Point Group Equivariant Networks for Material Systems

Thesis Proposal

Alex Heilman

July 23, 2025

Overview

Graph Neural Networks for Materials Science

Overview

Graph Neural Networks for Materials Science

- Hypergraph representations

Overview

Graph Neural Networks for Materials Science

- Hypergraph representations
- Importance of Equivariance

Overview

Graph Neural Networks for Materials Science

- Hypergraph representations
- Importance of Equivariance
- $O(3)$ Equivariance & Tensor Predictions

Overview

Graph Neural Networks for Materials Science

- Hypergraph representations
- Importance of Equivariance
- $O(3)$ Equivariance & Tensor Predictions
- Why we should use material symmetry groups!

Why Neural Networks?

Why Neural Networks?

Physics assumes there exists a map between configurations and properties of physical systems:

$$\{\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{\vec{r}_i'\}$$

Why Neural Networks?

Physics assumes there exists a map between configurations and properties of physical systems:

$$\{\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{\vec{r}_i'\}$$

Neural network techniques assume maps exist between abstract input and output spaces:

$$X \xrightarrow[\text{Neural Network}]{} Y$$

Why Neural Networks?

Physics assumes there exists a map between configurations and properties of physical systems:

$$\{\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{\vec{r}_i'\}$$

Neural network techniques assume maps exist between abstract input and output spaces:

$$X \xrightarrow[\text{Neural Network}]{} Y$$

These may be expressed in a basis of adequately large neural networks [1].

Why Neural Networks?

Physics assumes there exists a map between configurations and properties of physical systems:

$$\{\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{\vec{r}_i'\}$$

Neural network techniques assume maps exist between abstract input and output spaces:

$$X \xrightarrow[\text{Neural Network}]{} Y$$

These may be expressed in a basis of adequately large neural networks [1].

But how do we apply this to material systems?

Machine Learning on Crystal Systems

To perform predictive tasks for material systems, we need two things:

Machine Learning on Crystal Systems

To perform predictive tasks for material systems, we need two things:

1. A way to represent the system mathematically

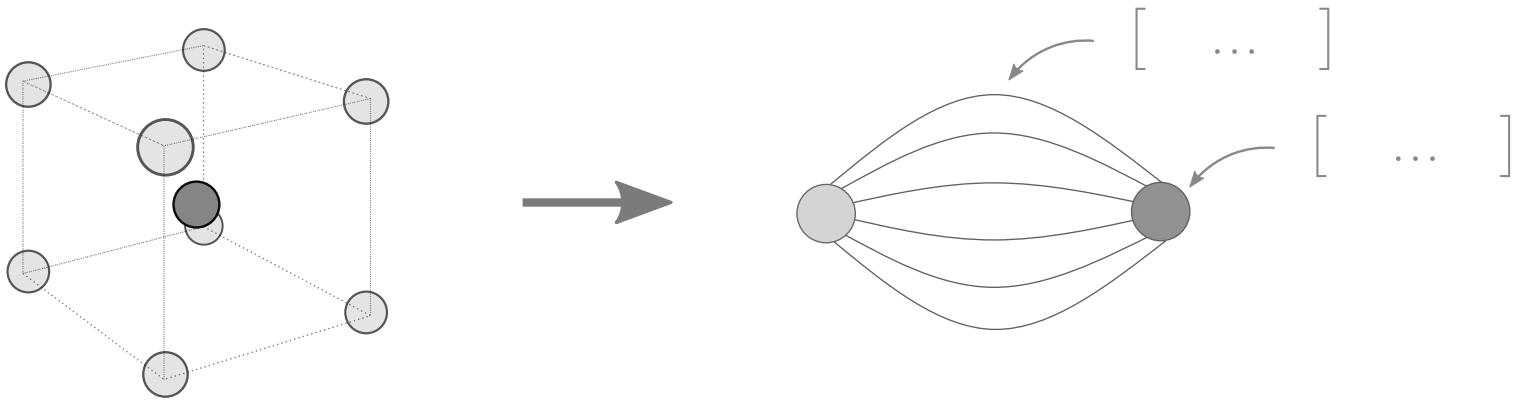
Machine Learning on Crystal Systems

To perform predictive tasks for material systems, we need two things:

1. A way to represent the system mathematically
2. A trainable model that acts on this representation

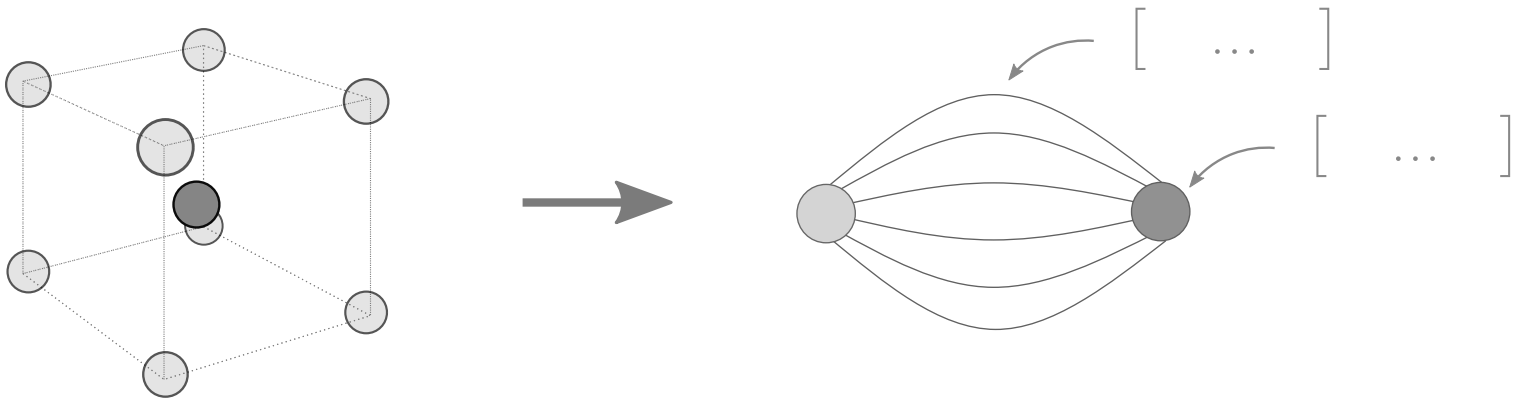
Usual Crystal Graph Construction (a la CGCNN)

- Common techniques represent crystalline systems as graphs (nodes and edges) [2, 3, 4]



Usual Crystal Graph Construction (a la CGCNN)

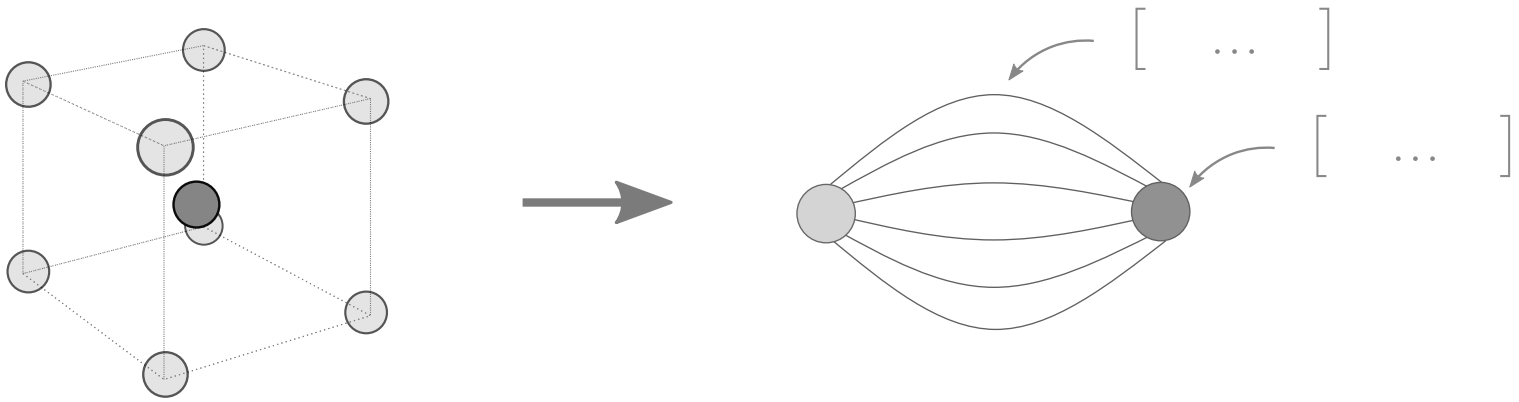
- Common techniques represent crystalline systems as graphs (nodes and edges) [2, 3, 4]



- Associate atomic features with nodes and geometric features with edges

Usual Crystal Graph Construction (a la CGCNN)

- Common techniques represent crystalline systems as graphs (nodes and edges) [2, 3, 4]

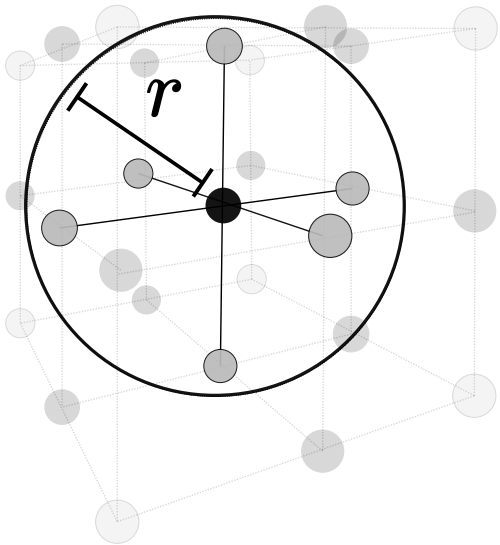


- Associate atomic features with nodes and geometric features with edges

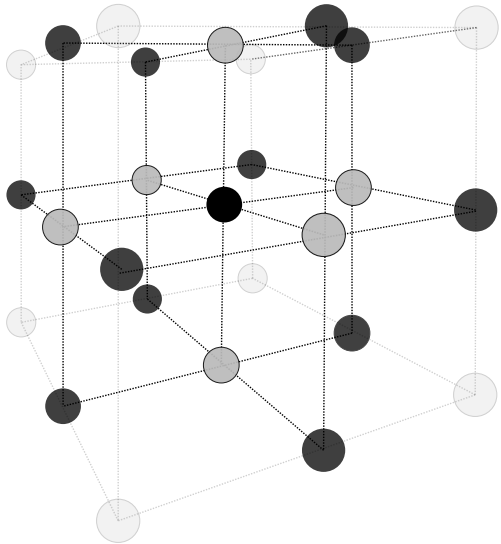
How do we construct these?

Crystal Graphs cont. I

- Edges are determined by distance cutoff (8 Ang.) and maximum number of neighbors (12)



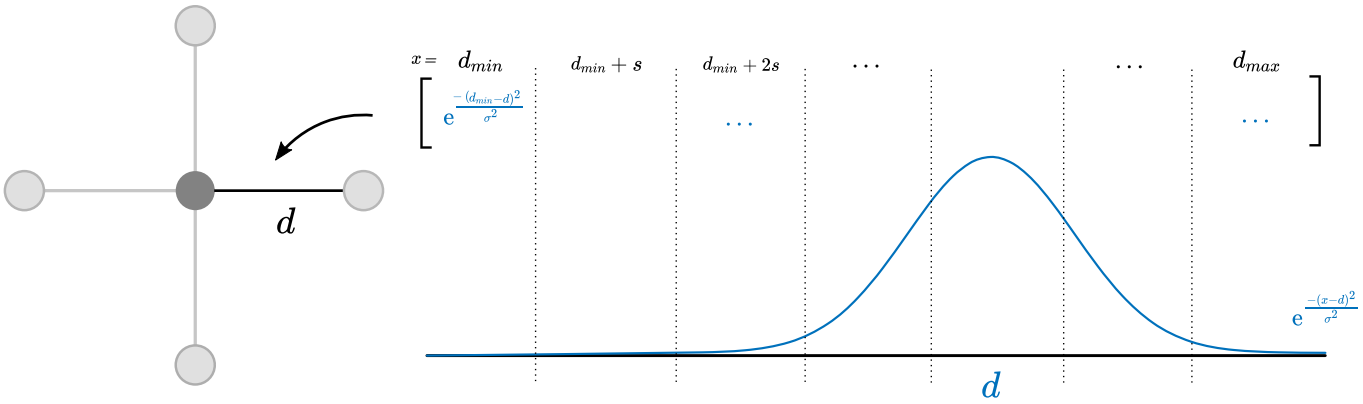
(within) Max radius r



N nearest neighbors

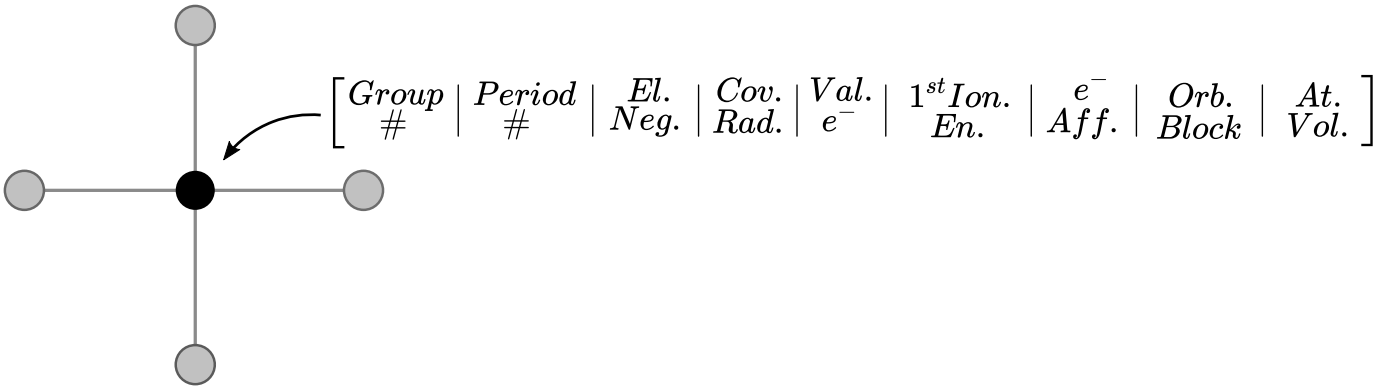
Crystal Graphs cont. II

- Edge attributes e then are a Gaussian distance expansion



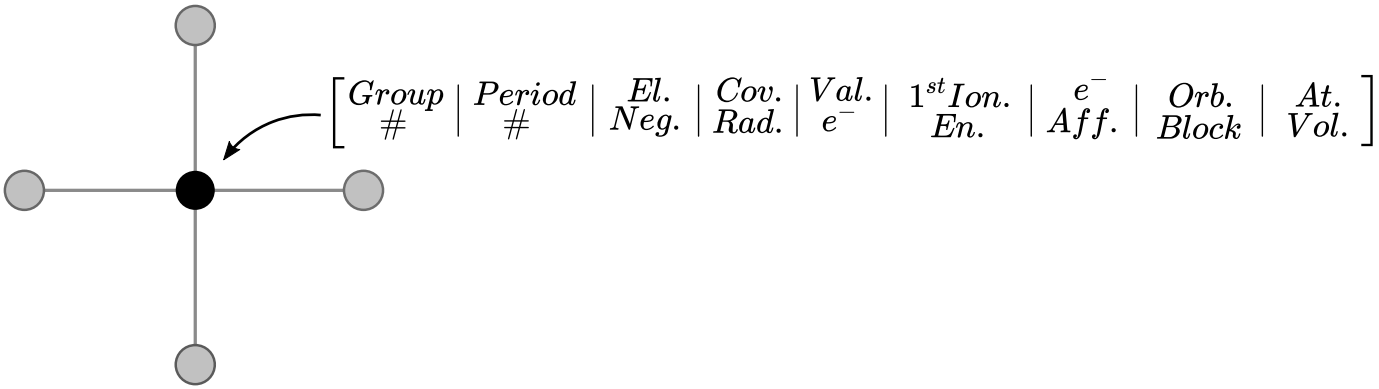
Crystal Graphs cont. III

- Node features n are encoded physical properties



Crystal Graphs cont. III

- Node features n are encoded physical properties



What do we do with these things?

Message Passing on Graphs

Update with GNN or message passing network [5]:

$$m_i^{t+1} = \sum_{n_j \in \mathcal{N}(i)} M_t(n_i^t, e_{ij}, n_j^t)$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

Message Passing on Graphs

Update with GNN or message passing network [5]:

$$m_i^{t+1} = \sum_{n_j \in \mathcal{N}(i)} M_t(n_i^t, e_{ij}, n_j^t)$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

Then, read out prediction from updated node features.

Continuous Filters

Atomic interactions are very sensitive to distance!

Continuous Filters

Atomic interactions are very sensitive to distance!

- Continuous-resolution convolutional filters [3]

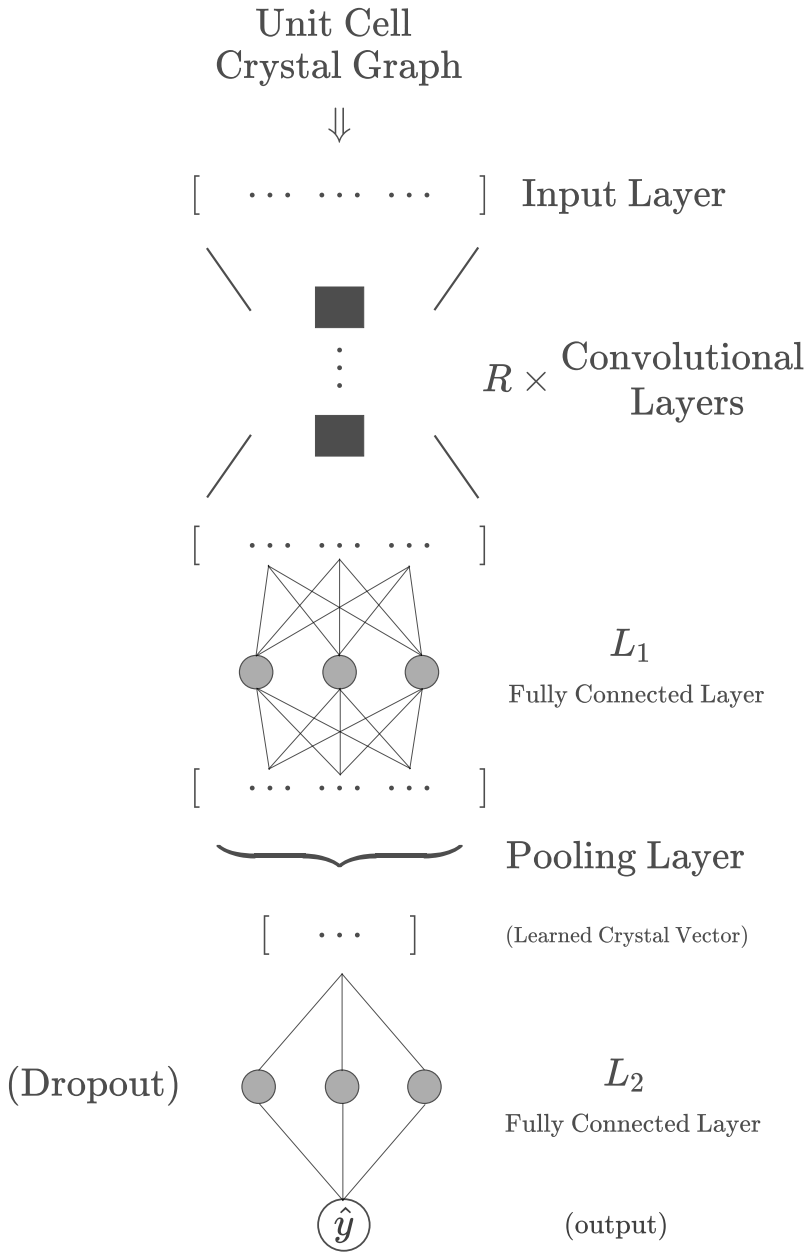
Continuous Filters

Atomic interactions are very sensitive to distance!

- Continuous-resolution convolutional filters [3]
- CGConv [2]:

$$
\begin{aligned}
n_i^{t+1} &= \sum_{b_j} f(n_i^t, b_j, \mathrm{AGG}(\{n_j^t \in b_j\})) \\
&= n_i^t + \mathrm{BN}\Bigg[\sum_{b_j} \sigma\big(W_c \cdot [n_j \oplus e_{ij} \oplus n_j]\big) \\
&\qquad \cdot S^+(W_f \cdot [n_j \oplus e_{ij} \oplus n_j])\Bigg]
\end{aligned}
$$

Graph Neural Network

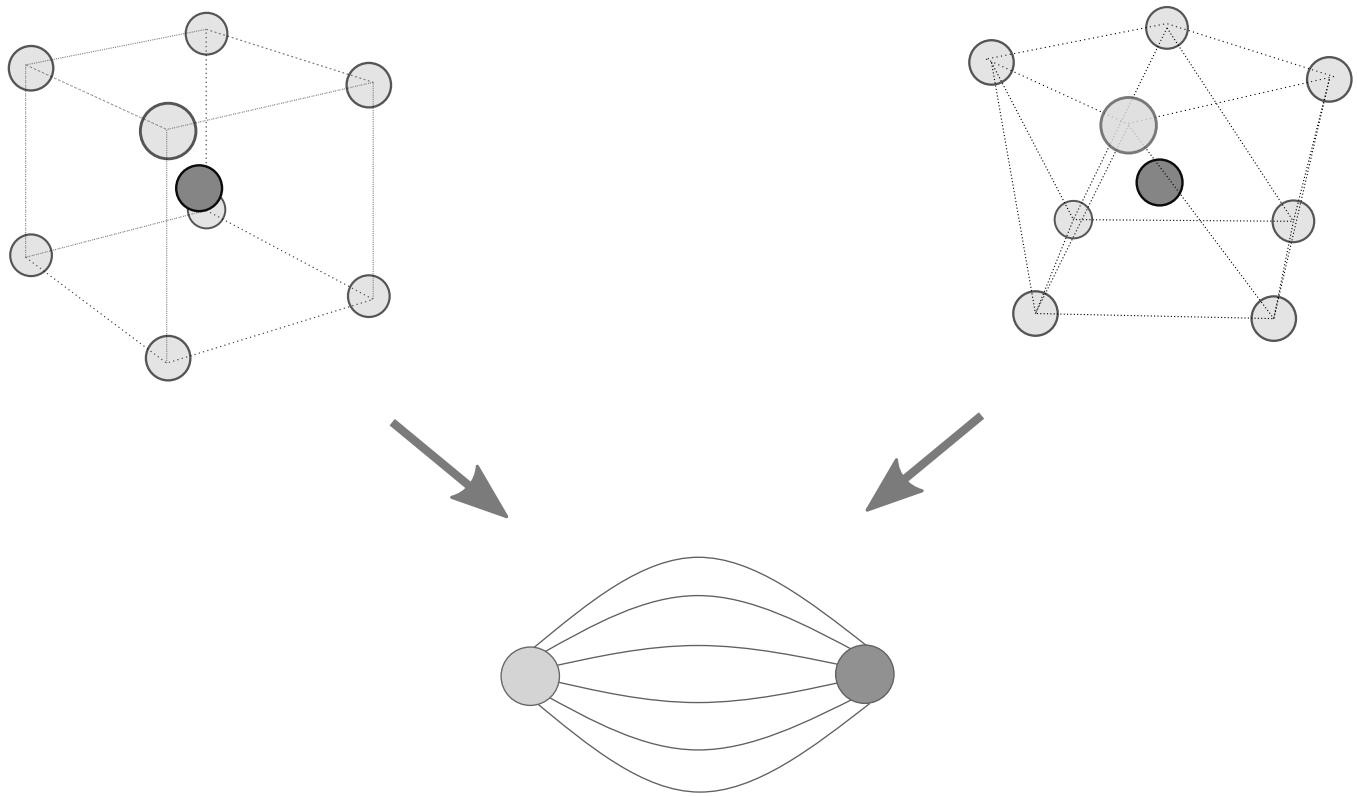


Graph Limitations

Problem: Our underlying representation encodes only distances between atoms!

Graph Limitations

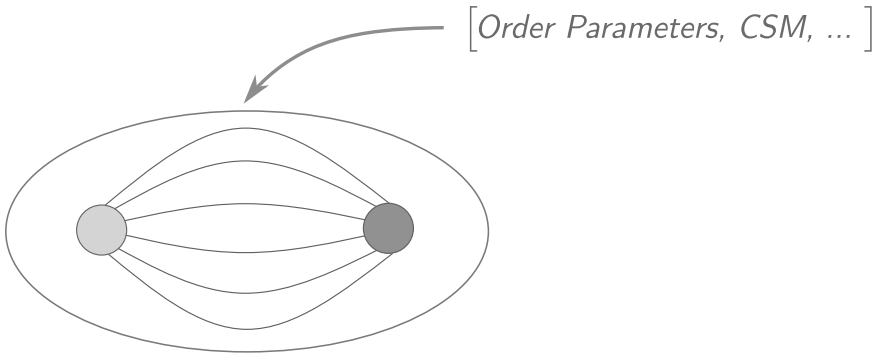
Problem: Our underlying representation encodes only distances between atoms!



These two have the same representations!

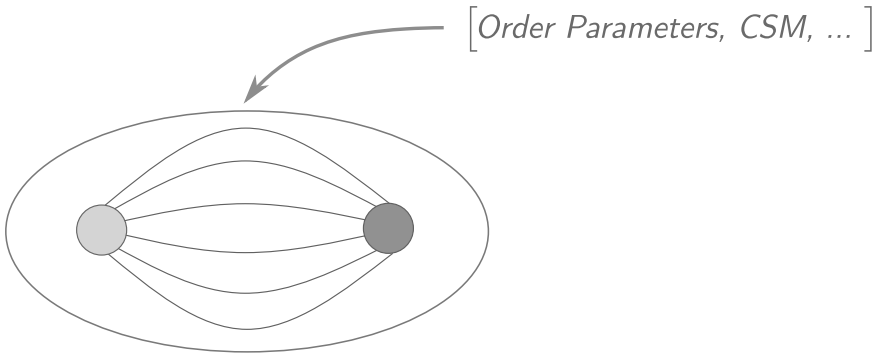
(One) Solution: Hypergraphs!

Hypergraphs allow us to have edges containing more than (or less than) two nodes.



(One) Solution: Hypergraphs!

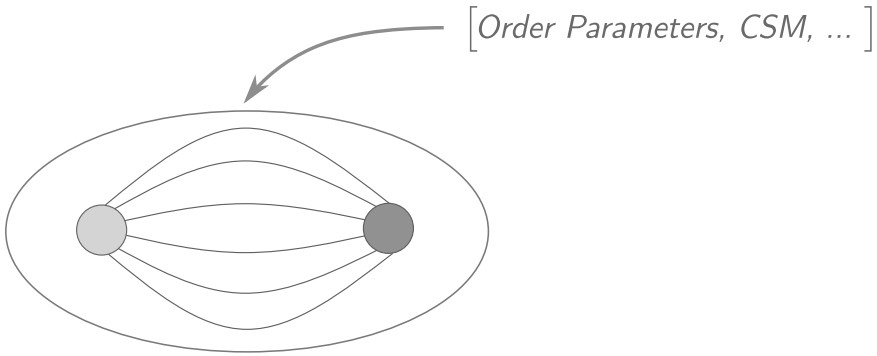
Hypergraphs allow us to have edges containing more than (or less than) two nodes.



• Gives a natural way to encode features with higher order structure

(One) Solution: Hypergraphs!

Hypergraphs allow us to have edges containing more than (or less than) two nodes.

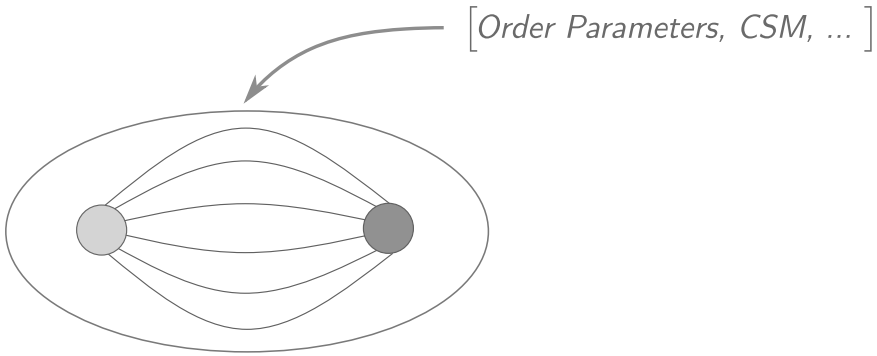


• Gives a natural way to encode features with higher order structure

• Hypergraph nodes still represent atoms of underlying crystal structure.

(One) Solution: Hypergraphs!

Hypergraphs allow us to have edges containing more than (or less than) two nodes.



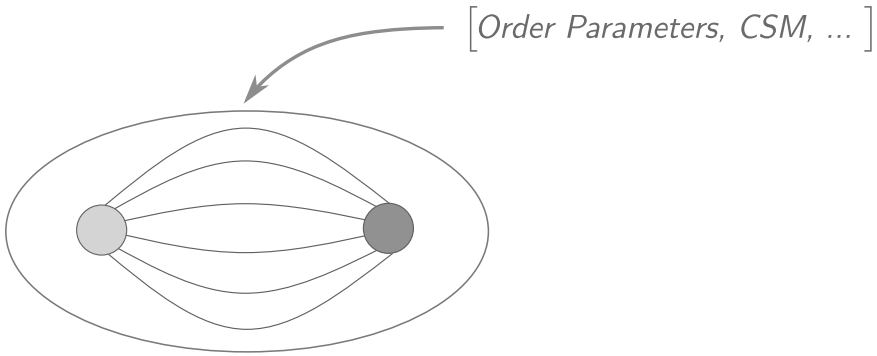
• Gives a natural way to encode features with higher order structure

• Hypergraph nodes still represent atoms of underlying crystal structure.

• Treat all different order structures on equal footing: each has a corresponding hyperedge with a feature

(One) Solution: Hypergraphs!

Hypergraphs allow us to have edges containing more than (or less than) two nodes.



• Gives a natural way to encode features with higher order structure

• Hypergraph nodes still represent atoms of underlying crystal structure.

• Treat all different order structures on equal footing: each has a corresponding hyperedge with a feature

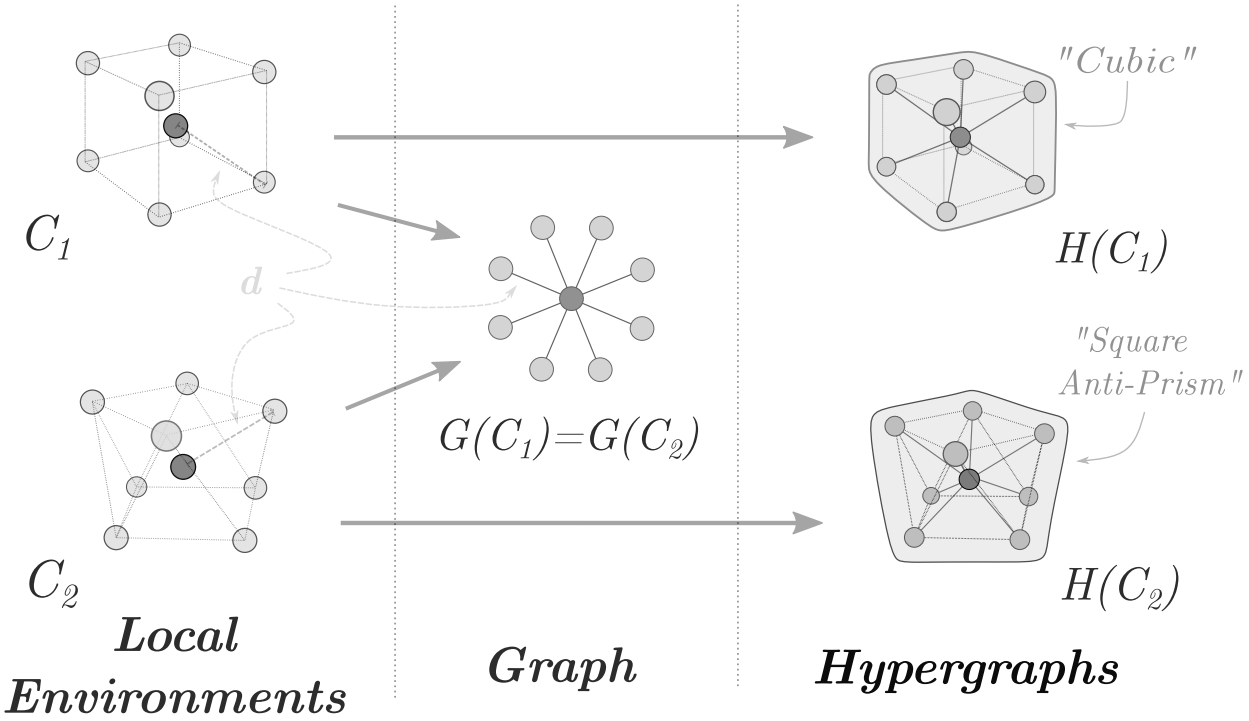
But what do we care about?

Coordination Environments as Hyperedges

Considering the previous problem; encode coordination environment in larger hyperedges.

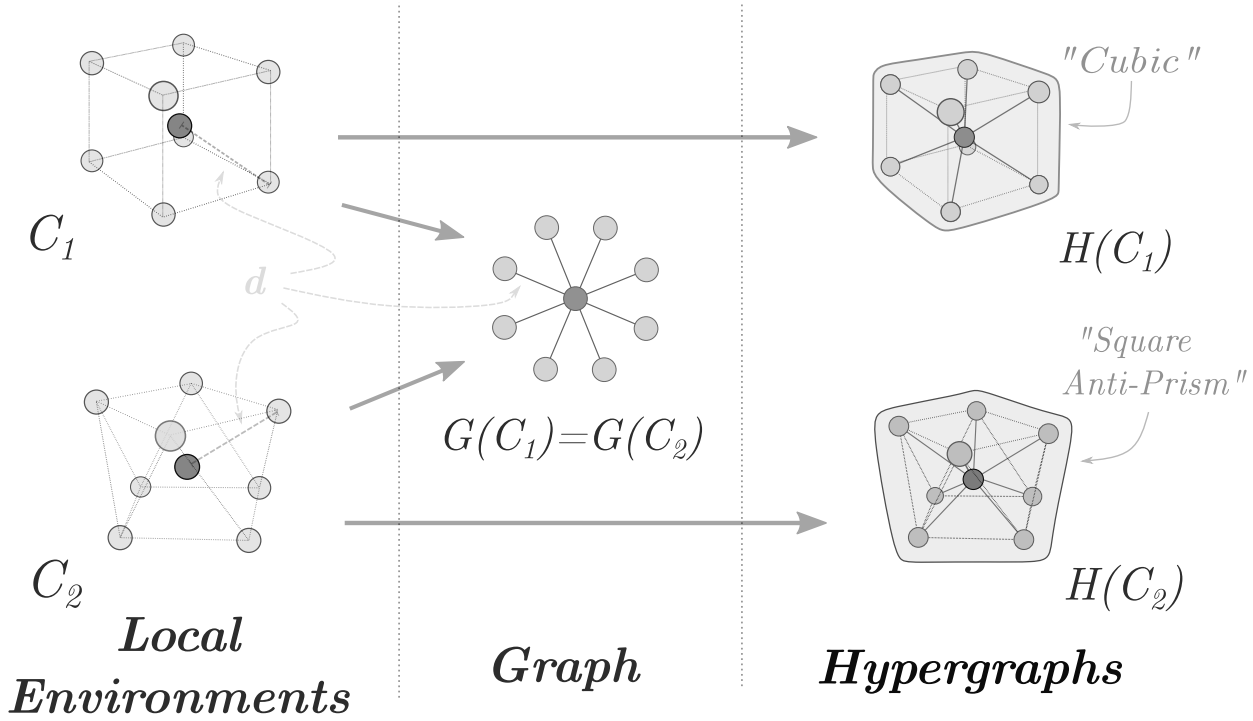
Coordination Environments as Hyperedges

Considering the previous problem; encode coordination environment in larger hyperedges.



Coordination Environments as Hyperedges

Considering the previous problem; encode coordination environment in larger hyperedges.



• Geometrical information can be encoded quantitatively with continuous symmetry measures, local structure order parameters, etc.

Extending Message Passing to Hypergraphs

Now, we need a suitable convolutional structure that applies to hypergraphs...

$$m_i^{t+1} = \sum_{h_j \ni x_i} M_t(n_i^t, \underbrace{h_j^t, \{n_w^t | n_w \in h_j\}}_{e_{ij}, n_j})$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

Extending Message Passing to Hypergraphs

Now, we need a suitable convolutional structure that applies to hypergraphs...

$$m_i^{t+1} = \sum_{h_j \ni x_i} M_t(n_i^t, \underbrace{h_j^t, \{n_w^t | n_w \in h_j\}}_{e_{ij}, n_j})$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

- Neighborhoods of nodes relevant to each message are now a set

Extending Message Passing to Hypergraphs

Now, we need a suitable convolutional structure that applies to hypergraphs...

$$m_i^{t+1} = \sum_{h_j \ni x_i} M_t(n_i^t, \underbrace{h_j^t, \{n_w^t | n_w \in h_j\}}_{e_{ij}, n_j})$$

$$n_i^{t+1} = U_t(n_i^t, m_i^{t+1})$$

- Neighborhoods of nodes relevant to each message are now a set

What can we do?

Hypergraph Convolution

Aggregate the node features!

$$n_i \rightarrow n_i + \sum_{e_{ij}} M_b(n_i \oplus e_{ij} \oplus n_j)$$
$$+ \sum_{h_j} M_m(n_i \oplus h_j \oplus \text{AGG}(\{n_w^t | n_w \in h_j\}))$$

Hypergraph Convolution

Aggregate the node features!

$$n_i \rightarrow n_i + \sum_{e_{ij}} M_b(n_i \oplus e_{ij} \oplus n_j)$$
$$+ \sum_{h_j} M_m(n_i \oplus h_j \oplus \mathrm{AGG}(\{n_w^t | n_w \in h_j\}))$$

- AGG can be maximum, minimum, mean, standard deviation

Hypergraph Convolution

Aggregate the node features!

$$n_i \rightarrow n_i + \sum_{e_{ij}} M_b(n_i \oplus e_{ij} \oplus n_j)$$
$$+ \sum_{h_j} M_m(n_i \oplus h_j \oplus \mathrm{AGG}(\{n_w^t | n_w \in h_j\}))$$

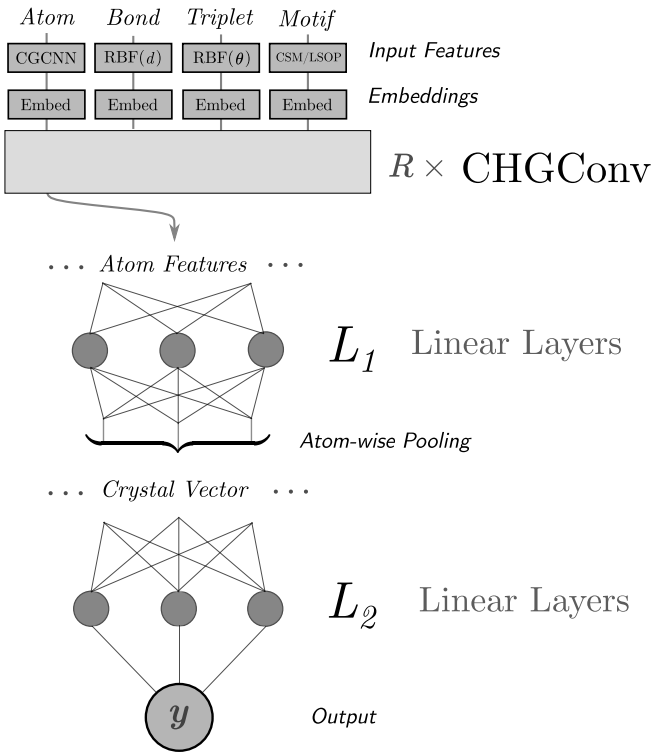
- AGG can be maximum, minimum, mean, standard deviation
- Can also be learnable mix of above with attention

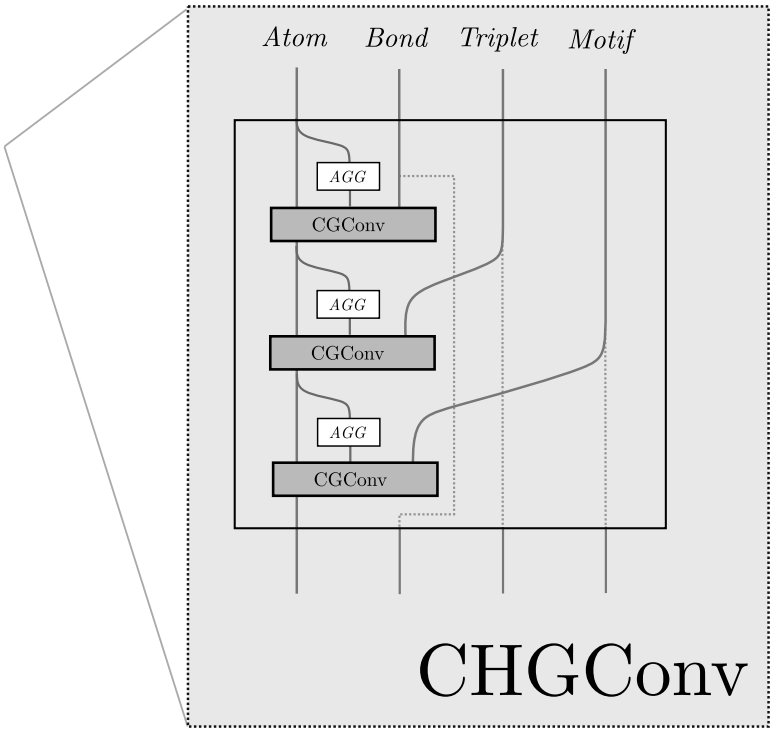
CHGConv

A generalization of CGConv with hyperedge aggregation:

$$
\begin{aligned}
n_i^{t+1} &= n_i^t + \sum_{b_j} f(n_i^t, b_j, \mathrm{AGG}(\{n_j^t \in b_j\})) \\
&= n_i^t + \mathrm{BN}\Bigg[\sum_{b_j} \sigma\big(W_c \cdot [n_j \oplus b_j \oplus \mathrm{AGG}(\{n_j^t \in b_j\}]\big) \\
&\qquad \cdot S^+(W_f \cdot (n_j \oplus b_j \oplus \mathrm{AGG}(\{n_j^t \in b_j\})))\Bigg]
\end{aligned}
$$

CHGCNN Schematic





CHGCNN Results

Hyperedge Types	E_g MAE (eV)	G_{vrh} MAE (Log_{10}GPa)	K_{vrh} MAE (Log_{10}GPa)
Bond-Only	0.257	0.0826	0.0688
Bond & Motif	0.232	0.0813	0.0623

CHGCNN Results

Hyperedge Types	E_g MAE (eV)	G_{vrh} MAE (Log_{10}GPa)	K_{vrh} MAE (Log_{10}GPa)
Bond-Only	0.257	0.0826	0.0688
Bond & Motif	0.232	0.0813	0.0623

- Allows us to effectively capture higher-order geometric features, in an *invariant* manner!

CHGCNN Results

Hyperedge Types	E_g MAE (eV)	G_{vrh} MAE (Log_{10}GPa)	K_{vrh} MAE (Log_{10}GPa)
Bond-Only	0.257	0.0826	0.0688
Bond & Motif	0.232	0.0813	0.0623

- Allows us to effectively capture higher-order geometric features, in an *invariant* manner!

But what if we care about about coordinate-system dependent quantities?

Equivariant Functions

A function $f : X \to Y$ is *equivariant* if it 'commutes' with a group:

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

Equivariant Functions

A function $f : X \to Y$ is *equivariant* if it 'commutes' with a group:

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

- X, Y are vector spaces

Equivariant Functions

A function $f : X \to Y$ is *equivariant* if it 'commutes' with a group:

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

- X, Y are vector spaces
- $\mathcal{D}$ are group representations of elements g

Equivariant Functions

A function $f : X \to Y$ is *equivariant* if it 'commutes' with a group:

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

- X, Y are vector spaces
- $\mathcal{D}$ are group representations of elements g

Note: Equivariance is always with respect to some group!

Why Equivariant Functions?

Physical processes respect coordinate system rotations R:

$$\{R\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{R\vec{r}'_i\}$$

Why Equivariant Functions?

Physical processes respect coordinate system rotations R:

$$\{R\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{R\vec{r}'_i\}$$

In general, neural networks do not:

$$RX \xrightarrow[\text{Neural Network}]{} \ ?$$

Why Equivariant Functions?

Physical processes respect coordinate system rotations R:

$$\{R\vec{r}_i\} \xrightarrow[\text{Nature}]{} \{R\vec{r}'_i\}$$

In general, neural networks do not:

$$RX \xrightarrow[\text{Neural Network}]{} ?$$

So what do we want?

Equivariant Networks

We want equivariant networks for physics!

$$RX \xrightarrow[\text{Neural Network}]{\text{Equivariant}} RY$$

Equivariant Networks

We want equivariant networks for physics!

$$RX \xrightarrow[\text{Neural Network}]{\text{Equivariant}} RY$$

In equivariant networks, we consider feature vectors in basis of representation space:

$$v_i^{\alpha}$$

Equivariant Networks

We want equivariant networks for physics!

$$RX \xrightarrow[\text{Neural Network}]{\text{Equivariant}} RY$$

In equivariant networks, we consider feature vectors in basis of representation space:

$$v_i^{\alpha}$$

Index α denotes irrep, and i the dimension of irrep space α.

Equivariant Networks (cont.)

Compositions of equivariant functions are equivariant [6].

Equivariant Networks (cont.)

Compositions of equivariant functions are equivariant [6].We consider building blocks:

- Non-linear functions along channels (for trivial irrep $\alpha = 1$):

$$v'_{nc} = f(v_{nc} + b_{nc})$$

Equivariant Networks (cont.)

Compositions of equivariant functions are equivariant [6].We consider building blocks:

- Non-linear functions along channels (for trivial irrep $\alpha = 1$):

$$v'_{nc} = f(v_{nc} + b_{nc})$$

- Group correlation with filters [6]:

$$[F \star v](g) = \sum_{h \in G} \sum_{c} F_c(h) v_c(g^{-1}h)$$

Equivariant Networks (cont.)

Compositions of equivariant functions are equivariant [6].We consider building blocks:

- Non-linear functions along channels (for trivial irrep $\alpha = 1$):

$$v'_{nc} = f(v_{nc} + b_{nc})$$

- Group correlation with filters [6]:

$$[F \star v](g) = \sum_{h \in G} \sum_{c} F_c(h) v_c(g^{-1}h)$$

- Tensor products with filters [7, 8]:

$$(v'_{nc})^{\gamma}_{n} = \sum_{ij} U^{\gamma n}_{\alpha i \beta j} \left(F^{\beta}_{j} \otimes V^{\alpha}_{i}\right)$$

Equivariant Networks (cont.)

Compositions of equivariant functions are equivariant [6].We consider building blocks:

- Non-linear functions along channels (for trivial irrep $\alpha = 1$):

$$v'_{nc} = f(v_{nc} + b_{nc})$$

- Group correlation with filters [6]:

$$[F \star v](g) = \sum_{h \in G} \sum_{c} F_c(h) v_c(g^{-1}h)$$

- Tensor products with filters [7, 8]:

$$(v'_{nc})^{\gamma}_{n} = \sum_{ij} U^{\gamma n}_{\alpha i \beta j} \left(F^{\beta}_{j} \otimes V^{\alpha}_{i}\right)$$

- Pooling over nodes or channels:

$$v_c = \sum_{n} v_{nc}$$

$SO(3)$ Properties

Most modern equivariant networks are specifically $SO(3)$-Equivariant.

$SO(3)$ Properties

Most modern equivariant networks are specifically $SO(3)$-Equivariant.

- Inifinite order $\rightarrow$ infinite irreps

$SO(3)$ Properties

Most modern equivariant networks are specifically $SO(3)$-Equivariant.

- Inifinite order $\rightarrow$ infinite irreps
- Irreps are Wigner $\mathcal{D}^{\ell}$ matrices

$SO(3)$ Properties

Most modern equivariant networks are specifically $SO(3)$-Equivariant.

- Inifinite order $\rightarrow$ infinite irreps
- Irreps are Wigner $\mathcal{D}^{\ell}$ matrices
- Natural basis set Y_m^{ℓ} of dimension $d_{\ell} = 2\ell + 1$

$SO(3)$ Properties

Most modern equivariant networks are specifically $SO(3)$-Equivariant.

- Inifinite order $\rightarrow$ infinite irreps
- Irreps are Wigner $\mathcal{D}^{\ell}$ matrices
- Natural basis set Y_m^{ℓ} of dimension $d_{\ell} = 2\ell + 1$

Many models are further referred to as $E(3)$-equivariant [8] for the Euclidean group (add parity and translation invariance).

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L
- $c_{\ell_f m_f \ell_i m_i}^{\ell_o m_o}$: Clebsch-Gordan coefficients

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L
- $c_{\ell_f m_f \ell_i m_i}^{\ell_o m_o}$: Clebsch-Gordan coefficients
- F_c^L: Filter function (trainable)

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L
- $c_{\ell_f m_f \ell_i m_i}^{\ell_o m_o}$: Clebsch-Gordan coefficients
- F_c^L: Filter function (trainable)
- r_{nb}: Radius between nodes n and neighbor $b \in \mathcal{N}(n)$

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L
- $c_{\ell_f m_f \ell_i m_i}^{\ell_o m_o}$: Clebsch-Gordan coefficients
- F_c^L: Filter function (trainable)
- r_{nb}: Radius between nodes n and neighbor $b \in \mathcal{N}(n)$

F is generally a neural network

Tensor Field Networks

Tensor field networks [9] use SO(3) equivariant convolution:

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell} = \left(v_{nc}^{L}\right)_{m}^{\ell} + \sum_{b \in \mathcal{N}(n)} \sum_{\ell_f m_f, \ell_i m_i}^{\ell_{\max} m_{\max}} c_{\ell_f m_f \ell_i m_i}^{\ell m} \left(F_c^L(r_{nb})\right)_{m_f}^{\ell_f} \left(v_{bc}^{L}\right)_{m_i}^{\ell_i}$$

- v_{nc}^{L}: Node feature of node n, channel c, layer L
- $c_{\ell_f m_f \ell_i m_i}^{\ell_o m_o}$: Clebsch-Gordan coefficients
- F_c^L: Filter function (trainable)
- r_{nb}: Radius between nodes n and neighbor $b \in \mathcal{N}(n)$

F is generally a neural network; r is often expanded with some radial basis function (RBF): Gaussian, Bessel, etc.

$SO(3)$-Equivariant Features

Remember that node (and edge) features now have additional indices ℓ and m!

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell}$$

$SO(3)$-Equivariant Features

Remember that node (and edge) features now have additional indices ℓ and m!

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell}$$

This allows us to keep track of which components transform like which irreps of $SO(3)$.

$SO(3)$-Equivariant Features

Remember that node (and edge) features now have additional indices ℓ and m!

$$\left(v_{nc}^{L+1}\right)_{m}^{\ell}$$

This allows us to keep track of which components transform like which irreps of $SO(3)$.

But what can we do with this?

J_z Basis

Recall $\ell = 1$ spherical harmonics (with Racah normalization):

$$Y_1^{+1} = -\frac{1}{\sqrt{2}}(x + iy) = \frac{1}{\sqrt{2}} \sin\phi e^{i\theta}$$
$$Y_1^0 \quad = z = \cos\phi$$
$$Y_1^{-1} = -\frac{1}{\sqrt{2}}(x - iy) = \frac{1}{\sqrt{2}} \sin\phi e^{-i\theta}$$

J_z Basis

Recall $\ell = 1$ spherical harmonics (with Racah normalization):

$$
\begin{aligned}
Y_1^{+1} &= -\frac{1}{\sqrt{2}}(x + iy) = \frac{1}{\sqrt{2}} \sin\phi e^{i\theta} \\
Y_1^{0} &= z = \cos\phi \\
Y_1^{-1} &= -\frac{1}{\sqrt{2}}(x - iy) = \frac{1}{\sqrt{2}} \sin\phi e^{-i\theta}
\end{aligned}
$$

Define J_z basis [10, 11]:

$$
\begin{bmatrix} a_+ \\ a_0 \\ a_- \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ -\frac{1}{\sqrt{2}} & +\frac{i}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}
$$

J_z Basis

Recall $\ell = 1$ spherical harmonics (with Racah normalization):

$$Y_1^{+1} = -\frac{1}{\sqrt{2}}(x + iy) = \frac{1}{\sqrt{2}} \sin\phi e^{i\theta}$$
$$Y_1^0 \quad = z = \cos\phi$$
$$Y_1^{-1} = -\frac{1}{\sqrt{2}}(x - iy) = \frac{1}{\sqrt{2}} \sin\phi e^{-i\theta}$$

Define J_z basis [10, 11]:

$$\begin{bmatrix} a_+ \\ a_0 \\ a_- \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ -\frac{1}{\sqrt{2}} & +\frac{i}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

so that $\vec{n} = a_+ \hat{Y}_1^1 + a_0 \hat{Y}_1^0 + a_- \hat{Y}_1^{-1}$

Clebsch-Gordon Expansion

Build larger spherical harmonic tensors with CG expansion:

$$Y_{\ell_1}^{m_1} \otimes Y_{\ell_2}^{m_2} = \sum_{L=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M=-L}^{L} C_{\ell_1 0 \ell_2 0}^{L0} C_{\ell_1 m_1 \ell_2 m_2}^{LM} Y_L^M$$

Clebsch-Gordon Expansion

Build larger spherical harmonic tensors with CG expansion:

$$Y_{\ell_1}^{m_1} \otimes Y_{\ell_2}^{m_2} = \sum_{L=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M=-L}^{L} C_{\ell_1 0 \ell_2 0}^{L0} C_{\ell_1 m_1 \ell_2 m_2}^{LM} Y_L^M$$

[12] where Y_L represents a $2L+1$ dimensional symmetric tensor space of rank L.

Clebsch-Gordon Expansion

Build larger spherical harmonic tensors with CG expansion:

$$Y_{\ell_1}^{m_1} \otimes Y_{\ell_2}^{m_2} = \sum_{L=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M=-L}^{L} C_{\ell_1 0 \ell_2 0}^{L0} C_{\ell_1 m_1 \ell_2 m_2}^{LM} Y_L^M$$

[12] where Y_L represents a $2L+1$ dimensional symmetric tensor space of rank L.

• Gives a relation between symmetric tensor's J_z basis components and other spherical harmonic tensors.

$$T^{(n)} = a_{\underbrace{\alpha\beta\ldots}_{n}} (Y_1^\alpha \otimes Y_1^\beta \otimes \ldots) \quad \Rightarrow \quad y_\ell^m Y_L^M$$

Clebsch-Gordon Expansion

Build larger spherical harmonic tensors with CG expansion:

$$Y_{\ell_1}^{m_1} \otimes Y_{\ell_2}^{m_2} = \sum_{L=|\ell_1-\ell_2|}^{\ell_1+\ell_2} \sum_{M=-L}^{L} C_{\ell_1 0 \ell_2 0}^{L0} C_{\ell_1 m_1 \ell_2 m_2}^{LM} Y_L^M$$

[12] where Y_L represents a $2L+1$ dimensional symmetric tensor space of rank L.

• Gives a relation between symmetric tensor's J_z basis components and other spherical harmonic tensors.

$$T^{(n)} = a_{\underbrace{\alpha\beta\ldots}_{n}} (Y_1^\alpha \otimes Y_1^\beta \otimes \ldots) \quad \Rightarrow \quad y_\ell^m Y_L^M$$

But, what about asymmetric tensors?

$SO(3)$ Invariant Tensor Subspaces

- An arbitrary tensor T transforms as:

$$T_{x_1 x_2 \ldots x_n} \rightarrow T_{x_1' x_2' \ldots x_n'} = R^{x_1}_{x_1'} R^{x_2}_{x_2'} R^{x_3}_{x_3'} T_{x_1 x_2 \ldots x_n},$$

$SO(3)$ Invariant Tensor Subspaces

- An arbitrary tensor T transforms as:

$$T_{x_1 x_2 \ldots x_n} \rightarrow T_{x_1' x_2' \ldots x_n'} = R^{x_1}_{x_1'} R^{x_2}_{x_2'} R^{x_3}_{x_3'} T_{x_1 x_2 \ldots x_n},$$

- T may be broken into a set of irreducible (but not necessarily unique) symmetric, $SO(3)$ invariant subtensors:

$$\{h^{(\ell)}\} \rightarrow \{h'^{(\ell)}\} = \{\mathcal{D}^{\ell}(R) h^{(\ell)}\}$$

$SO(3)$ Invariant Tensor Subspaces

- An arbitrary tensor T transforms as:

$$T_{x_1 x_2 \ldots x_n} \rightarrow T_{x_1' x_2' \ldots x_n'} = R^{x_1}_{x_1'} R^{x_2}_{x_2'} R^{x_3}_{x_3'} T_{x_1 x_2 \ldots x_n},$$

- T may be broken into a set of irreducible (but not necessarily unique) symmetric, $SO(3)$ invariant subtensors:

$$\{h^{(\ell)}\} \rightarrow \{h'^{(\ell)}\} = \{\mathcal{D}^{\ell}(R) h^{(\ell)}\}$$

- This decomposition can be constructed by consecutive decomposition with respect to GL and then O and SL

$$SO = (SL \cap O) \subset GL$$

Decomposition Example: Rank 2

Consider a rank-two tensor T_{ij}:

$$S_{ij} = \frac{1}{2}(T_{ij} + T_{ji})$$

$$A_{ij} = \frac{1}{2}(T_{ij} - T_{ji})$$

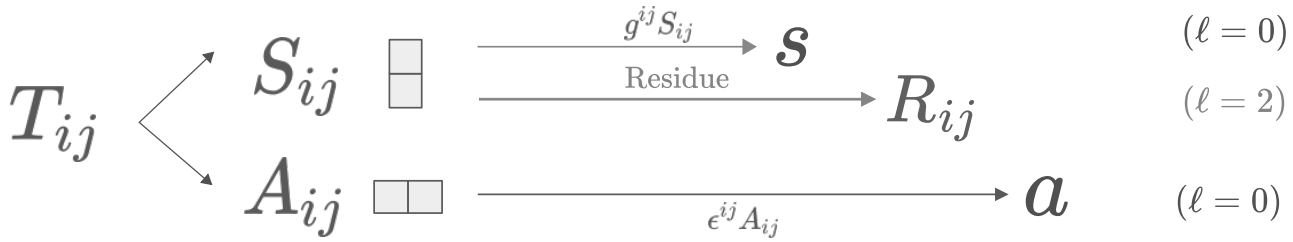
Decomposition Example: Rank 2

Consider a rank-two tensor T_{ij}:

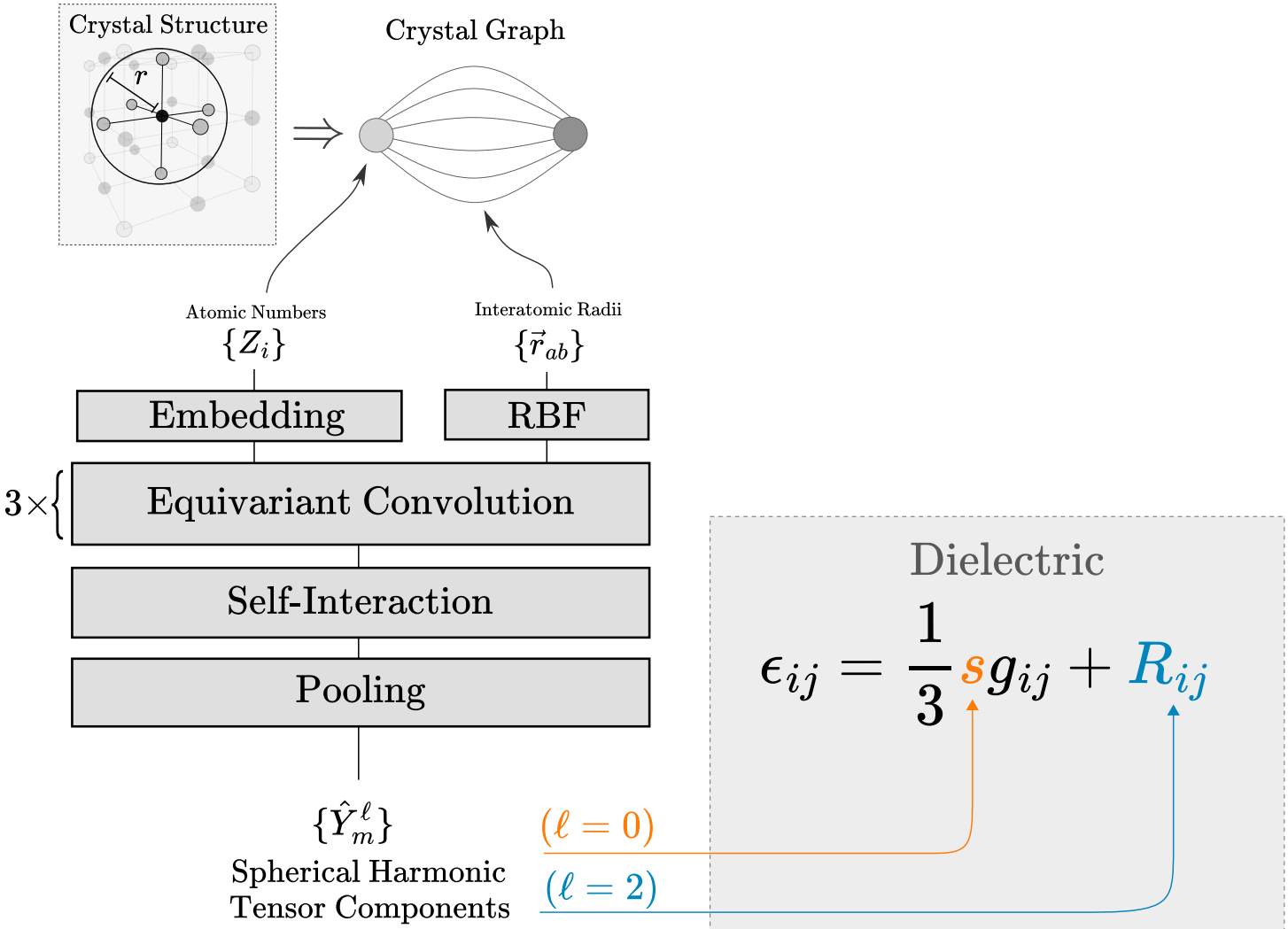
$$S_{ij} = \frac{1}{2}(T_{ij} + T_{ji})$$

$$A_{ij} = \frac{1}{2}(T_{ij} - T_{ji})$$

Then, contract with g and ε:



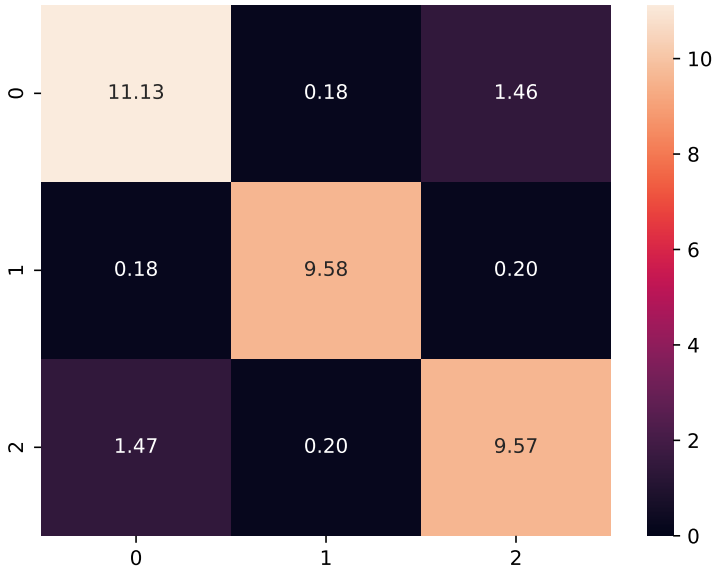
Model Architecture



Tensor Prediction: Examples

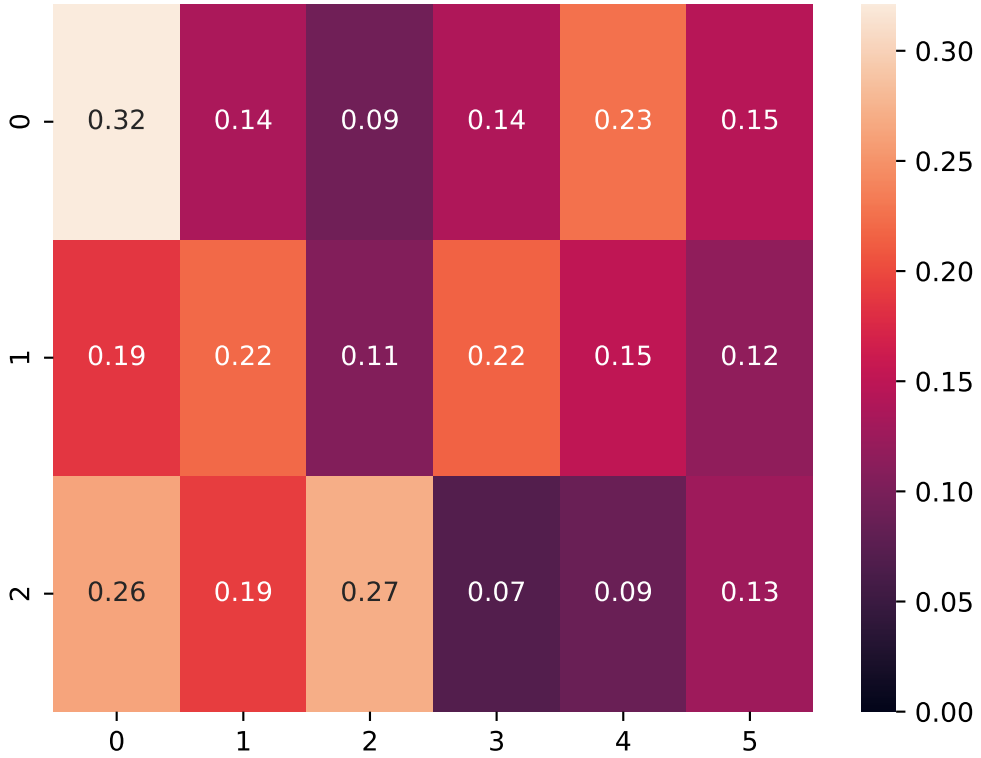
Applications to materials:

- Dielectric Tensor ϵ_{ij}



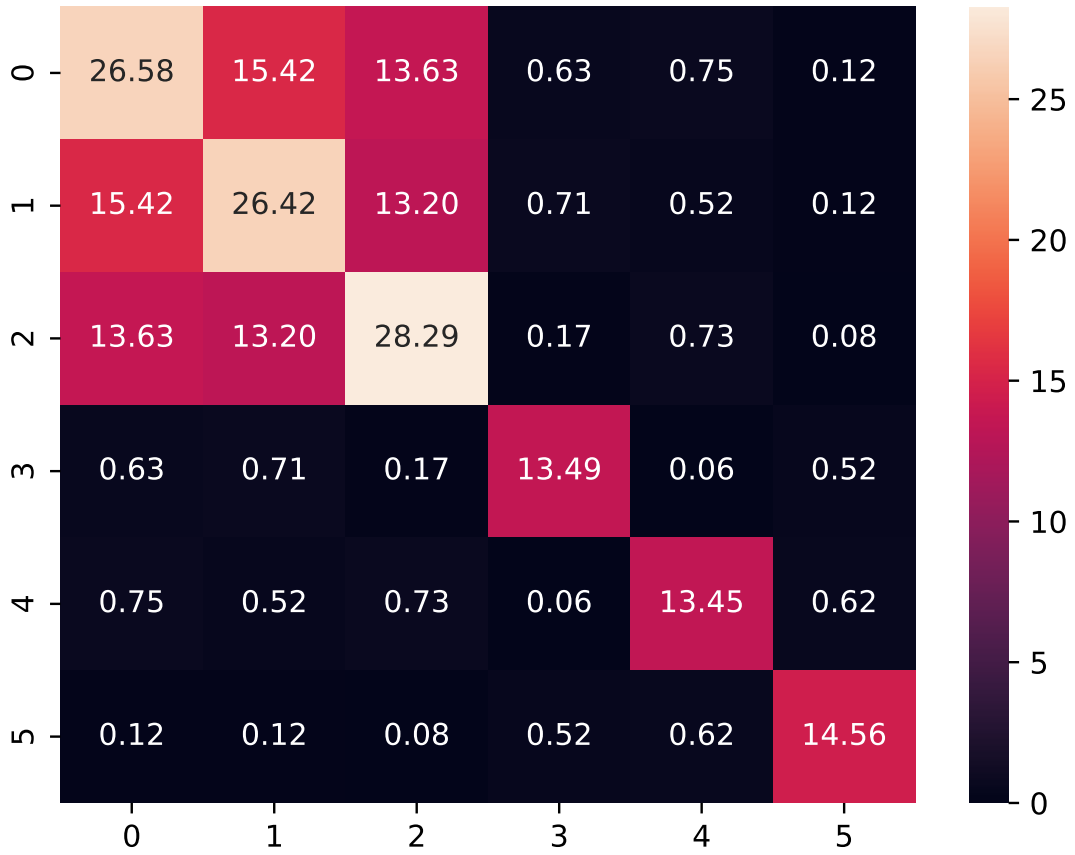
Tensor Prediction: Examples

- Piezoelectric Tensor d_{ijk}



Tensor Prediction: Examples

- Elasticity Tensor C_{ijkl}



Tensor Prediction with $SO(3)$-Networks

Target Components	MAE (Averaged Over Components) SEGNN	SEConv	SETransformer
Elastic *[GPa]*	8.139	7.689	7.941
Dielectric	4.82	4.702	4.718
Piezoelectric $[C/m^2]$	0.170	0.170	0.1714

Tensor Prediction with $SO(3)$-Networks

Target Components	MAE (Averaged Over Components) SEGNN	SEConv	SETransformer
Elastic *[GPa]*	8.139	7.689	7.941
Dielectric	4.82	4.702	4.718
Piezoelectric $[C/m^2]$	0.170	0.170	0.1714

- We can read off the output of $SO(3)$-networks as tensor components

Tensor Prediction with $SO(3)$-Networks

Target Components	MAE (Averaged Over Components) SEGNN	SEConv	SETransformer
Elastic *[GPa]*	8.139	7.689	7.941
Dielectric	4.82	4.702	4.718
Piezoelectric $[C/m^2]$	0.170	0.170	0.1714

- We can read off the output of $SO(3)$-networks as tensor components
- These naturally respect coordinate transformations acting on the input space!

Tensor Prediction with $SO(3)$-Networks

Target Components	MAE (Averaged Over Components) SEGNN	SEConv	SETransformer
Elastic *[GPa]*	8.139	7.689	7.941
Dielectric	4.82	4.702	4.718
Piezoelectric $[C/m^2]$	0.170	0.170	0.1714

- We can read off the output of $SO(3)$-networks as tensor components
- These naturally respect coordinate transformations acting on the input space!
- Can learn, but doesn't restrict to symmetry of crystals...

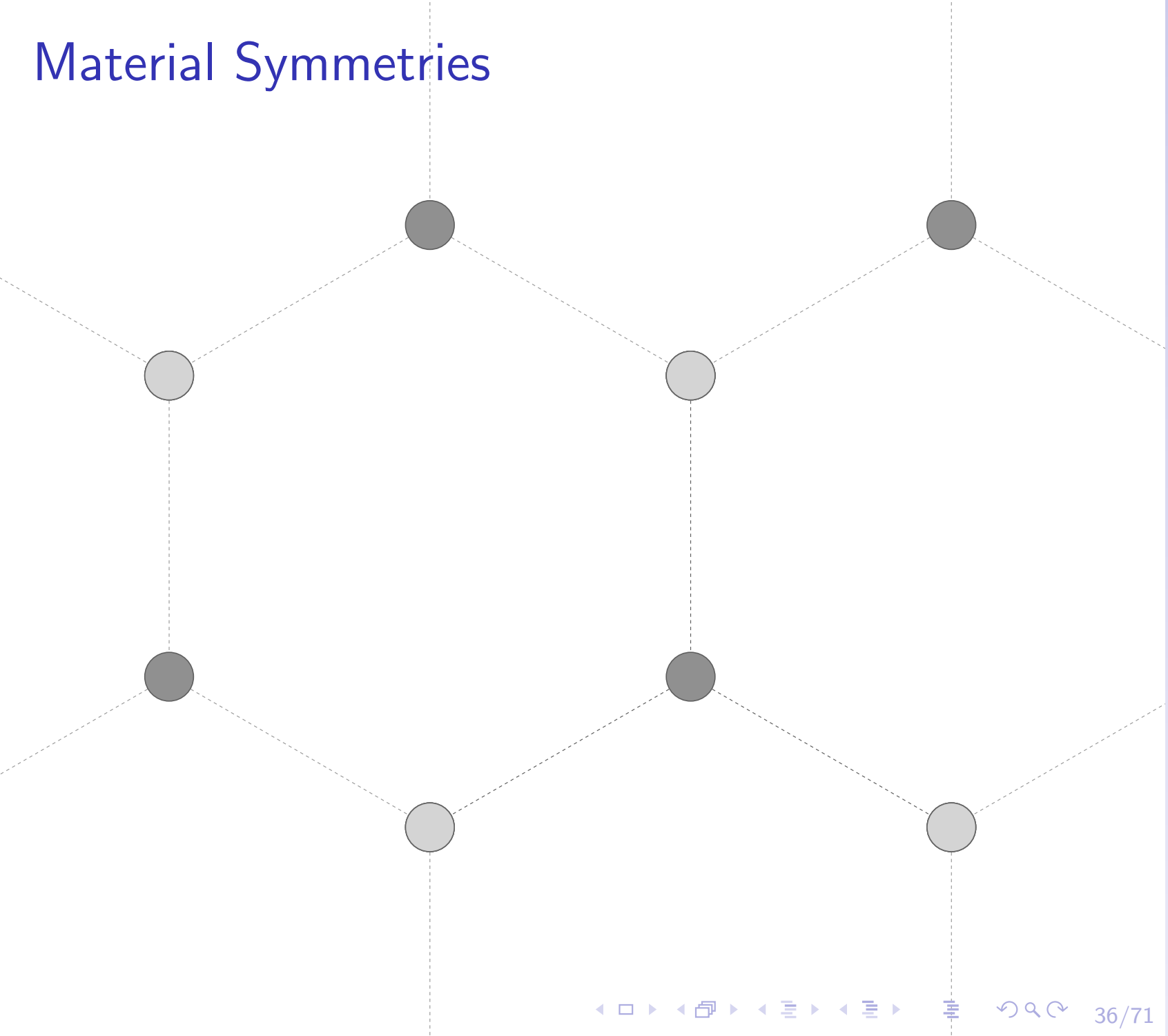
Tensor Prediction with $SO(3)$-Networks

Target Components	MAE (Averaged Over Components) SEGNN	SEConv	SETransformer
Elastic *[GPa]*	8.139	7.689	7.941
Dielectric	4.82	4.702	4.718
Piezoelectric $[C/m^2]$	0.170	0.170	0.1714

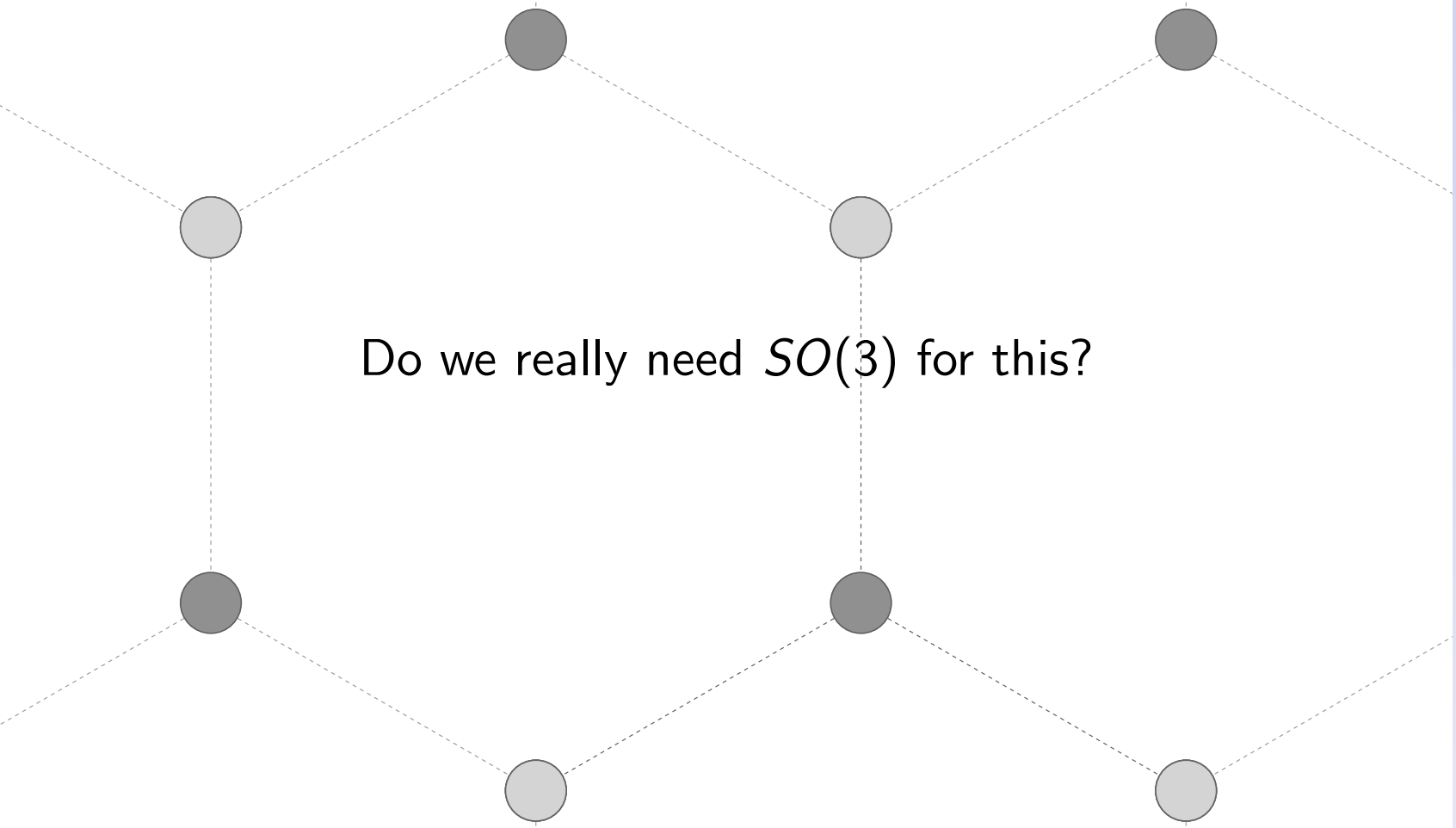
- We can read off the output of $SO(3)$-networks as tensor components
- These naturally respect coordinate transformations acting on the input space!
- Can learn, but doesn't restrict to symmetry of crystals...

Can we get more out of symmetry considerations?

Material Symmetries



Material Symmetries



Do we really need $SO(3)$ for this?

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D
- Finite number of irreps for each

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D
- Finite number of irreps for each
- All irreps of point groups are of dimension $d_\alpha = 1, 2, 3$

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D
- Finite number of irreps for each
- All irreps of point groups are of dimension $d_\alpha = 1, 2, 3$
- All allow a (often reducible) 3D representation

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D
- Finite number of irreps for each
- All irreps of point groups are of dimension $d_\alpha = 1, 2, 3$
- All allow a (often reducible) 3D representation
- Decompositions with $c_{\alpha\beta\gamma} = 0, 1, 2$

Point Group Properties

Groups of operations that leave at least one point fixed and are compatible with a Bravais lattice.

Point groups (in 3D) have following properties:

- 32 crystallographic point groups in 3D
- Finite number of irreps for each
- All irreps of point groups are of dimension $d_\alpha = 1, 2, 3$
- All allow a (often reducible) 3D representation
- Decompositions with $c_{\alpha\beta\gamma} = 0, 1, 2$

(Point group representations may be used to generate induced representations of space groups)

Site-symmetry Equivariant Networks

Index vectors by α, i from site-symmetry group instead!

$$\left(v_{nc}^{L+1}\right)_{n}^{\gamma} = \left(v_{nc}^{L}\right)_{n}^{\gamma} + \sum_{b \in \mathcal{N}(n)} \sum_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} \left(F_{c}^{L}(r_{nb})\right)_{j}^{\beta} \left(v_{bc}^{L}\right)_{i}^{\alpha}$$

Site-symmetry Equivariant Networks

Index vectors by α, i from site-symmetry group instead!

$$\left(v_{nc}^{L+1}\right)_{n}^{\gamma} = \left(v_{nc}^{L}\right)_{n}^{\gamma} + \sum_{b \in \mathcal{N}(n)} \sum_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} \left(F_{c}^{L}(r_{nb})\right)_{j}^{\beta} \left(v_{bc}^{L}\right)_{i}^{\alpha}$$

- Natural cutoff (finite summation over α, β)

Site-symmetry Equivariant Networks

Index vectors by α, i from site-symmetry group instead!

$$\left(v_{nc}^{L+1}\right)_{n}^{\gamma} = \left(v_{nc}^{L}\right)_{n}^{\gamma} + \sum_{b \in \mathcal{N}(n)} \sum_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} \left(F_{c}^{L}(r_{nb})\right)_{j}^{\beta} \left(v_{bc}^{L}\right)_{i}^{\alpha}$$

- Natural cutoff (finite summation over α, β)
- Natural readout for symmetric properties ($\gamma = 1$)

Site-symmetry Equivariant Networks

Index vectors by α, i from site-symmetry group instead!

$$\left(v_{nc}^{L+1}\right)_{n}^{\gamma} = \left(v_{nc}^{L}\right)_{n}^{\gamma} + \sum_{b \in \mathcal{N}(n)} \sum_{\alpha i, \beta j} U_{\alpha i \beta j}^{\gamma n} \left(F_{c}^{L}(r_{nb})\right)_{j}^{\beta} \left(v_{bc}^{L}\right)_{i}^{\alpha}$$

- Natural cutoff (finite summation over α, β)
- Natural readout for symmetric properties ($\gamma = 1$)
- Accounts for all physical symmetry but not more

Applications

With these models, I propose we can predict:

Applications

With these models, I propose we can predict:

- Scalar quantities that are invariant under material symmetry group (not just spherically symmetric)

Applications

With these models, I propose we can predict:

- Scalar quantities that are invariant under material symmetry group (not just spherically symmetric)
- Tensor quantities that respect material symmetry [13]

Applications

With these models, I propose we can predict:

- Scalar quantities that are invariant under material symmetry group (not just spherically symmetric)
- Tensor quantities that respect material symmetry [13]
- Symmetry-adapted Hamiltonian elements [14, 15]

Coupling Coefficients

Coupling coefficients for point groups can be computed from irreps with Dirl's Formula [16]:

$$(U^{\gamma n}_{\alpha i \beta j})^m = \sqrt{\frac{d_\gamma}{N_G}} \Big[\sum_{g \in G} \Gamma^{\alpha}_{qq}(g) \Gamma^{\beta}_{ss}(g) \Gamma^{\gamma\dagger}_{aa}(g) \Big]^{-\frac{1}{2}} \cdot \sum_{g \in G} \Gamma^{\alpha}_{iq}(g) \Gamma^{\beta}_{js}(g) \Gamma^{\gamma\dagger}_{na}(g)$$

Coupling Coefficients

Coupling coefficients for point groups can be computed from irreps with Dirl's Formula [16]:

$$(U^{\gamma n}_{\alpha i \beta j})^m = \sqrt{\frac{d_\gamma}{N_G}} \Big[\sum_{g \in G} \Gamma^{\alpha}_{qq}(g) \Gamma^{\beta}_{ss}(g) \Gamma^{\gamma\dagger}_{aa}(g) \Big]^{-\frac{1}{2}} \cdot \sum_{g \in G} \Gamma^{\alpha}_{iq}(g) \Gamma^{\beta}_{js}(g) \Gamma^{\gamma\dagger}_{na}(g)$$

These have been tabulated several times as well [17, 18].

Coupling Coefficients

Coupling coefficients for point groups can be computed from irreps with Dirl's Formula [16]:

$$(U^{\gamma n}_{\alpha i \beta j})^m = \sqrt{\frac{d_\gamma}{N_G}} \Big[\sum_{g \in G} \Gamma^{\alpha}_{qq}(g) \Gamma^{\beta}_{ss}(g) \Gamma^{\gamma\dagger}_{aa}(g) \Big]^{-\frac{1}{2}} \cdot \sum_{g \in G} \Gamma^{\alpha}_{iq}(g) \Gamma^{\beta}_{js}(g) \Gamma^{\gamma\dagger}_{na}(g)$$

These have been tabulated several times as well [17, 18].

T **70**.11 Clebsch–Gordan coefficients *(cont.)*

^2e	t	T		
		1	2	3
1	1	1	0	0
1	2	0	ω^*	0
1	3	0	0	ω

^2e	$e_{1/2}$	$^2F_{3/2}$	
		1	2
1	1	1	0
1	2	0	1

^2e	$^1f_{3/2}$	$E_{1/2}$	
		1	2
1	1	1	0
1	2	0	1

Projection Operators

- We may project onto the k-th basis function f_k^α of IR α with $\hat{P}_\alpha^{kk}$[19]:

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

where d_α is the dimension of IR α

Projection Operators

- We may project onto the k-th basis function f_k^α of IR α with $\hat{P}_\alpha^{kk}$[19]:

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

where d_α is the dimension of IR α

- For any function, we then have:

$$f_\alpha^k(\vec{r}) = \hat{P}_\alpha^{kk} f(\vec{r})$$

Projection Operators

- We may project onto the k-th basis function f_k^α of IR α with $\hat{P}_\alpha^{kk}$[19]:

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

where d_α is the dimension of IR α

- For any function, we then have:

$$f_\alpha^k(\vec{r}) = \hat{P}_\alpha^{kk} f(\vec{r})$$

- This may also be applied to tensors [13]

Hamiltonian Group

- Operator $\hat{O}_G$ is representation of group G; acts on H as:

$$\hat{O}_G(g)\hat{H}(\vec{r}) = \hat{H}(g^{-1}\vec{r})$$

Hamiltonian Group

- Operator $\hat{O}_G$ is representation of group G; acts on H as:

$$\hat{O}_G(g)\hat{H}(\vec{r}) = \hat{H}(g^{-1}\vec{r})$$

- "Group of the Hamiltonian" is largest commuting group

$$[\hat{H}, \hat{O}(g)] = 0 \qquad \forall g \in G$$

Hamiltonian Group

- Operator $\hat{O}_G$ is representation of group G; acts on H as:

$$\hat{O}_G(g)\hat{H}(\vec{r}) = \hat{H}(g^{-1}\vec{r})$$

- "Group of the Hamiltonian" is largest commuting group

$$[\hat{H}, \hat{O}(g)] = 0 \qquad \forall g \in G$$

- $\hat{O}_G$ must have a simultaneous set of eigenvectors ψ_α^k that spans the space of functions:

$$f(\vec{r}) = \sum_{k,\alpha} c_k^\alpha \psi_\alpha^k = \sum_{k,\alpha} f_\alpha^k(\vec{r})$$

Hamiltonian Group

- Operator $\hat{O}_G$ is representation of group G; acts on H as:

$$\hat{O}_G(g)\hat{H}(\vec{r}) = \hat{H}(g^{-1}\vec{r})$$

- "Group of the Hamiltonian" is largest commuting group

$$[\hat{H}, \hat{O}(g)] = 0 \qquad \forall g \in G$$

- $\hat{O}_G$ must have a simultaneous set of eigenvectors ψ_α^k that spans the space of functions:

$$f(\vec{r}) = \sum_{k,\alpha} c_k^\alpha \psi_\alpha^k = \sum_{k,\alpha} f_\alpha^k(\vec{r})$$

- There is some freedom in choice of this set

Hydrogen Orbitals

The Hydrogen Hamiltonian $\hat{H}_H(\vec{r})$ for a non-relativistic electron has separable eigenfunctions [12]:

$$\psi_i(\vec{r}) = R_i(r)\Omega_i(\theta, \phi)$$

Hydrogen Orbitals

The Hydrogen Hamiltonian $\hat{H}_H(\vec{r})$ for a non-relativistic electron has separable eigenfunctions [12]:

$$\psi_i(\vec{r}) = R_i(r)\Omega_i(\theta, \phi)$$

Furthermore, $\hat{H}_H$ commutes with $SO(3)$ so these must be simultaneous with $SO(3)$:

$$[\hat{H}, \hat{O}(g)] = 0 \quad \forall g \in SO(3) \quad \Rightarrow \quad \psi^{\ell}_{nm}(\vec{r}) = R^{\ell}_n(r)\, Y^{\ell}_m(\theta, \phi)$$

(where R is also indexed by ℓ due to an 'accidental' symmetry)

Hydrogen Orbitals

The Hydrogen Hamiltonian $\hat{H}_H(\vec{r})$ for a non-relativistic electron has separable eigenfunctions [12]:

$$\psi_i(\vec{r}) = R_i(r)\Omega_i(\theta, \phi)$$

Furthermore, $\hat{H}_H$ commutes with $SO(3)$ so these must be simultaneous with $SO(3)$:

$$[\hat{H}, \hat{O}(g)] = 0 \quad \forall g \in SO(3) \quad \Rightarrow \quad \psi^{\ell}_{nm}(\vec{r}) = R^{\ell}_n(r)\, Y^{\ell}_m(\theta, \phi)$$

(where R is also indexed by ℓ due to an 'accidental' symmetry)

- θ, ϕ dependence determined by group theory

Hydrogen Orbitals

The Hydrogen Hamiltonian $\hat{H}_H(\vec{r})$ for a non-relativistic electron has separable eigenfunctions [12]:

$$\psi_i(\vec{r}) = R_i(r)\Omega_i(\theta, \phi)$$

Furthermore, $\hat{H}_H$ commutes with $SO(3)$ so these must be simultaneous with $SO(3)$:

$$[\hat{H}, \hat{O}(g)] = 0 \quad \forall g \in SO(3) \quad \Rightarrow \quad \psi^{\ell}_{nm}(\vec{r}) = R^{\ell}_n(r)\, Y^{\ell}_m(\theta, \phi)$$

(where R is also indexed by ℓ due to an 'accidental' symmetry)

- θ, ϕ dependence determined by group theory
- r dependence determined by specific form of $\hat{H}_H$

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

- Localized at points p as basis for many particle system

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

- Localized at points p as basis for many particle system
- $\hat{H}$ components in H-Like basis:

$$\hat{H}_{nmp\ell,jkhl} = \langle \psi^{l}_{jkh} | \hat{H} | \psi^{\ell}_{nmp} \rangle$$

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

- Localized at points p as basis for many particle system
- $\hat{H}$ components in H-Like basis:

$$\hat{H}_{nmp\ell,jkhl} = \langle \psi^{l}_{jkh} | \hat{H} | \psi^{\ell}_{nmp} \rangle$$

- Have been predicted in earlier works [20]

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

- Localized at points p as basis for many particle system
- $\hat{H}$ components in H-Like basis:

$$\hat{H}_{nmp\ell,jkhl} = \langle \psi^{l}_{jkh} | \hat{H} | \psi^{\ell}_{nmp} \rangle$$

- Have been predicted in earlier works [20]

Now, in point group symmetry adapted basis $|\phi^{\alpha}_{ndp}\rangle$ that transform as irreducible representations [14, 15]

Tight-Binding Approximations

Simplest form: H-like orbitals $|\psi^{\ell}_{nmp}\rangle$

- Localized at points p as basis for many particle system
- $\hat{H}$ components in H-Like basis:

$$\hat{H}_{nmp\ell,jkhl} = \langle \psi^{l}_{jkh} | \hat{H} | \psi^{\ell}_{nmp} \rangle$$

- Have been predicted in earlier works [20]

Now, in point group symmetry adapted basis $|\phi^{\alpha}_{ndp}\rangle$ that transform as irreducible representations [14, 15]

- $\hat{H}$ components in symmetry-adapted basis:

$$\hat{H}_{ndp\alpha,jqh\beta} = \langle \psi^{\beta}_{jqh} | \hat{H} | \psi^{\alpha}_{ndp} \rangle$$

Where to start? (O_h)

- The largest crystalline point group is O_h (Most restriction on $SO(3)$)

Where to start? (O_h)

- The largest crystalline point group is O_h (Most restriction on $SO(3)$)
- Materials Project [21] contains 15,268 materials with this symmetry.

Where to start? (O_h)

- The largest crystalline point group is O_h (Most restriction on $SO(3)$)
- Materials Project [21] contains 15,268 materials with this symmetry.
- Implement working concept model for O_h systems that predicts invariant quantities

Where to start? (O_h)

- The largest crystalline point group is O_h (Most restriction on $SO(3)$)
- Materials Project [21] contains 15,268 materials with this symmetry.
- Implement working concept model for O_h systems that predicts invariant quantities
- Predict symmetry-adapted tensorial properties

Where to start? (O_h)

- The largest crystalline point group is O_h (Most restriction on $SO(3)$)
- Materials Project [21] contains 15,268 materials with this symmetry.
- Implement working concept model for O_h systems that predicts invariant quantities
- Predict symmetry-adapted tensorial properties
- Predict Hamiltonian components in symmetry-adapted Wannier basis

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations, great but lack direction

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations, great but lack direction
- Equivariance

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations, great but lack direction
- Equivariance is important

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations, great but lack direction
- Equivariance is important
- Equivariant networks can predict tensors

Conclusion

Graph Neural Networks for Materials Science

- Hypergraph representations, great but lack direction
- Equivariance is important
- Equivariant networks can predict tensors
- We should use material symmetry groups!

References I

[1] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[2] Tian Xie and Jeffrey C Grossman. "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties". In: *PRL* 120.14 (2018), p. 145301.

[3] K. T. Schütt et al. "SchNet – A deep learning architecture for molecules and materials". In: *J. Chem. Phys.* 148.24 (Mar. 2018), p. 241722. ISSN: 0021-9606. DOI: 10.1063/1.5019779. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5019779/16655678/241722_1_online.pdf. URL: https://doi.org/10.1063/1.5019779.

References II

[4] Chi Chen et al. "Graph networks as a universal machine learning framework for molecules and crystals". In: *Chem. Mater.* 31.9 (2019), pp. 3564–3572.

[5] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *34th Int. Conf. Mach. Learn.* 70 (2017), pp. 1263–1272.

[6] Taco Cohen and Max Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 2990–2999. URL: https://proceedings.mlr.press/v48/cohenc16.html.

References III

[7] Nathaniel Thomas et al. "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds". In: (2018). arXiv: 1802.08219 [cs.LG].

[8] Mario Geiger and Tess Smidt. "e3nn: Euclidean neural networks". In: *arXiv:2207.09453* (2022).

[9] Nathaniel Thomas et al. "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds". In: *arXiv:1802.08219* (2018). arXiv: 1802.08219 [cs.LG]. URL: https://arxiv.org/abs/1802.08219.

[10] Eiji Mochizuki. "Spherical harmonic decomposition of an elastic tensor". In: *Geophysical Journal International* 93.3 (1988), pp. 521–526. URL: https://academic.oup.com/gji/article/93/3/521/662002.

References IV

[11] Eiji Mochizuki. "Spherical Harmonic Decomposition of an Elastic Tensor II". In: *Journal of Physics of the Earth* 44.1 (1996), pp. 79–84. URL: https://www.jstage.jst.go.jp/article/jpe1952/44/1/44_1_79/_article/-char/ja/.

[12] Jun John Sakurai and Jim Napolitano. *Modern quantum mechanics*. Cambridge University Press, 2020.

[13] Hiroaki Kusunose, Rikuto Oiwa, and Satoru Hayami. "Symmetry-adapted modeling for molecules and crystals". In: *Phys. Rev. B* 107 (19 2023), p. 195118. DOI: 10.1103/PhysRevB.107.195118. URL: https://link.aps.org/doi/10.1103/PhysRevB.107.195118.

References V

[14] R. Sakuma. "Symmetry-adapted Wannier functions in the maximal localization procedure". In: *Physical Review B* 87.23 (June 2013). ISSN: 1550-235X. DOI: 10.1103/physrevb.87.235109. URL: http://dx.doi.org/10.1103/PhysRevB.87.235109.

[15] Rikuto Oiwa et al. "Symmetry-adapted closest Wannier modeling based on complete multipole basis set". In: *Physical Review B* 112.3 (July 2025). ISSN: 2469-9969. DOI: 10.1103/2s5q-p42x. URL: http://dx.doi.org/10.1103/2s5q-p42x.

[16] R Dirl. "Clebsch–Gordan coefficients: General theory". In: *Journal of Mathematical Physics* 20.4 (1979), pp. 659–663.

[17] George F. Koster et al. *Properties of The Thirty-Two Point Groups*.

References VI

[18] Simon L. Altmann and Peter Herzig. *Point-group Theory Tables*.

[19] Artur F Izmaylov. "On construction of projection operators". In: *The Journal of Physical Chemistry A* 123.15 (2019), pp. 3429–3433.

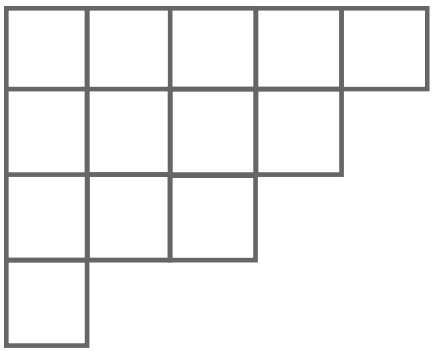
[20] Xiaoxun Gong et al. "General framework for E (3)-equivariant neural network representation of density functional theory Hamiltonian". In: *Nature Communications* 14.1 (2023), p. 2848.

References VII

[21] Anubhav Jain et al. “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL Mater.* 1.1 (July 2013), p. 011002. ISSN: 2166-532X. DOI: 10.1063/1.4812323. eprint: https://pubs.aip.org/aip/apm/article-pdf/doi/10.1063/1.4812323/13163869/011002_1_online.pdf. URL: https://doi.org/10.1063/1.4812323.

GL Decomposition

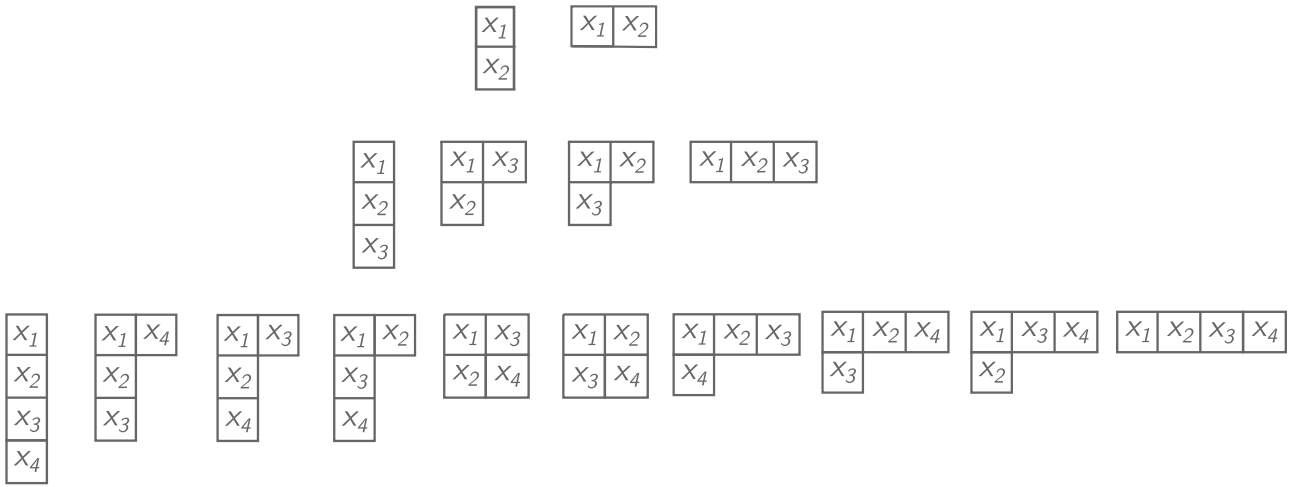
- Decompositions under general linear group *GL* are simultaneous with decompositions under symmetric group *S* (*Schur-Weyl Duality*)
- Irreducible representations of symmetric group are diagramatically described by Young diagrams.



Young diagrams are said to be of some shape $\lambda : (\lambda_1, \lambda_2, ..., \lambda_k)$, where λ_i refers to the depth of row i and $\lambda_{i+1} \leq \lambda_i \leq \lambda_{i-1}$. Above: $(5, 4, 3, 1)$

GL Decomposition cont.

We can then form a set of Young tableaux from diagrams by filling in the boxes from a set of ordered indices $\{x_1, x_2, ..., x_k\}$ corresponding to tensor components $T^{x_1 x_2 \dots x_k}$.



A standard tableu is one filled with indices x_i (without repeats) with entries increasing in index i down each column and across (to the right) rows.

GL Decomposition cont.

Each of these standard tableaux correspond to an invariant subspace under S_k.

These invariant subspaces may be constructed by correspond products of symmetrizers s and antisymmetrizers a:

$$s_\lambda = \prod_{\mathcal{I} \in \text{Cols}(\lambda)} \mathcal{S}(\mathcal{I})$$

$$a_\lambda = \prod_{\mathcal{I} \in \text{Rows}(\lambda)} \mathcal{A}(\mathcal{I})$$

where $\mathcal{S}$ and $\mathcal{A}$ act component-wise as:

$$\left[\mathcal{S}(\mathcal{I})\, T\right]_{ijk\ldots} = \sum_{\sigma_\mathcal{I}} T_{\sigma_\mathcal{I}(ijk\ldots)}$$

$$\left[\mathcal{A}(\mathcal{I})\, T\right]_{ijk\ldots} = \sum_{\sigma_\mathcal{I}} \text{sgn}(\sigma_\mathcal{I})\, T_{\sigma_\mathcal{I}(ijk\ldots)}$$

O Decomposition

Orthogonal group O, preserves inner product between vectors $< \cdot, \cdot >$. defined by means of a metric tensor g_{ij} , which transforms as a second rank tensor under some transformation R as:

$$g_{x_1 x_2} \rightarrow g_{x_1' x_2'} = R^{x_1}_{x_1'} R^{x_2}_{x_2'} g_{x_1 x_2}$$

Contractions of arbitrary tensors T with g_{ij} are invariant under O.

For example, consider the rank-3 covariant T^{ijk}:

$$g_{i'j'} T^{i'j'k'} = S^{k'} = R^i_{i'} R^j_{j'} g_{ij} R^{i'}_i R^{j'}_j R^{k'}_k T^{ijk} = R^{k'}_k S^k$$

so that S acts like an invariant vector subspace under O.

SL Decomposition

Special linear group SL of transformations is defined as invertible linear transformations with determinant equal to positive one.

Under SL, orientation and volume are preserved, where volume is defined as the contraction of a tensor with the fully antisymmetric tensor ϵ_{ijk}, which transforms under $R \in GL(3)$ as:

$$\epsilon_{x_1 x_2 x_3} \rightarrow \epsilon_{x_1' x_2' x_3'} = \det(R) R^{x_1}_{x_1'} R^{x_2}_{x_2'} R^{x_3}_{x_3'} \epsilon_{x_1 x_2 x_3}$$

Similar to the case of g_{ij} in O, contractions with ϵ_{ijk} of arbitrary tensors yield SL invariant subspaces.

$SO(3)$ Decompositions

In $SO(3)$, we may use all of the above (Young, g_{ij}, ϵ_{ijk}):

- Young symmetrizers return a set of tensors of known symmetries (under index permutation).
- Contractions with g_{ij} along antisymmetric pairs of indices vanish.
- Contractions with ϵ_{ij} along symmetric sets of indices vanish.

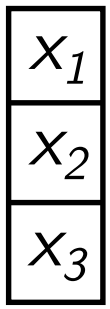
These may all be used together to decompose an arbitrary tensor into a set of symmetric, $SO(3)$ invariant, symmetric tensor subspaces. These then may be related to harmonic coefficients by way of the CG expansion from the J_z basis given before.

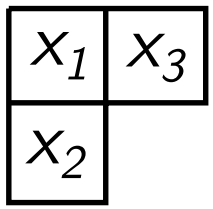
Example: Piezoelectric Tensors

The piezoelectric strain components d_{ijk} are symmetric under i, j so that:

$$d_{ijk} = d_{jik}$$

according to this symmetry, we see all Young tableaux but the following must disappear:





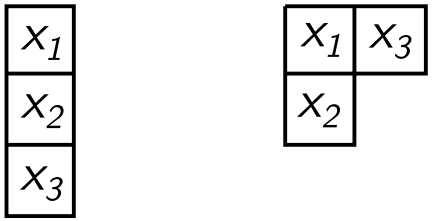
defined component-wise (using the defining symmetry):

$$S_{ijk} = \frac{1}{3}\left(d_{ijk} + d_{jki} + d_{ikj}\right)$$

$$A_{ijk} = \frac{1}{3}\left(2d_{ijk} - d_{jki} - d_{ikj}\right)$$

Example: Piezoelectric Tensors

This may be derived as (adopting the convention of symmetrization before antisymmetrization):



$$\mathcal{S}(ijk) \qquad \mathcal{A}(ij)\mathcal{S}(ik)$$

$$\begin{aligned}\mathcal{S}(ijk)d_{ijk} &= d_{ijk} + d_{ikj} + d_{kji} + d_{jki} + d_{kij} + d_{kji} \\ &= 2(d_{ijk} + d_{ikj} + d_{kji})\end{aligned}$$

$$\begin{aligned}\mathcal{A}(ik)\mathcal{S}(ij)d_{ijk} &= \mathcal{A}(ij)(d_{ijk} + d_{jik}) \\ &= d_{ijk} - d_{kji} + d_{jik} - d_{jki} \\ &= 2d_{ijk} - d_{kji} - d_{jki}\end{aligned}$$

Where the respective normalization coefficients (neglected here) may be derived from the diagram's shape via the hook-length formula.

Example: Piezoelectric Tensors

Further decompose A into the trace vector v_i:

$$v^i = g_{jk} A^{ijk}$$

and the traceless, symmetric tensor b_{ij}:

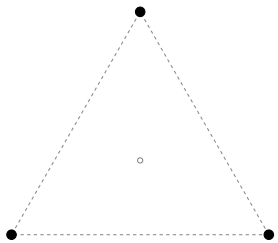
$$b_{ij} = \frac{1}{2}\left(\epsilon_i^{mk} A_{mkj} + \epsilon_j^{mk} A_{mki}\right)$$

Group Representations

A representation ρ_G of a group G is a homomorphism from elements g to a set of linear operators (square matrices).

Example: 3D Representation of C_3

Consider three identical points in 3D space forming an equilateral triangle.



These clearly are symmetrical under three-fold rotations about the origin in the xy plane. These C_3 group actions act on this Cartesian basis with the representation ρ defined:

$$\rho(\mathbb{I}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \rho(C_3) = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \rho(C_3^2) = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Irreducible Representations (IRs)

For atomic arrangements, 3D group representations ρ are often reducible in terms of a direct sum of 'smaller' group representations $\rho^{(\alpha)}$:

$$\rho = \bigoplus_{\alpha} c_\alpha \rho^{(\alpha)}$$

Maschke's theorem guarantees that any given representation is always decomposable as a direct sum of irreducible representations.

This set may always be taken to satisfy:

- Unitarity
- Orthogonality
- $\sum_\alpha d_\alpha^2 = N$ where d_α is the dimension of IR α and N is the order

IRs (cont.)

Example: IRs of D_3 The previously shown representation of C_3 elements is reducible into a two-dimensional subspace and a one-dimensional subspace of D_3.

$$\rho(\mathbb{I}) = \rho^{(2)}(\mathbb{I}) \oplus \rho^{(1)}(\mathbb{I}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \end{bmatrix}$$

$$\rho(C_3) = \rho^{(2)}(C_3) \oplus \rho^{(1)}(C_3) = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \oplus \begin{bmatrix} 1 \end{bmatrix}$$

$$\rho(C_3^2) = \rho^{(2)}(C_3^2) \oplus \rho^{(1)}(C_3^2) = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \oplus \begin{bmatrix} 1 \end{bmatrix}$$

Orthogonality Theorems

IRs are orthogonal in the following ways:

$$\frac{1}{N}\sum_{g}\chi^{(\alpha)*}(g)\chi^{(\beta)}(g) = \delta_{\alpha\beta}$$

$$\sum_{\alpha}\chi^{(\alpha)*}(c_k)\chi^{(\alpha)}(c_h) = \frac{N}{N_k}\delta_{kh}$$

where N is the number of elements in G and N_k is the number of elements in equivalence class k.

This allows us to decompose reducible representations by determining the coeffecients of expansion c_α as:

$$c_\alpha = \frac{1}{N}\sum_{g}\chi^{(\alpha)*}(g)\chi(g)$$

Orthogonality Theorems (cont.)

Example: d-shell Splitting in Octohedral Coordinations
Take the Hydrogen-like orbitals $\psi_{\ell m}$ as a basis for spherically symmetric states. The d-shell orbitals are the basis functions of the $\ell = 2$ representations.

The octohedral complex's symmetry group is O, with it's character table and the $\Gamma^{\ell=2}$ representation:

O	$1\langle\mathbb{I}\rangle$	$8\langle C_3\rangle$	$3\langle C_2\rangle$	$6\langle C_2'\rangle$	$6\langle C_4^3\rangle$
$(d)\ \Gamma^{\ell=2}$	5	-1	1	1	-1
A_1	1	1	1	1	1
A_2	1	1	1	-1	-1
E	2	-1	2	0	0
T_1	3	0	-1	-1	1
T_2	3	0	-1	1	-1

Orthogonality then gives $\gamma^{\ell=2} = E \oplus T_2$. In practice, this results in a 5-fold degeneracy being lifted into a two- and three-fold degeneracy.

Basis Functions

Every representation of a group inherits a vector space on which it has a natural action. This vector space is spanned by a chosen set of basis functions $|\psi_i^\alpha\rangle$.

- An irrep Γ^α's components are determined by basis:

$$\left(\Gamma^\alpha(g)\right)_i^j = \langle\psi_j^\alpha|\Gamma^\alpha(g)|\psi_i^\alpha\rangle$$

- Basis functions are said to "transform as an irrep α" if:

$$\Gamma^\alpha(g)|\psi_i^\alpha\rangle = \sum_j \left(\Gamma^\alpha(g)\right)_i^j |\psi_j^\alpha\rangle$$

Basis Functions (cont.)

Example: $3D$ Basis Functions of D_3
Consider previous $3D$ representation of C_3 rotations in D_3, which act on vectors $\vec{r}_\rho$:

$$\vec{r}_\rho = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \vec{r}_{\rho^{(2)}} \oplus \vec{r}_{\rho^{(1)}} = \begin{bmatrix} x \\ y \end{bmatrix} \oplus \begin{bmatrix} z \end{bmatrix}$$

Clearly, x, y is a basis transforming as Γ^2 and z is a basis for Γ^1.

Example: Tight Binding
The tight binding approximation often uses localized Hydrogen-like orbitals ψ^ℓ_{nm} as a basis for many-body systems

Projection Operators

If we have an explicit form for IR α, we may project an arbitrary function onto the k-th basis function f_k^α of IR α with $\hat{P}_\alpha^{kk}$:

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

where d_α is the dimensional of IR α, and then we have:

$$f_\alpha^k(\vec{r}) = \hat{P}_\alpha^{kk} f(\vec{r})$$

From the characters alone, we may project a function onto it's total α subspace with $\hat{P}_\alpha$:

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

Coupling Coefficients

Consider a direct product decomposition of some irreps Γ:

$$\Gamma^\alpha \otimes \Gamma^\beta = \bigoplus_\gamma c_{\alpha\beta\gamma}\Gamma^\gamma$$

Products of basis functions $u_i^\alpha v_j^\beta$ then decompose similarly into a direct sum of irreps with basis functions ψ_n^γ via the coupling coefficients $U^{\gamma n}_{\alpha i \beta j}$ as:

$$\psi_n^\gamma = \sum_{i,j} U^{\gamma n}_{\alpha i \beta j} u_i^\alpha v_j^\beta$$

Example: Clebsch-Gordan Coefficients

The Clebsch-Gordan coefficients $C^{\ell_f m_f}_{\ell_1 m_1 \ell_2 m_2}$ are the coupling coefficients of $SO(3)$, which relate tensor product spaces of spherical harmonics Y^ℓ_m to direct sums of spherical harmonics.

$$Y^{m_1}_{\ell_1}(\Omega) Y^{m_2}_{\ell_2}(\Omega) = \sum_{\ell_3, m_3} \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)}{4\pi(2\ell_3+1)}} C^{\ell_3 m_3}_{\ell_1 m_1 \ell_2 m_2} C^{\ell_3 0}_{\ell_1 0 \ell_2 0} Y_{\ell_3 m_3}(\Omega)$$