applications in computational chemistry, both on the side of molecular property prediction and molecule generation. For small molecules, these methods are close to a level of precision that would make them suitable for practical applications [48]. However, less attention has been put to the problem of designing neural network architectures for the larger assemblies of atoms that constitute materials. This is a crucial problem, as deep learning could advantageously complement computationally demanding ab initio simulation methods like DFT [39, 28]. Random forests and kernel methods have already yielded promising results for concrete applications in materials science, thanks to a large amount of materials data now available. Examples include determination of stable structures [76], photovoltaic cells design [72, 35] and discovery of magnetic and superconducting materials [42, 62, 17].

In this paper, we address the problem of designing equivariant layers for supervised learning on materials. We focus on crystals, materials characterized by the ordered arrangement of their atoms in lattices. Crystalline materials are largely present around us and essential in technological applications, as they include a large number of metals, ceramics, and salts, among others [3]. They can be described as the regular repetition of a set of atoms, called the *unit cell*, in all directions of space. These patterns are analogous to wallpaper patterns, for which the repeated structure is an image. Crystals are characterized by a high degree of symmetry which is fundamental in understanding their physical properties. Consequently, we suggest that equivariant deep learning provides an appropriate framework to design function approximators for crystals.

Graph Neural Networks (GNNs) have recently been proposed for supervised prediction tasks on crystals [56, 74, 12]. We hypothesize here that these models may fall short in exploiting crystal symmetry and that structural information is lost when mapping a crystal structure to a graph. In particular, GNNs are equivariant to the permutation of atoms given as input to the model. This condition is overly restrictive and amounts to forgetting about the *ordered* nature of a crystal. We propose to use models equivariant to a product of groups $G_\Lambda \times S_C$, where G_Λ acts at the level of the *Bravais lattice*, the underlying periodic grid of the crystal, and S_C on the unit cells. We show that our proposed equivariant architecture is more expressive than GNNs on this data structure.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

By using the crystal structure, our approach amounts to defining a group-equivariant convolution kernel on the crystal in a way that is completely analogous to convolutional neural networks (CNNs). This convolution is defined on a graph associated with a crystal structure. Our contributions are the following: 1) We derive different equivariant models based on the group-theoretical properties of crystals. 2) We show some of the limitations of GNNs for crystal data and propose an alternative data structure to be used with our architecture. 3) We perform a rigorous analysis, cleaning, and processing of the Materials Project database [33] and share the resulting processed dataset to serve as a benchmark for materials applications. 4) We perform experimental tests of our models on the Materials Project database and report results comparable to or better than baselines.

2 Related works

Equivariant neural networks It is well known that neural networks have to incorporate inductive biases to be useful in practice [70, 7]. Using models that are invariant or equivariant to the symmetry of the data has proven to be a particularly important inductive bias to promote generalization [8]. The first notable application of this idea was the CNN, for which each convolution layer is equivariant to translation [44]. An alternative parameter-sharing view also appears in early works [59]; while more recent equivariant networks have used both convolutional [13, 41, 15, 20], and parameter-sharing view [53, 25]. A few notable symmetries considered in recent years are rotation in image and volumetric data [13, 71, 68], permutation symmetry in sets [77, 52] and graphs [40, 46], as well as Euclidean [63, 10], and rotational symmetry [14, 2, 22, 57, 24].

Deep learning for materials The increasing availability of large materials datasets from high-throughput calculations [19, 33, 37, 49, 31, 11], makes deep learning more and more relevant for materials science. Following the successes of GNNs on molecular data, similar models have been proposed for materials as alternatives to methods based on feature engineering. Note that in what follows, we refer to GNNs in a general sense that includes message passing neural networks (MPNNs) [27].Many variants of GNNs exist [36, 65, 75, 46], the underlying idea being for each node to aggregate features of neighboring nodes in a permutation invariant, or in the case of [40] in an equivariant way. The CGCNN [74] and MEGNet models [12] rely on mapping crystal structures to graphs and applying GNNs to obtain a prediction. For the SchNet model, the correspondence with graphs is less explicit [56], but still present. Other approaches combine permutation equivariance with E(3)-equivariance to design models for molecules and materials [5, 55].

3 Background on crystal symmetry

We first start by introducing some principles of crystallography that will be used to derive our main results. The exposition presented in the present work is inspired by [61]. A comprehensive treatment can also be found in [21], whereas basics of group theory are covered in [54] for example.

Lattices A crystal can be described as the periodic and infinite repetition of a pattern in all directions of space. Crystals are conveniently described using lattices as their underlying structures.

An n-dimensional *lattice* Λ can be defined as the set of integral combinations of the linearly independent *lattice basis vectors* $\mathbf{a}_i \in \mathbb{R}^n$:

$$\Lambda \doteq \left\{ \sum_i^n m_i \mathbf{a}_i \mid m_i \in \mathbb{Z} \right\} \tag{1}$$

The lattice is entirely specified by its basis vectors $\mathbf{a}_i$. A lattice is associated with a group of translations T_Λ for which the multiplication rule is addition. This captures the translational symmetry of a crystal. A lattice Λ also defines subsets of $\mathbb{R}^n$ called *unit cells*. These subsets have the property of tilling the space when translated by lattice vectors. Of particular importance is the primitive cell U for the basis $\mathbf{a}_i$:

$$U \doteq \left\{ \sum_i^n x_i \mathbf{a}_i \mid 0 \leq x_i < 1 \right\} \tag{2}$$

In a material, the unit cell comprises a set of atomic positions $S = \{(Z_i, \mathbf{x}_i) \mid \mathbf{x}_i \in U\}$, where the integer Z_i is the atomic number and $\mathbf{x}_i$ the position of the atom. S can contain an arbitrary number

of atoms and must not possess a particular structure. Together with the lattice Λ, the atomic positions provides a complete description of the crystal structure.

It is often useful to define the concept of a *sublattice*. A sublattice Λ_P of Λ is a lattice with basis vectors $\mathbf{b}_i$ such that $\Lambda_P \subset \Lambda$. Correspondingly, the *supercell* C_P is the unit cell associated with the sublattice Λ_P, for which $C_P \supset U$. The full lattice is generated by translations of the sublattice by a set of *centring vectors* $\{\mathbf{0} \dots \mathbf{v}_s\}$. More formally, the sublattice is associated with a normal subgroup T_{Λ_P} of lattice translations, which specifies a coset decomposition of the original lattice $T_\Lambda = (\mathbf{0} + T_{\Lambda_P}) \cup \dots \cup (\mathbf{v}_s + T_{\Lambda_P})$. The centring vectors are coset representatives with respect to that decomposition. It is clear that the centering vectors are also the set of lattice points contained in the supercell C_P.

Example 3.1 (Graphene) In graphene, carbon atoms are arranged in a two-dimensional crystal with honeycomb structure. The underlying lattice Λ has basis vectors $\mathbf{a}_1 = \frac{a}{2}\begin{bmatrix}3 & \sqrt{3}\end{bmatrix}$ and $\mathbf{a}_2 = \frac{a}{2}\begin{bmatrix}3 & -\sqrt{3}\end{bmatrix}$, where a is a lattice constant. The set of atoms in the unit cell is $S = \left\{\left(6, -\frac{a}{2}[1 \quad 0]\right), \left(6, \frac{a}{2}[1 \quad 0]\right)\right\}$

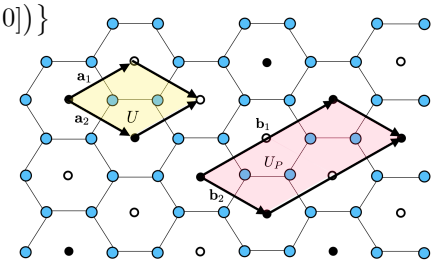


Figure 1: Graphene crystal structure

In Fig.1, blue dots represent carbon atoms. Black and white dots identify points of lattice Λ with unit cell U. Black points identify a sublattice Λ_P defined by basis vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ and with supercell U_P. The centring vectors $\{\mathbf{0}, \mathbf{a}_1\}$ correspond to lattice points contained in the supercell.

Space groups Symmetry plays a major in their description of crystals. This symmetry is often directly visible through facets in naturally occurring crystals. Mathematically, it is described by a *space group* G, the set of isometries that maps a crystal structure to itself. As isometries, space groups are subgroups of the Euclidean group $E(n)$. A space group element can be described as a tuple $(\mathbf{W}, \mathbf{t})$, where $\mathbf{W}$ is the linear part of the transformation and $\mathbf{t}$ a translation. An element maps a vector $\mathbf{x} \in \mathbb{R}^n$ to $\mathbf{W}\mathbf{x} + \mathbf{t}$. The multiplication of space group elements is therefore given by $(\mathbf{W}_1, \mathbf{t}_1)(\mathbf{W}_2, \mathbf{t}_2) = (\mathbf{W}_1\mathbf{W}_2, \mathbf{W}_2\mathbf{t}_1 + \mathbf{t}_2)$. The point group P of a space group is the group obtained from linear part operations in G, which will in general be rotations and reflections. Considering only elements in G for which the linear part is identity $(\mathbf{I}, \mathbf{t})$, we obtain the translation subgroup of the space group T. It is a normal subgroup, which allows defining the factor group G/T isomorphic to the point group P.

The crystallographic restriction theorem guarantees that only certain finite groups are valid point groups of space groups [18]. In particular, in 2 and 3 dimensions, only n-fold rotations with $n \in \{2, 3, 4, 6\}$ are allowed. Two space groups G and G' are said to belong to be of same type if they can be related by a change of coordinate system. There are 17 space group types in 2 dimensions and 230 in 3 dimensions [61]. For a lattice Λ, we call the *Bravais group* of the lattice P_Λ the set of linear isometries that map Λ to itself. Bravais groups provide a way to classify lattices according to their symmetry. We say that two lattices Λ and Λ' belong to the same *Bravais type* if their Bravais groups are the same matrix groups when written for suitable basis vectors of each lattice. The 5 Bravais types in 2 dimensions and the 14 in 3 dimensions are enumerated in the Appendix. The full symmetry group G_Λ of lattice a Λ is given by the semidirect product of its translation group with its Bravais group : $G_\Lambda = T_\Lambda \rtimes P_\Lambda$.

Consider the space group G a crystal structure with lattice Λ and unit cell U. From the definition of a crystal structure, it is clear that the translation subgroup of G will be T_Λ. However, the point group of G will, in general, not be $P = P_\Lambda$. This is because the unit cell may have less symmetry than the underlying lattice, which will result in $P \subseteq P_\Lambda$ (see example 3.2). This is the reason why the number of space groups is much larger than the number of Bravais lattices.

Example 3.2 (Wallpaper pattern) The wallpaper in Figure 2 is described by a square Bravais lattice Λ and unit cell U with 4-fold rotational symmetry. The square lattice has a symmetry group $P4m$, with D_4 Bravais group. However, the unit cell reduces the symmetry of the overall pattern since it does not have reflection axes. The symmetry group of the wallpaper is, therefore, $P4$, with point group C_4.

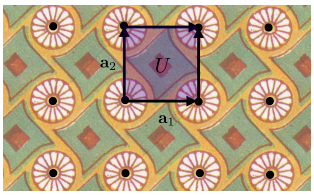


Figure 2: Egyptian wallpaper [34]

4 Limitations of GNNs

In this section, we examine some of the limitations of GNNs on crystalline data, motivating our model by addressing these limitations. Typical models for crystalline data [74, 12] will build a graph from atoms in the unit cell and assign a feature vector $\mathbf{h}_i \in \mathbb{R}^n$ to each. Edge features encoding distance might also be used. For convenience, this graph can be represented as a sparse tensor $\mathbf{H} \in \mathbb{R}^{N^2 \times n}$, where N is the number of atoms considered, and in which node and edge features are encoded on diagonal and off-diagonal entries, respectively. A GNN is a function approximator $f : \mathbb{R}^{N^2 \times n} \to \mathbb{Y}$ that produce a prediction, by successive application of layers $\phi\colon \mathbb{R}^{N^2 \times n_\ell} \to \mathbb{R}^{N^2 \times n_{\ell+1}}$, where ℓ is the layer count.

Expressivity GNNs are usually stack of layers that are permutation equivariant

$$\phi\left((\mathbf{P}(g) \otimes \mathbf{P}(g))\mathbf{H}\right) = (\mathbf{P}(g) \otimes \mathbf{P}(g))\phi\left(\mathbf{H}\right), \forall g \in S_N, \mathbf{H} \in \mathbb{R}^{N^2 \times n}, \tag{3}$$

where $\mathbf{P}(g)$ is the permutation matrix associated with the group element g. While $\mathbf{P}(g)$ permutes nodes, $\mathbf{P}(g) \otimes \mathbf{P}(g)$ permutes the vectorized adjacency matrix. In other words, our use of Kronecker product is due to the equality $\text{vec}(\mathbf{PAP}^\top) = (\mathbf{P} \otimes \mathbf{P})\text{vec}(\mathbf{A})$ for $\mathbf{A} \in \mathbb{R}^{N \times N}$.

This captures the fact that these models are designed to be insensitive to node ordering in $\mathbf{H}$. The material is, in a sense, treated as if it was a molecule. Much of the crystal structure is forgotten because it cannot be captured by ordering the nodes in a specific way and is only encoded in positions or distances. We argue that permutation equivariance is an overly strong requirement and was mainly used for practical reasons. When possible, it should be more beneficial to use a model equivariant to the actual symmetry of the data; in this case, the crystal space groups, which will, in general, be much smaller than the symmetric group. Being equivariant to a smaller group results in less restrictive parameter sharing and a more expressive model.

Note that this requirement is different from that of $E(3)$-equivariance characteristic of some architectures [63, 55, 5]. In these models, the $E(n)$ group only acts on the position $\mathbf{x}_i \in \mathbb{R}^3$ part of each atom's feature vector. By contrast, the symmetry group of a crystal structure maps the crystal to itself, and its action is a permutation. **Revision** *It therefore offers an alternative to building more powerful architectures for these systems that does not suppose that coordinate information is available. In particular, this approach is more suitable for abstract condensed matter systems, like spin and free-fermion lattice models, in which coordinates are not relevant. Moreover, space groups are subgroups of the Euclidean group, therefore equivariance to space groups is less restrictive. Therefore, in this work, we choose to concentrate on space group equivariance. Note that equivariance (and not only invariance) is important, as expressive invariant functions can be built by composing equivariant layers with an output pooling layer. Equivariant functions can also be used to predict local properties like magnetization and charge distribution for example.*

Invariance and symmetry breaking Even if the arrangement of atoms in a crystal is symmetric, this does not have to carry over to all local properties of the material. *Spontaneous symmetry breaking* is common in materials and crucial to describing phenomena such as magnetism and superconductivity [43, 6]. Building a graph only at the unit cell level does not allow to capture local properties that

differ across unit cells. Using a supercell of multiple unit cells does not suffice to solve this problem since equivariant models have the property that identical inputs will be mapped to identical outputs [60, 78].

5 Input representation

Supercell Following the arguments of Section 4 we consider the set of atoms in a supercell instead of only in the unit cell to increase the expressivity of the model. Since this increases the computational complexity of the method, we choose to keep the supercells small and define them with sublattice vectors $\mathbf{b}_i = 2\mathbf{a}_i$. The supercell is therefore 8 times larger than the unit cell, with centring vectors $\left\{\sum_i^3 m_i \mathbf{a}_i \mid m_i \in \{0,1\}\right\}$. For each atom, we build a feature vector with a one-hot encoding of the atomic number and a one-hot encoding identification of the unit cell it belongs to using the closest centering vector. We keep track of the index of each atom within the unit cell, and the index of the atoms' unit cell $\mathbf{h}_{(a,i)} = [\text{onehot}\,(Z_i)\,, \text{onehot}\,(a)]$, where $a \in \{1, \ldots, 8\}$ and $i \in C$, where C is the number of atoms in the unit cell. The encoding of the unit cell allows to break the symmetry between atoms mapped into each other by lattice translations.

Graph We construct a graph from atoms in the supercell, encoding relative distances between atoms as edge features. Inspired by [56], an edge feature vector is built from the distance between atoms $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ as $\mathbf{e}_{ij} = \exp\left(-\gamma\left(d_{ij} - \boldsymbol{\mu}\right)^2\right)$. The vector of Gaussian centers $\boldsymbol{\mu}$ and γ are hyperparameters. **Revision** *We use this approach to facilitate comparaison to previous works [56, 74], although Bessel encondings [38] could also be considered.* It can be seen as "soft" binning of interatomic distances. Using this approach, we use only distance features in contrast to methods that use position vectors. This has the benefit of simplicity while still allowing a complete description of the input structure [69, 4, 66].

Sparsity has proven to be a useful inductive bias in graph representation learning [26], and it is also beneficial in reducing computational complexity. However, atomic bonds are not unambiguously defined in crystals [16, 1]. We choose to follow a similar approach to [32]: an edge is drawn between atoms if they share a Voronoi face and if the distance between atoms is smaller than the sum of atomic Cordero radii plus a cutoff $\Delta = 0.5$Å. This approach has the advantage of being physically sound and producing graphs that are relatively sparse. We provide more details on the graph-building strategy and compute metrics on the resulting graphs in Appendix A.1.

To preserve translational invariance for atoms at the boundary of the supercell, edges are initially also drawn to atoms outside the supercell. Then, if an edge points outside the supercell, its head is mapped to the corresponding representative node inside the supercell. This is analogous to circular padding in image processing.

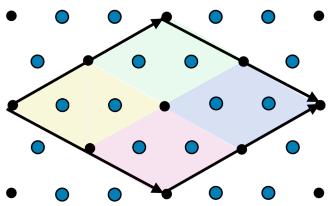
6 Equivariant crystal networks

Product groups Consider a crystal structure with space group G and in which each atom has a feature vector $\mathbf{h}_i$. Then the space group acts as a permutation of the input atoms

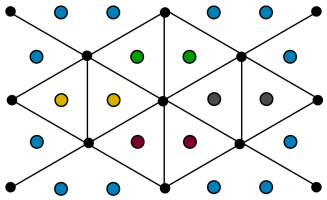
$$g\mathbf{h}_{(a,i)} = \mathbf{h}_{\left(\pi_g^S(a), \pi_g^U(i)\right)} \forall g \in G \tag{4}$$

where π_g^S and π_g^U are the permutations associated with group element g on the supercells and within the unit cells respectively.

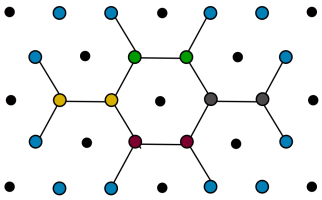
If a dataset contains only samples that share the same crystal structure, then a model equivariant with respect to G can readily be used. However, the case of a dataset with multiple different crystal structures that would be used in a typical supervised learning setting is more challenging for two reasons. First, the samples may have different space groups, which would require different models that would each be trained with a fraction of the data. Second, the group action may be different even for the same space group. This is because the unit cells may have different structures and numbers of atoms.



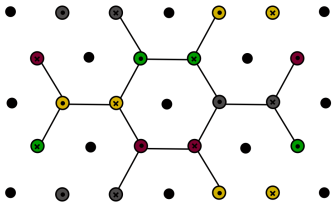
(a) Identification of the supercell. The sublattice vectors are $\mathbf{b}_i = 2\mathbf{a}_i$



(b) Voronoï tessellation.



(c) Drawing edges between atoms that share Voronoï faces. The additional condition related to the Cordero radius is not relevant here.



(d) Periodization of the graph. Edges that point to nodes outside the supercell are mapped back to the corresponding atom inside the supercell. Markings show identical atoms.

Figure 3: Building a graph for the graphene crystal structure from Example 3.1.

A solution to address the second issue is to consider equivariance to a *direct product* of groups $G_\Lambda \times S_C$, where the symmetry group of the lattice G_Λ acts *across* unit cells and S_C *within* unit cells:

$$(g,h)\mathbf{h}_{(a,i)} = \mathbf{h}_{\left(\pi_g^S(a), \pi_h^U(i)\right)} \forall g \in G_\Lambda, h \in S_C \tag{5}$$

Revision *In this way, differences in unit cell structures are dealt with by the symmetric group, where parameter-sharing can handle variable-sized inputs [77]. We still have to accommodate 14 different group actions G_Λ corresponding to the different Bravais lattices. To avoid using a different model for each lattice, we propose two groups to deal with all the lattices. The first option is to consider the least symmetric Bravais lattice of primitive triclinic type and use its symmetry group $P\bar{1} = T_\Lambda \rtimes C_2$. This group is a subgroup of all the other lattice symmetry groups. The second option we consider is to simply use the symmetric group S_Λ that is the symmetric group across unit cells, which is an overgroup of all the lattice symmetry groups. This leaves us with a hierarchy of groups, with S_N, the symmetric group over all atoms of a supercell, being the largest* :

$$\underbrace{P\bar{1} \times S_C}_{P\bar{1}\text{-model}} \subseteq G_\Lambda \times S_C \subseteq \underbrace{S_\Lambda \times S_C}_{S_\Lambda\text{-model}} \subseteq S_N \tag{6}$$

In our experiments, we use the two groups in this hierarchy, which offer different levels of expressivity.

Equivariant message passing Having defined the group action, we can now build the Equivariant Crystal Network (ECN). We will seek to use the message passing framework, which has demonstrated good performance on molecular data, and generalize it to obtain equivariance to other groups than the symmetric group. The update equations for message passing framework are

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij}\right), \quad \mathbf{m}_i = \sum_{j \in N_i} \mathbf{m}_{ij}, \quad \mathbf{h}_i^{t+1} = \phi_h\left(\mathbf{h}_i^t, \mathbf{m}_i^{t+1}\right) \tag{7}$$

The idea is to define parameter-sharing patterns for functions ϕ_e and ϕ_h, such that there can be multiple ones while still retaining equivariance. Following [53], we first define the *parameter-sharing pattern* of the set of input nodes $\mathbb{N}$, with respect to group G as the colored bipartite graph $\Omega \equiv (\mathbb{N}, \alpha, \beta)$, with the edge-color function $\alpha : \mathbb{N} \times \mathbb{N} \to \{1, \ldots, C_e\}$ and node-color function $\beta : \mathbb{N} \to \{1, \ldots, C_h\}$. We also consider the action of the group G on edges Ω as $g \cdot (i,j) \doteq (\pi_g(i), \pi_g(j)) \, \forall g \in G$. We define the orbit $G \cdot (i,j)$ of edge (i,j) as the set of edges in which it can be moved to by the group action : $G \cdot (i,j) \doteq \{g \cdot (i,j) \mid g \in G\}$. A similar definition applies to the orbit of a node, $G \cdot i \doteq \{\pi_g(i) \mid g \in G\}$.

We then make the following claim:

Claim 6.1 *The layer defined by the C_e functions $\phi_e^{\alpha(i,j)}$ and the C_n functions $\phi_h^{\beta(i)}$*

$$\mathbf{m}_{ij} = \phi_e^{\alpha(i,j)}\left(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij}\right) \tag{8}$$

$$\mathbf{m}_i = \sum_{j \in N_i} \mathbf{m}_{ij} \tag{9}$$

$$\mathbf{h}_i^{t+1} = \phi_h^{\beta(i)}\left(\mathbf{h}_i^t, \mathbf{m}_i^t\right) \tag{10}$$

is G-equivariant if the parameter-sharing pattern Ω respects the equivariance condition:

$$\alpha(i,j) = \alpha(k,l) \iff (k,l) \in G \cdot (i,j) \tag{11}$$

$$\beta(i) = \beta(j) \iff j \in G \cdot i \tag{12}$$

The proof of this claim follows in Appendix A.2. The computational process for producing the pattern is to find the orbit of G-action on the edges (nodes) and assign the same color to all edges (nodes) in the same orbit [53]. The computational cost of this orbit-finding process grows linearly with the number of edges (nodes) [30].

This layer generalizes both MPNNs and equivariant multilayer perceptrons, such as CNNs. The MPNN is recovered with $G = S_n$ and a standard CNN with circular convolution with $G = T_\Lambda$, $\phi_e^{\alpha(i,j)}\left(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij}\right) = \mathbf{w}^{\alpha(i,j)} \cdot \mathbf{h}_i^t$ and $\phi_h^{\beta(i)}\left(\mathbf{h}_i^t, \mathbf{m}_i\right) = \text{ReLU}\left(\mathbf{m}_i^{t+1} + \mathbf{b}^{\beta(i)}\right)$

We now consider a product group $G \times H$, acting according to Eq. (5). From Claim 1 of [67], we know that the equivariant linear map for this group will be the Kronecker product of equivariant maps for individual groups; see also [47]. The reformulation for parameter-sharing patterns is the following. If parameters-sharing patterns Ω_1 and Ω_2 satisfy the equivariance condition for G and H respectively, then the parameter-sharing pattern $\Omega = (\mathbb{N} \times \mathbb{M}, \alpha, \beta)$ satisfies the equivariant condition if

$$\alpha : \mathbb{N} \times \mathbb{M} \times \mathbb{N} \times \mathbb{M} \to \{1, \ldots, C_{e,1}\} \times \{1, \ldots, C_{e,2}\}$$
$$\alpha(a,i,b,j) = (\alpha_1(a,b), \alpha_2(i,j))$$

and

$$\beta : \mathbb{N} \times \mathbb{M} \to \{1, \ldots, C_{h,1}\} \times \{1, \ldots, C_{h,2}\} \tag{13}$$

$$\beta(a,i) = (\beta_1(a), \beta_2(i)) \tag{14}$$

This simply means that a new color is defined in the product pattern for each possible combination of colors in the original patterns.

For our model, we choose functions $\phi_e^{\alpha(i,j)}$ and $\phi_h^{\beta(i)}$ to be MLPs with one hidden layer. In addition, we add a weighting factor to the edge aggregation 9, as this as been shown to be beneficial by [74, 55]:

$$\mathbf{m}_i = \sum_{j \in N_i} e_{ij}\mathbf{m}_{ij}, \quad \text{where} \quad e_{ij} = \phi_a(\mathbf{m}_{ij}) \tag{15}$$

and ϕ_a is simply a linear layer. This change does not affect the equivariance of the model.

Example 6.1 In our running example, the parameter-sharing pattern for the message passing is produced by the Kronecker product of the pattern for the $P2$ group (see Appendix A.3 for a similar example with the $P6m$ group) and the pattern for the symmetric group S_2 as shown in the figure below (Top left). However, we only need to keep the colors for which there is a corresponding edge (Top right). Message passing is used on the resulting edge/node colored graph where similar colored nodes and edges use the same functions in message passing.

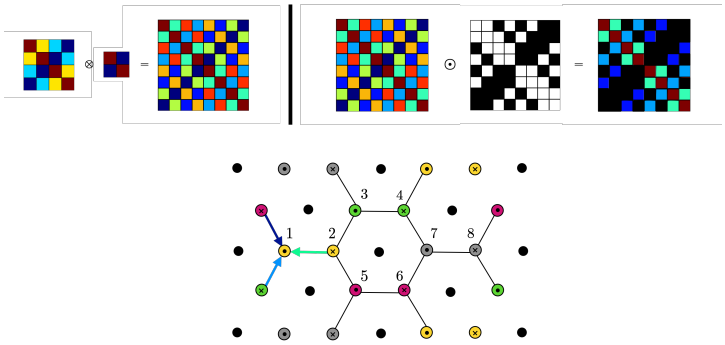


Figure 4: Equivariant message passing for graphene

7 Model implementation

Network architecture We aimed to keep the architecture of our ECN model as simple as possible. The model receives crystal structure graphs as input, with one-hot encoded feature vectors for each atom. We use 128-dimensional embeddings and keep the same dimension for hidden layers throughout the network. The ECN consists of 6 layers of the equivariant message passing operation 6.1. This is followed by a mean-pooling operation over node embeddings of each graph and a two-layer MLP that outputs the final prediction. Mean pooling is preferred to other options because we only predict *intensive* physical quantities. These properties are "per-atom" and do not depend on the choice of supercell. Therefore, selecting a pooling operation that respects this invariance makes sense. If we were to predict *extensive* quantities, sum pooling should be the preferred option.

Input features Many choices to encode atomic features are possible and have been suggested in the literature. We perform experiments over these variants. For each atom, we encode its type in a feature vector. We consider two strategies, using only information available from the periodic table. The first one is to use only the atomic number. The second one is to use the one-hot encoded group and period numbers for each atom. The second strategy could benefit from promoting better generalization since embeddings of atoms belonging to the same group or period will show a certain similarity. Around 24% of the structures retained in the Materials Project dataset have been computed using the Hubbard-U extension of DFT. Since this information can significantly influence the resulting properties [64] it is added to the atomic feature vector as a binary feature.

Other details We implemented our model using Pytorch [51]. We use a sparse implementation of the equivariant message-passing based on the Pytorch Scatter package [23]. We use the AdamW optimizer [45] with weight decay regularization. The full hyperparameter setup is provided in Appendix A.5.

8 Experiments

Materials Project We perform experiments using the Materials Project dataset [33] [1]. This standard dataset of materials informatics comprises more than 120K materials with a complete specification of their crystal structure and some important physical properties obtained with high-throughput DFT calculations.

[1] We use version 2021.05, the most recent at the time of publication

Table 1: Results on the Materials Project dataset.

	Method	Property					
		E(eV/atom)	E_F (eV)	M (μ_B/atom)	E_g (eV)	Metal precision	Nonmetal precision
Original	CGCNN	0.039	0.363	-	0.388	80%	95%
	MEGNET	0.028	-	-	0.33	78.9%	90.6%
	SCHNET	0.041	-	-	-	-	-
Ours	CGCNN	0.048	0.310	0.110	0.399	**79.57%**	**87.5%**
	MEGNET	0.056	0.303	0.108	0.422	79.21%	84.4%
	ECN-$P\bar{1}$	0.052 ±0.001	0.303 ±0.004	0.108 ±0.002	0.44 ±0.02	80% ±4.0	84% ±4.0
	ECN-S_Λ	**0.046 ±0.002**	**0.281 ±0.007**	**0.106 ±0.002**	**0.390 ±0.020**	79.8% 2.0	83.2% 1.0

The Materials Project dataset has not been initially built to serve as a machine learning benchmark; we perform some preprocessing to make it suitable. First, since multiple DFT calculations were sometimes performed on close initial configurations, some samples only show marginal structure differences and resulting properties. This is exemplified by the compound $Li_9Mn_2Co_5O_{16}$, which appears 322 times in the database. Such duplicates can result in training-test leakage. To prevent this, we consider two structures redundant if they have the same unit cell chemical formula and the same space group. Amongst a set of duplicate structures, the one with the lowest formation energy is chosen. Second, we filter out one-dimensional and two-dimensional materials from the dataset to only keep three-dimensional materials. Finally, we remove materials for which the unit cell contains more than 50 atoms. These are often associated with molecular and inorganic crystals with very different properties than the other materials. Training, validation, and test splits are 80%, 10%, and 10% of the dataset. We provide statistics on the processed dataset in Appendix A.4.

Following previous work, we predict a few relevant energetic properties: the formation energy E, the Fermi energy E_F, and the band gap E_g for the subset of insulating materials. We also predict the binary insulator or conductor character material. Finally, we also predict the magnetic moment per atom M. This can be seen as a graph regression or classification task. For regression, training is performed using the mean-squared error (MSE) loss function, but we report the mean-absolute error (MAE). For classification, we use the cross-entropy loss function.

Revision *We compare the results obtained by our models to baselines [56, 74, 12]. Note that because these papers used different training and test splits and preprocessing schemes (even between themselves), our results cannot be directly compared. To alleviate that, we trained our own versions of two of the baselines using our splits and a similar training procedure. We obtain slightly better or comparable results to the baselines on all targets when evaluated on the same splits. The S_Λ version offers better performance than the $P\bar{1}$ model overall, showing that it is more beneficial to lean on the side of having slightly more symmetry than necessary at the cost of some expressivity. We provide additional results for the model variants in Appendix A.6. The benefits of the increased expressivity on this task is in the not crucial, which we think can be explained in part by the relatively small size of the Materials Project dataset. In a larger data regime, we expect that the benefit of increased expressivity will outweight the cost in generalization capability.*

Perov-5 *Finally, we perform experiments using the Perov-5 dataset [9] as provided by [73]. In this dataset, all the materials share the same Perovskite crystal structure. The task considered is the regression of the heat of formation computed through DFT. Results are shown on Table 2.The improvement on this dataset is significantly more important for the proposed model compared to the baselines than on the Materials Project dataset. We hypothesize that the fact that all the structures are shared in this dataset allows the model to specialize more efficiently leading to better generalization.*

Method	Property Heat All
CGCNN	0.047 ±0.000
MEGNET	0.059 ±0.006
ECN-S_Λ	**0.038 ±0.004**

Table 2: Perov-5 results

Conclusion

We have shown how to leverage crystal symmetry to build more expressive and physically motivated neural networks for materials data. This allows us to obtain a close equivalent of group equivariant convolution on this data structure. These models show excellent accuracy in supervised property prediction, which supports the idea that symmetry is a useful inductive bias. Such models could be used for other tasks on materials such as dynamics prediction **Revision** *, if the dynamics approximately preserves the crystal structure. We also think that these models have significant potential on more abstract condensed matter systems such as spin models and free-fermion models.* We have also defined equivariant message passing, a generalization of the MPNN framework that can potentially be used on any data structures for which a group can capture the symmetry in sparse interactions between the basic elements. One limitation

of this approach and area of future improvement is on the computational efficiency of the equivariant message passing, which does not benefit from optimized algorithms available for convolutions.

References

[1] *Alvarez Santiago*. A cartography of the van der Waals territories // Dalton Trans. 2013. 42. 8617–8636.

[2] *Anderson Brandon, Hy Truong Son, Kondor Risi*. Cormorant: Covariant molecular neural networks // Advances in neural information processing systems. 2019. 32.

[3] *Ashcroft Neil W, Mermin N David, others* . Solid state physics. 1976.

[4] *Bartók Albert P, Kondor Risi, Csányi Gábor*. On representing chemical environments // Physical Review B. 2013. 87, 18. 184115.

[5] *Batzner Simon, Musaelian Albert, Sun Lixin, Geiger Mario, Mailoa Jonathan P., Kornbluth Mordechai, Molinari Nicola, Smidt Tess E., Kozinsky Boris*. E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials. 2021.

[6] *Beekman Aron, Rademaker Louk, Wezel Jasper van*. An introduction to spontaneous symmetry breaking // SciPost Physics Lecture Notes. 2019. 011.

[7] *Bengio Y., Courville A., Vincent P.* Representation Learning: A Review and New Perspectives // IEEE Transactions on Pattern Analysis and Machine Intelligence. Aug 2013. 35, 8. 1798–1828.

[8] *Bronstein Michael M, Bruna Joan, Cohen Taco, Velic̆ković Petar*. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges // arXiv preprint arXiv:2104.13478. 2021.

[9] *Castelli Ivano E, Landis David D, Thygesen Kristian S, Dahl Søren, Chorkendorff Ib, Jaramillo Thomas F, Jacobsen Karsten W*. New cubic perovskites for one-and two-photon water splitting using the computational materials repository // Energy & Environmental Science. 2012. 5, 10. 9034–9043.

[10] *Cesa Gabriele, Lang Leon, Weiler Maurice*. A Program to Build E (N)-Equivariant Steerable CNNs // International Conference on Learning Representations. 2021.

[11] *Chanussot* Lowik, Das* Abhishek, Goyal* Siddharth, Lavril* Thibaut, Shuaibi* Muhammed, Riviere Morgane, Tran Kevin, Heras-Domingo Javier, Ho Caleb, Hu Weihua, Palizhati Aini, Sriram Anuroop, Wood Brandon, Yoon Junwoong, Parikh Devi, Zitnick C. Lawrence, Ulissi Zachary*. Open Catalyst 2020 (OC20) Dataset and Community Challenges // ACS Catalysis. 2021.

[12] *Chen Chi, Ye Weike, Zuo Yunxing, Zheng Chen, Ong Shyue Ping*. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals // Chemistry of Materials. 2019. 31, 9. 3564–3572.

[13] *Cohen Taco, Welling Max*. Group equivariant convolutional networks // International conference on machine learning. 2016. 2990–2999.

[14] *Cohen Taco S., Geiger Mario, Köhler Jonas, Welling Max*. Spherical CNNs // International Conference on Learning Representations. 2018.

[15] *Cohen Taco S, Geiger Mario, Weiler Maurice*. A general theory of equivariant cnns on homogeneous spaces // Advances in neural information processing systems. 2019. 32.

[16] *Cordero Beatriz, Gómez Verónica, Platero-Prats Ana E., Revés Marc, Echeverría Jorge, Cremades Eduard, Barragán Flavia, Alvarez Santiago*. Covalent radii revisited // Dalton Trans. 2008. 2832–2838.

[17] *Court Callum J., Cole Jacqueline M.* Magnetic and superconducting phase diagrams and transition temperatures predicted using text mining and machine learning // npj Computational Materials. Mar 2020. 6, 1.

[18] *Coxeter H. S. M.* Introduction to geometry. New York: John Wiley & Sons Inc., 1961. xvii+443.

[19] *Curtarolo Stefano, Setyawan Wahyu, Hart Gus L.W., Jahnatek Michal, Chepulskii Roman V., Taylor Richard H., Wang Shidong, Xue Junkai, Yang Kesong, Levy Ohad, Mehl Michael J., Stokes Harold T., Demchenko Denis O., Morgan Dane*. AFLOW: An automatic framework for high-throughput materials discovery // Computational Materials Science. 2012. 58. 218 – 226.

[20] *Dehmamy Nima, Walters Robin, Liu Yanchen, Wang Dashun, Yu Rose*. Automatic Symmetry Discovery with Lie Algebra Convolutional Network // Advances in Neural Information Processing Systems. 2021. 34.

[21] *Dresselhaus Mildred S, Dresselhaus Gene, Jorio Ado*. Group theory: application to the physics of condensed matter. 2007.

[22] *Esteves Carlos, Makadia Ameesh, Daniilidis Kostas*. Spin-weighted spherical cnns // Advances in Neural Information Processing Systems. 2020. 33. 8614–8625.

[23] *Fey Matthias*. PyTorch Scatter. 2022.

[24] *Finkelshtein Ben, Baskin Chaim, Maron Haggai, Dym Nadav*. A simple and universal rotation equivariant point-cloud network // arXiv preprint arXiv:2203.01216. 2022.

[25] *Finzi Marc, Welling Max, Wilson Andrew Gordon*. A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups // arXiv preprint arXiv:2104.09459. 2021.

[26] *Garg Vikas, Jegelka Stefanie, Jaakkola Tommi*. Generalization and representational limits of graph neural networks // International Conference on Machine Learning. 2020. 3419–3430.

[27] *Gilmer Justin, Schoenholz Samuel S., Riley Patrick F., Vinyals Oriol, Dahl George E.* Neural Message Passing for Quantum Chemistry // Proceedings of the 34th International Conference on Machine Learning. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017. 1263–1272. (Proceedings of Machine Learning Research).

[28] *Hasnip Philip J., Refson Keith, Probert Matt I. J., Yates Jonathan R., Clark Stewart J., Pickard Chris J.* Density functional theory in the solid state // Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2014. 372, 2011. 20130270.

[29] *Hiß Gerhard, Holt Derek F, Newman Michael F*. Computational Group Theory // Oberwolfach Reports. 2007. 3, 3. 1795–1878.

[30] *Holt Derek F, Eick Bettina, O'Brien Eamonn A*. Handbook of computational group theory. 2005.

[31] *Horton Matthew Kristofer, Montoya Joseph Harold, Liu Miao, Persson Kristin Aslaug*. High-throughput prediction of the ground-state collinear magnetic order of inorganic materials using Density Functional Theory // npj Computational Materials. jun 2019. 5, 1.

[32] *Isayev Olexandr, Oses Corey, Toher Cormac, Gossett Eric, Curtarolo Stefano, Tropsha Alexander*. Universal fragment descriptors for predicting properties of inorganic crystals // Nature Communications. Jun 2017. 8, 1.

[33] *Jain Anubhav, Ong Shyue Ping, Hautier Geoffroy, Chen Wei, Richards William Davidson, Dacek Stephen, Cholia Shreyas, Gunter Dan, Skinner David, Ceder Gerbrand, al. et*. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation // APL Materials. Jul 2013. 1, 1. 011002.

[34] *Jones Bedford Francis Waring J. B. Westwood J. O. Wyatt M. Digby Day & Son Owen*. The Grammar of Ornament. London: Published by Day and Son, Lithographers to the Queen, Gate Street, Lincoln's Inn Fields, 1856.

[35] *Kailkhura Bhavya, Gallagher Brian, Kim Sookyung, Hiszpanski Anna, Han T. Yong-Jin*. Reliable and explainable machine-learning methods for accelerated material discovery // npj Computational Materials. Nov 2019. 5, 1.

[36] *Kipf Thomas N., Welling Max*. Semi-Supervised Classification with Graph Convolutional Networks // 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. 2017.

[37] *Kirklin Scott, Saal James E, Meredig Bryce, Thompson Alex, Doak Jeff W, Aykol Muratahan, Rühl Stephan, Wolverton Chris*. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies // npj Computational Materials. 2015. 1. 15010.

[38] *Klicpera Johannes, Groß Janek, Günnemann Stephan*. Directional Message Passing for Molecular Graphs // International Conference on Learning Representations. 2020.

[39] *Kohn W., Sham L. J.* Self-Consistent Equations Including Exchange and Correlation Effects // Phys. Rev. Nov 1965. 140. A1133–A1138.

[40] *Kondor Risi, Son Hy Truong, Pan Horace, Anderson Brandon, Trivedi Shubhendu*. Covariant compositional networks for learning graphs // arXiv preprint arXiv:1801.02144. 2018.

[41] On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups. // . 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018. 2747–2755. (Proceedings of Machine Learning Research).

[42] *Kusne Aaron Gilad, Gao Tieren, Mehta Apurva, Ke Liqin, Nguyen Manh Cuong, Ho Kai-Ming, Antropov Vladimir, Wang Cai-Zhuang, Kramer Matthew J., Long Christian, Takeuchi Ichiro*. On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets // Scientific Reports. sep 2014. 4, 1.

[43] *Landau Lev Davidovich*. On the theory of phase transitions. I. // Phys. Z. Sowjet. 1937. 11. 26.

[44] *LeCun Yann, Bengio Yoshua, others* . Convolutional networks for images, speech, and time series // The handbook of brain theory and neural networks. 1995. 3361, 10. 1995.

[45] *Loshchilov Ilya, Hutter Frank*. Decoupled Weight Decay Regularization // International Conference on Learning Representations. 2019.

[46] *Maron Haggai, Ben-Hamu Heli, Shamir Nadav, Lipman Yaron*. Invariant and Equivariant Graph Networks // International Conference on Learning Representations. 2019.

[47] *Maron Haggai, Litany Or, Chechik Gal, Fetaya Ethan*. On learning sets of symmetric elements // International Conference on Machine Learning. 2020. 6734–6744.

[48] *Noé Frank, Tkatchenko Alexandre, Müller Klaus-Robert, Clementi Cecilia*. Machine Learning for Molecular Simulation // Annual Review of Physical Chemistry. 2020. 71, 1. 361–390. PMID: 32092281.

[49] *O'Mara Jordan, Meredig Bryce, Michel Kyle*. Materials Data Infrastructure: A Case Study of the Citrination Platform to Examine Data Import, Storage, and Access // JOM. Jun 2016. 68, 8. 2031–2034.

[50] *Ong Shyue Ping, Richards William Davidson, Jain Anubhav, Hautier Geoffroy, Kocher Michael, Cholia Shreyas, Gunter Dan, Chevrier Vincent L., Persson Kristin A., Ceder Gerbrand*. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis // Computational Materials Science. 2013. 68. 314 – 319.

[51] *Paszke Adam, Gross Sam, Massa Francisco, Lerer Adam, Bradbury James, Chanan Gregory, Killeen Trevor, Lin Zeming, Gimelshein Natalia, Antiga Luca, others* . Pytorch: An imperative style, high-performance deep learning library // Advances in neural information processing systems. 2019. 32. 8026–8037.

[52] *Qi Charles R, Su Hao, Mo Kaichun, Guibas Leonidas J*. Pointnet: Deep learning on point sets for 3d classification and segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. 652–660.

[53] *Ravanbakhsh Siamak, Schneider Jeff G., Póczos Barnabás*. Equivariance Through Parameter-Sharing // ICML. 2017. 2892–2901.

[54] *Rotman Joseph J*. An introduction to the theory of groups. 148. 2012.

[55] *Satorras Victor Garcia, Hoogeboom Emiel, Fuchs Fabian B, Posner Ingmar, Welling Max*. E (n) equivariant normalizing flows // arXiv preprint arXiv:2105.09016. 2021.

[56] *Schütt Kristof, Kindermans Pieter-Jan, Felix Huziel Enoc Sauceda, Chmiela Stefan, Tkatchenko Alexandre, Müller Klaus-Robert*. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions // Advances in Neural Information Processing Systems. 2017. 991–1001.

[57] *Shakerinava Mehran, Ravanbakhsh Siamak*. Equivariant Networks for Pixelized Spheres // Proceedings of the 38th International Conference on Machine Learning. 139. 18–24 Jul 2021. 9477–9488. (Proceedings of Machine Learning Research).

[58] *Shakerinava Mehran, Ravanbakhsh Siamak*. Equivariant networks for pixelized spheres // International Conference on Machine Learning. 2021. 9477–9488.

[59] *Shawe-Taylor J.* Building symmetries into feedforward networks // 1989 First IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313). 1989. 158–162.

[60] *Smidt Tess E., Geiger Mario, Miller Benjamin Kurt*. Finding symmetry breaking order parameters with Euclidean neural networks // Phys. Rev. Research. Jan 2021. 3. L012002.

[61] *Souvignier B.* A general introduction to space groups // International Tables for Crystallography. 2016. 1.3, 22–41.

[62] *Stanev Valentin, Oses Corey, Kusne A. Gilad, Rodriguez Efrain, Paglione Johnpierre*. Machine learning modeling of superconducting critical temperature // npj Computational Materials. December 2018. 4, 2.

[63] *Thomas Nathaniel, Smidt Tess, Kearnes Steven, Yang Lusann, Li Li, Kohlhoff Kai, Riley Patrick*. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds // arXiv preprint arXiv:1802.08219. 2018.

[64] *Tolba Sarah A., Gameel Kareem M., Ali Basant A., Almossalami Hossam A., Allam Nageh K.* The DFT+U: Approaches, Accuracy, and Applications // Density Functional Calculations. Rijeka: IntechOpen, 2018. 1.

[65] *Velicković Petar, Cucurull Guillem, Casanova Arantxa, Romero Adriana, Liò Pietro, Bengio Yoshua*. Graph Attention Networks // International Conference on Learning Representations. 2018.

[66] *Villar Soledad, Hogg David W, Storey-Fisher Kate, Yao Weichi, Blum-Smith Ben*. Scalars are universal: Equivariant machine learning, structured like classical physics // Advances in Neural Information Processing Systems. 2021.

[67] *Wang Renhao, Albooyeh Marjan, Ravanbakhsh Siamak*. Equivariant Maps for Hierarchical Structures. 2020.

[68] *Weiler Maurice, Geiger Mario, Welling Max, Boomsma Wouter, Cohen Taco*. 3d steerable cnns: Learning rotationally equivariant features in volumetric data // Advances in Neural Information Processing Systems. 2018. 10381–10392.

[69] *Weyl Hermann*. The Classical Groups. 1939.

[70] *Wolpert D.H., Macready W.G.* No free lunch theorems for optimization // IEEE Transactions on Evolutionary Computation. 1997. 1, 1. 67–82.

[71] *Worrall Daniel E, Garbin Stephan J, Turmukhambetov Daniyar, Brostow Gabriel J*. Harmonic networks: Deep translation and rotation equivariance // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. 5028–5037.

[72] *Wu Yao, Guo Jie, Sun Rui, Min Jie*. Machine learning for accelerating the discovery of high-performance donor/acceptor pairs in non-fullerene organic solar cells // npj Computational Materials. 2020. 6, 1. 120.

[73] *Xie Tian, Fu Xiang, Ganea Octavian-Eugen, Barzilay Regina, Jaakkola Tommi*. Crystal Diffusion Variational Autoencoder for Periodic Material Generation // arXiv preprint arXiv:2110.06197. 2021.

[74] *Xie Tian, Grossman Jeffrey C.* Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties // Physical Review Letters. Apr 2018. 120, 14.

[75] *Xu Keyulu, Hu Weihua, Leskovec Jure, Jegelka Stefanie*. How Powerful are Graph Neural Networks? // International Conference on Learning Representations. 2019.

[76] *Ye Weike, Chen Chi, Wang Zhenbin, Chu Iek-Heng, Ong Shyue Ping*. Deep neural networks for accurate predictions of crystal stability // Nature Communications. sep 2018. 9, 1.

[77] *Zaheer Manzil, Kottur Satwik, Ravanbakhsh Siamak, Poczos Barnabas, Salakhutdinov Russ R, Smola Alexander J*. Deep Sets // Advances in Neural Information Processing Systems 30. 2017. 3391–3401.

[78] *Zhang Yan, Zhang David W, Lacoste-Julien Simon, Burghouts Gertjan J., Snoek Cees G. M.* Multiset-Equivariant Set Prediction with Approximate Implicit Differentiation // International Conference on Learning Representations. 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] On both the theoretical and experimental side, the paper's contributions are aligned with the abstract and introduction.
   (b) Did you describe the limitations of your work? [Yes] See Conclusion
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not think that this is relevant to this work
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Theorem 6.1 and its proof in the Appendix A.2
   (b) Did you include complete proofs of all theoretical results? [Yes] The only proof is included in Appendix A.2
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The details needed to reproduce the experiments are included in the supplementary material Appendix A.5. A link to the relevant code will also be accessible upon publication of the paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The details needed to reproduce the experiments are included in the supplementary material Appendix A.5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] This is due to a lack of computational resources.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] This is provided in the supplementary material Appendix A.5
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Introduction and Model implementation sections
   (b) Did you mention the license of the assets? [Yes] This is mentioned in the supplementary material Appendix A.4
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] The used dataset will be provided upon publication of the paper.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] This is not relevant according to the license of the used assets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] This is not relevant for this paper.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Table 3: Bravais lattices in 2 and 3 dimensions.

BRAVAIS TYPE	SYMMETRY GROUP	ABSTRACT POINT GROUP
2 dimensions		
Oblique	$P2$	C_2
Rectangular	Pmm	D_2
Centered rectangular	Cmm	D_2
Square	$P4m$	D_4
Hexagonal	$P6m$	D_6
3 dimensions		
Primitive triclinic	$P1$	C_2
Primitive monoclinic	$P2/m$	C_{2h}
Base-centered monoclinic	$C2/m$	C_{2h}
Primitive orthorhombic	$Pmmm$	D_{2h}
Base-centered orthorhombic	$Cmmm$	D_{2h}
Body-centered orthorhombic	$Immm$	D_{2h}
Face-centered orthorhombic	$Fmmm$	D_{2h}
Primitive tetragonal	$P4/mmm$	D_{4h}
Body-centered tetragonal	$C4/mmm$	D_{4h}
Rhombohedral	$R\bar{3}m$	D_{3d}
Hexagonal	$P6/mmm$	D_{6h}
Primitive cubic	$Pm\bar{3}m$	O_h
Body-centered cubic	$Im\bar{3}m$	O_h
Face-centered cubic	$Fm\bar{3}m$	O_h

A.1 Graph-building strategies

Revision *The graphs were built using the `IsayevNN` class from the `pymatgen` [50] package. It implements the commonly used Voronoi tessalation to define neighbors. Two atoms are considered bonded if they share a face in the Voronoi tessalation of the supercell and their distance is less than the sum of the atomic Cordero radii (a measure of the atomic radius) plus a cutoff $\Delta = 0.5$Å. This value of the cutoff was increase compared to [32] to reduce the number of disconnected graphs.*

We provide statistics for the graphs obtained by the method described in Section 5. A hard cutoff on atomic distances of 6Å is also imposed on atomic distances.

A.2 Proof of Claim 6.1

We prove that the function defined by

$$\mathbf{m}_{ij} = \phi_e^{\alpha(i,j)}\left(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij}\right), \tag{16}$$

$$\mathbf{m}_i = \sum_{j \in N_i} \mathbf{m}_{ij}, \tag{17}$$

$$\mathbf{h}_i^{t+1} = \phi_h^{\beta(i)}\left(\mathbf{h}_i^t, \mathbf{m}_i^t\right), \tag{18}$$

with

$$\alpha(i,j) = \alpha(k,l) \iff (k,l) \in G \cdot (i,j), \tag{19}$$

$$\beta(i) = \beta(j) \iff j \in G \cdot i, \tag{20}$$

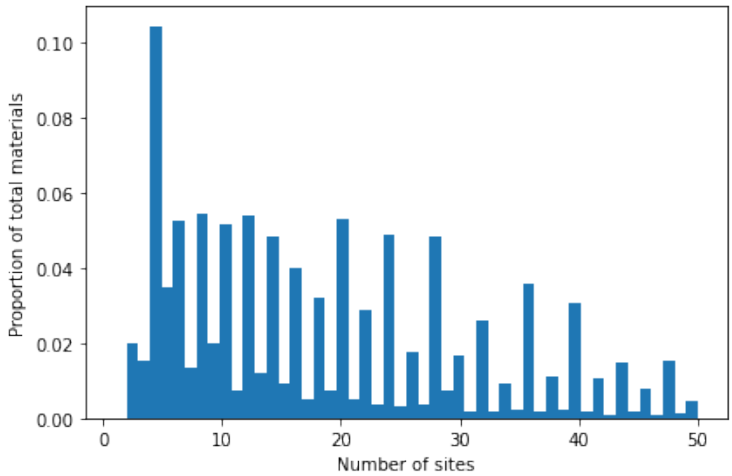


Figure 5: Histogram of the number of primitive cell sites per material in the processed Materials Project dataset.

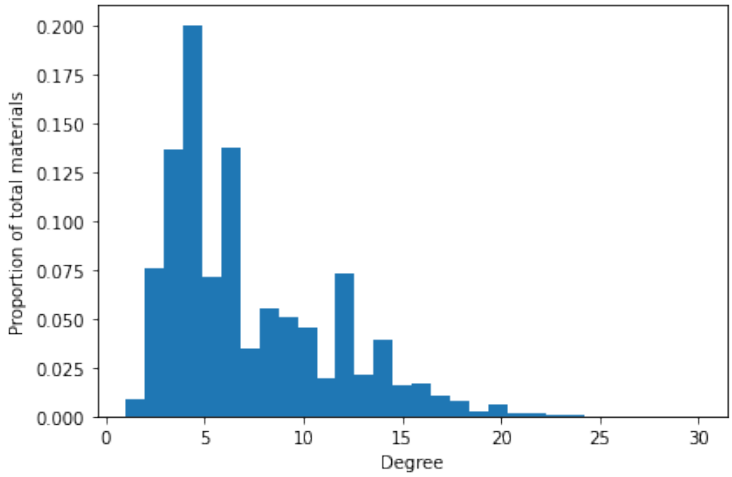


Figure 6: Degree distribution for all graphs.

is G-equivariant by proving the equivariance of every step and using the fact that function composition preserves equivariance.

First, we need to show equivariance of the message function on Equation (16)

$$\mathbf{m}_{\pi_g(i),\pi_g(j)} = \phi_e^{\alpha(i,j)}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{h}^t_{\pi_g(j)}, \mathbf{e}_{\pi_g(i),\pi_g(i)}\right) \forall g \in G. \tag{21}$$

Using Equation (19), we have

$$\phi_e^{\alpha(i,j)}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{h}^t_{\pi_g(j)}, \mathbf{e}_{\pi_g(i),\pi_g(i)}\right) = \phi_e^{\alpha(\pi_g(i),\pi_g(j))}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{h}^t_{\pi_g(j)}, \mathbf{e}_{\pi_g(i),\pi_g(i)}\right) \forall g \in G. \tag{22}$$

The right-hand side is equal to $\mathbf{m}_{\pi_g(i),\pi_g(j)}$ by definition.

The message aggregation step at Equation (17) is permutation equivariant

$$\mathbf{m}_{\pi_g(i)} = \sum_{\pi_g(j) \in N_{\pi_g(i)}} \mathbf{m}_{\pi_g(i),\pi_g(j)}. \tag{23}$$

Finally, we need to show that the node function on Equation (18) is also equivariant

$$\mathbf{h}^{t+1}_{\pi_g(i)} = \phi_h^{\beta(i)}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{m}^t_{\pi_g(i)}\right) \forall g \in G. \tag{24}$$

Using Equation (20), we find

$$\phi_h^{\beta(i)}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{m}^t_{\pi_g(i)}\right) = \phi_h^{\beta(\pi_g(i))}\left(\mathbf{h}^t_{\pi_g(i)}, \mathbf{m}^t_{\pi_g(i)}\right) \forall g \in G. \tag{25}$$

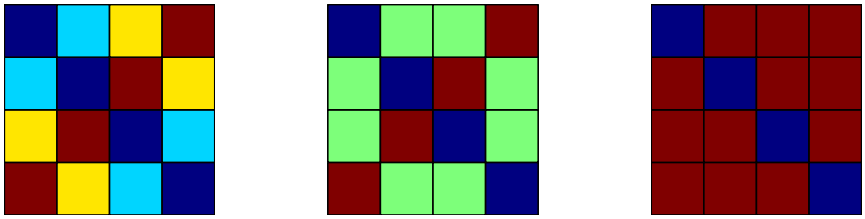
The right-hand side is equal to $\mathbf{h}^{t+1}_{\pi_g(i)}$ by definition.

A.3 Parameter sharing patterns

Revision *The parameter sharing patterns are computed using a group-theoretical orbit finding algorithm that has linear complexity in the number of size of the generating sets of a group and in the number of edges in the coloured bipartite graphs [29, 58].*

We show the parameter sharing patterns for the different Bravais lattice groups (Figure 7 and Figure 8). For the 2-dimensional groups, we use a 2×2 supercell and for the 3-dimensional groups a $2 \times 2 \times 2$ supercell. Note that the numbering of the unit cells within the supercell is chosen by convention and can vary for different lattices.

Notice that the parameter-sharing patterns for different groups can be the same. This is because, for different groups, the group action can induce the same orbits on the bipartite graph. In particular, for some groups (Figure 7c), the pattern collapses to that of the symmetric group. This can be undesirable since it reduces expressivity. This can be alleviated by using larger supercells or eliminated by considering the group acting on itself instead of on supercells, as done in [13].



(a) Pattern for groups $P2$ and Pmm (b) Pattern for group Cmm (c) Pattern for groups $P4m$ and $P4m$ and S_4

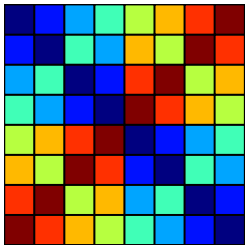
Figure 7: Parameter sharing patterns for 2-dimensional Bravais lattices groups

A.4 Materials Project dataset

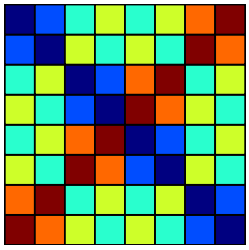
We hereafter report the number of samples in the Materials Project dataset at different levels of the preprocessing scheme :

(a) Full dataset
(b) No duplicates
(c) No duplicates and unit cell size constraint
(d) No duplicates, unit cell size constraint and 3D materials
(e) No duplicates, unit cell size constraint, 3D materials and valid graphs
(f) Insulators, no duplicates, unit cell size constraint, 3D materials and valid graphs

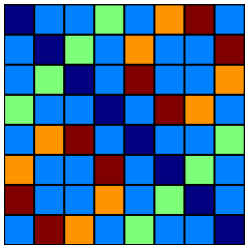
We also report the mean and standard deviation of each target for the processed dataset



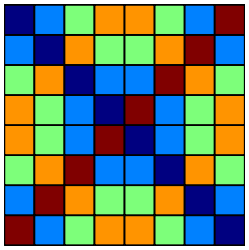
(a) Pattern for groups $P\bar{1}$, $P2/m$, $Pmmm$



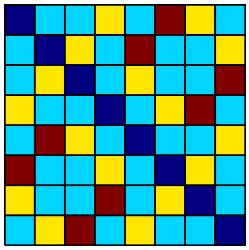
(b) Pattern for groups $C2/m$, $Cmmm$, $P4/mmm$



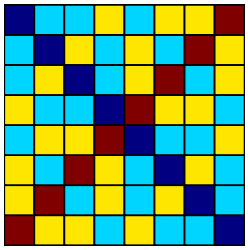
(c) Pattern for groups $Immm$



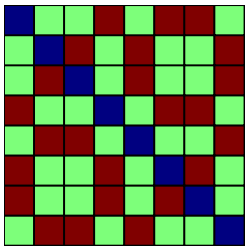
(d) Pattern for groups $Fmmm$



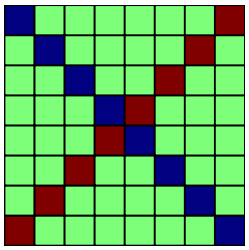
(e) Pattern for groups $C4/mmm$



(f) Pattern for groups $R\bar{3}m$, $P6/mmm$ and $Pm\bar{3}m$



(g) Pattern for groups $Im\bar{3}m$



(h) Pattern for groups $Fm\bar{3}m$

Figure 8: Parameter sharing patterns for 3-dimensional Bravais lattices groups

Table 4: Number of entries in the Materials Project dataset with processing

(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
126126	114605	96315	82229	78649	33971

A.5 Hyperparameters

We use the same training setup for all the models. The learning rate is initialized at 1×10^{-3} with a scheduler halving it after each 25 epoch plateau on the validation loss. Training is perfomed for 1000 epochs or until the learning rate reaches 1×10^{-6}. **Revision** *We performed sweeps over learning rates for all the models to verify that this is indeed a setup in which all models train well.*

Revision *For the ECN models, we optimized the number of layers, the embedding dimension, the weight decay parameter and dropout (without noticing significant improvement).* We use 6 layers of message passing operations. The $\phi_e^{\alpha(i,j)}$ and $\phi_h^{\beta(i)}$ functions are 1-hidden layer MLPs. For the $\phi_e^{\alpha(i,j)}$ functions, the weights for the hidden layer and output layers are shared across all $\alpha\,(i,j)$. This was found to perform slightly better. The SWISH activation function is used. The feature dimension of node embeddings is 100 across all the network. For edge embeddings, the dimension is set at 20.

Table 5: Targets

Property	E(eV/atom)	E_F (eV)	M (μ_B/atom)	E_g (eV)
Mean	-1.42	3.77	0.16	2.02
Std. dev.	1.13	2.63	0.44	1.56

For the CGCNN model, we used the same embedding sizes (they were not specified in the original paper) and 2 convolution layers. For MEGNet, we used the same architecture setup as in the original paper.

A.6 Supplementary results

Method	**Property**					
	E(eV/atom)	E_F (eV)	M (μ_B/atom)	E_g (eV)	Metal precision	Nonmetal precision
ECN-S_Λ Hubbard	0.052	0.290	0.109	0.368	81.2%	82.9%
ECN-S_Λ Group+Period	0.051	0.295	0.109	0.386	77.6%	85.6%

Table 6: Results for model variants on Materials Project

Method	**Inference time** Heat All
CGCNN	0.047 $\pm$0.000
MEGNET	0.059 $\pm$0.006
ECN-S_Λ	**0.038 $\pm$0.004**

Table 7: Inference time for Materials Project