Point Group Equivariant Networks for Material Systems

Thesis Proposal

Alex Heilman

July 22, 2025

Graph Neural Networks

Crystal Hypergraphs

$O(3)$-Equivariant Networks

Tensor Prediction

Material Symmetry

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?

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 between abstract input and output spaces exist:

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

and that these may be expressed in a basis of adequately large neural networks.

But how do we apply this to material systems?

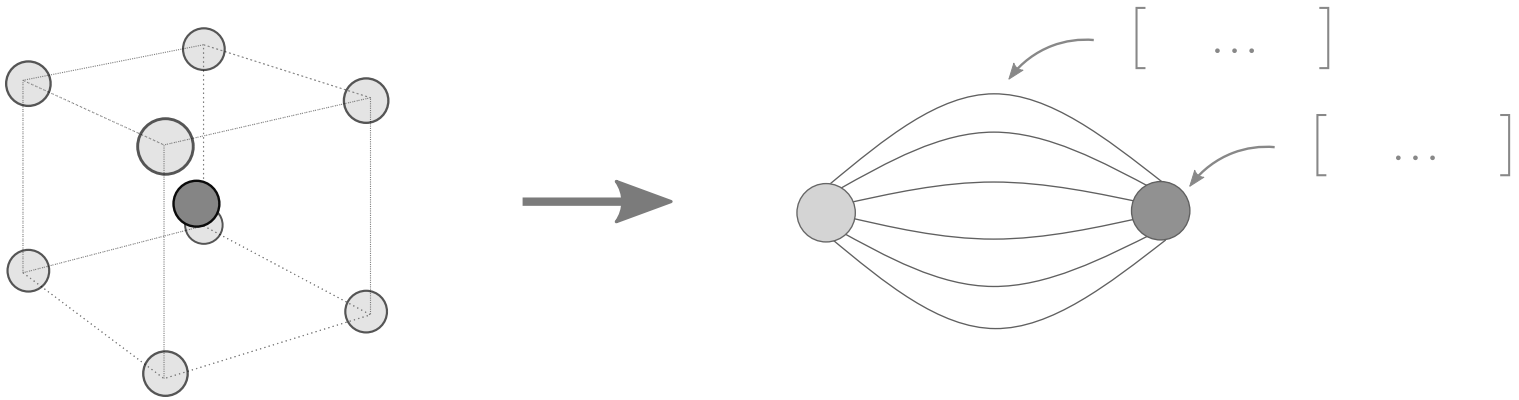
Machine Learning on Crystal Systems

To perform predictive tasks for material systems, such as crystalline structures, we essentially need two things:

- A way to represent the material system mathematically
- A trainable predictive model or set of functions which takes, as input, the material system's representation (as well as a large set of data)

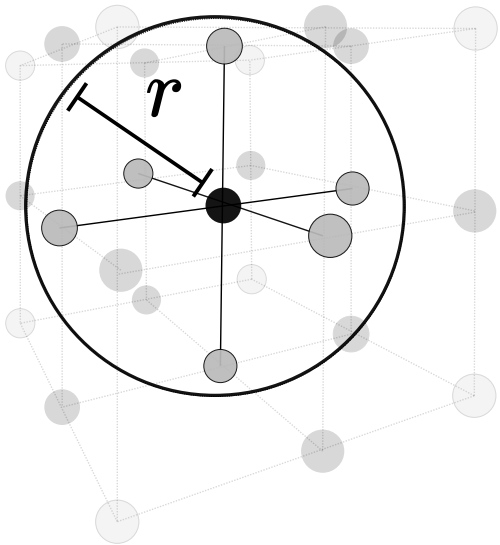
Usual Crystal Graph Construction (a la CGCNN)

The usual technique is to represent crystalline systems as graphs (nodes and edges), with atomic features associated with nodes and geometric features associated with edges

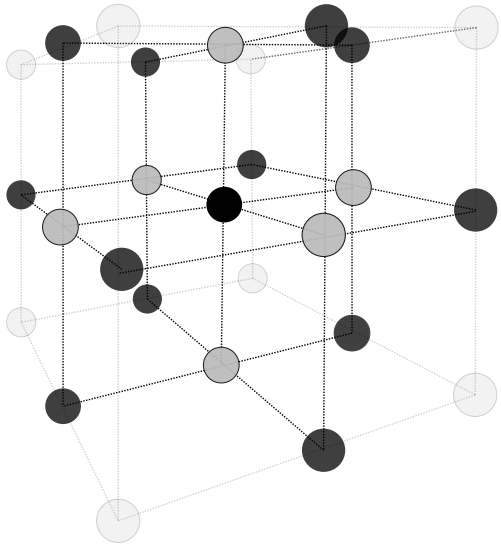


Crystal Graphs cont. I

- Edges are determined by distance cutoff (4 Ang.) and maximum number of neighbors (12)



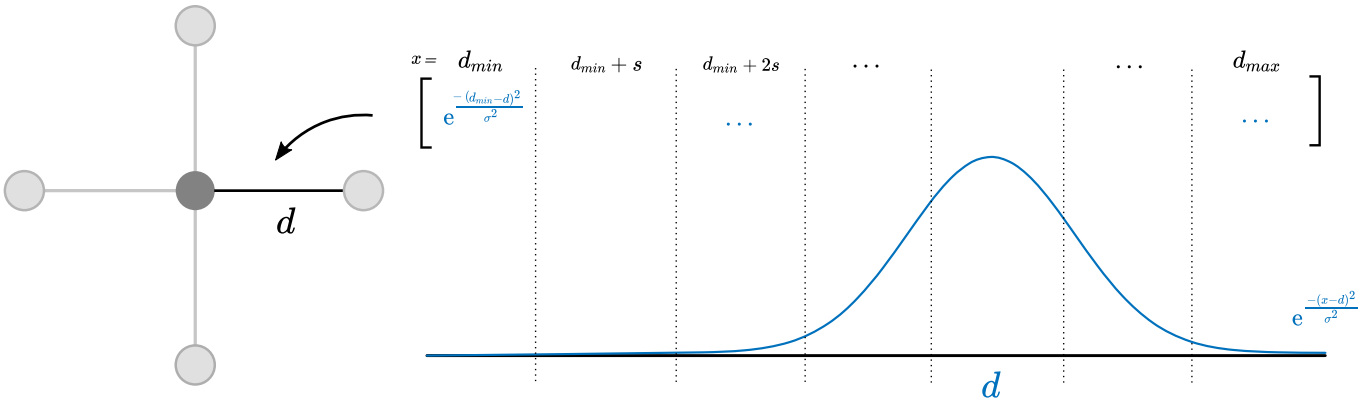
(within) Max radius r



N nearest neighbors

Crystal Graphs cont. II

Edge attributes then are a Gaussian distance expansion



Message Passing on Graphs

Now, we need to update these features through some trainable function that acts on graph representations.

This is most generally accomplished by a message passing network applied to graph representations.

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

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

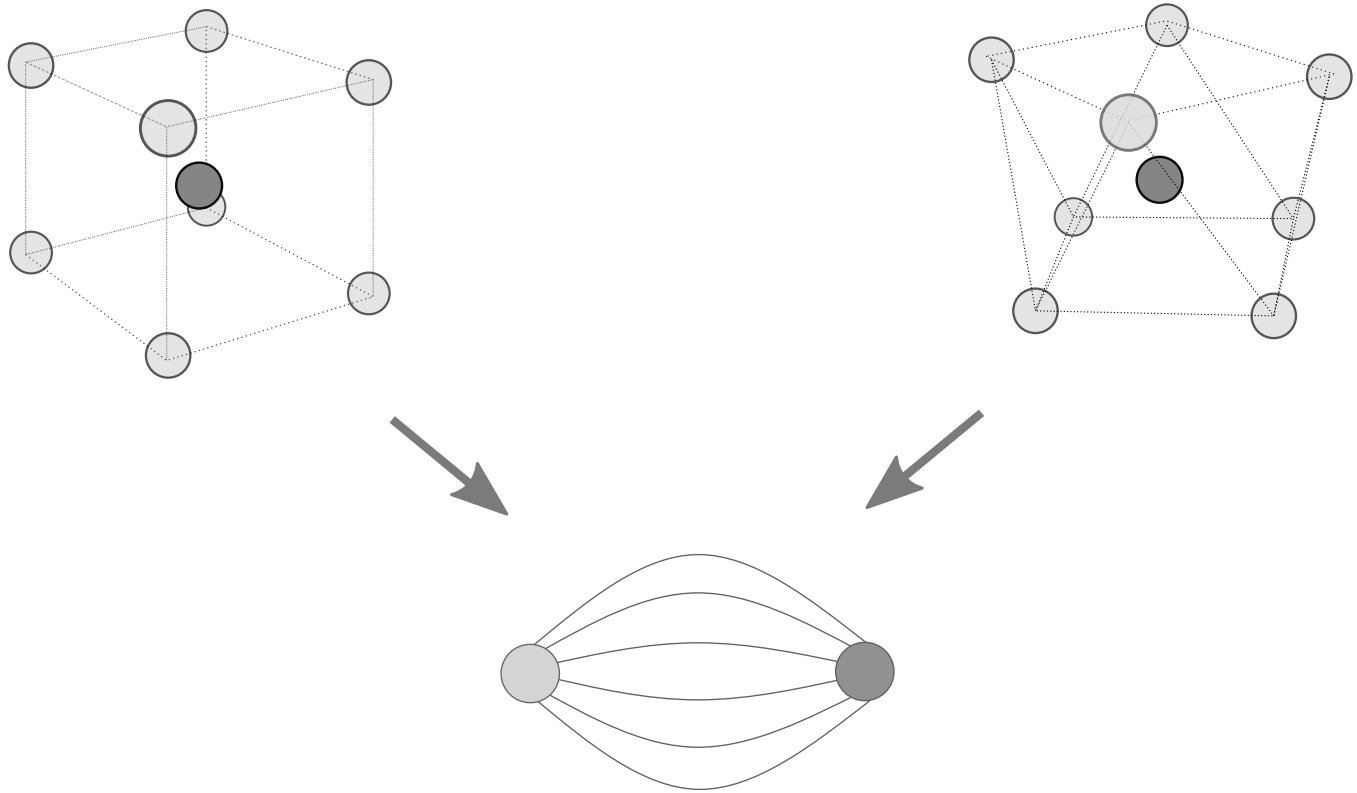
Here, each node n from layer t to $t+1$ is updated according to an update function U, which takes as input messages formed from each pair of nodes containing the node to be updated.

Graph Neural Network

Add schematic with pooling

Graph Limitations

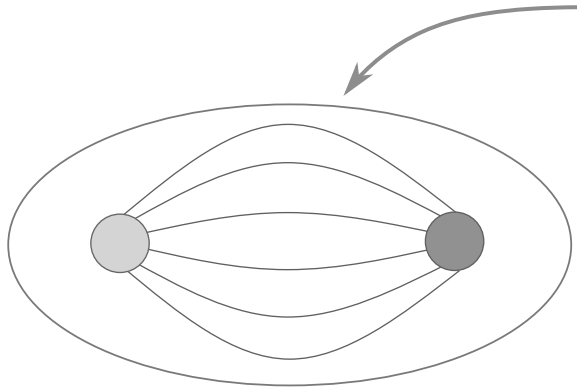
Problem: Our underlying representation encodes only distances between atoms!



So, as a rough example, the above two crystalline structures would have the same representations!

Solution: Hypergraphs!

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

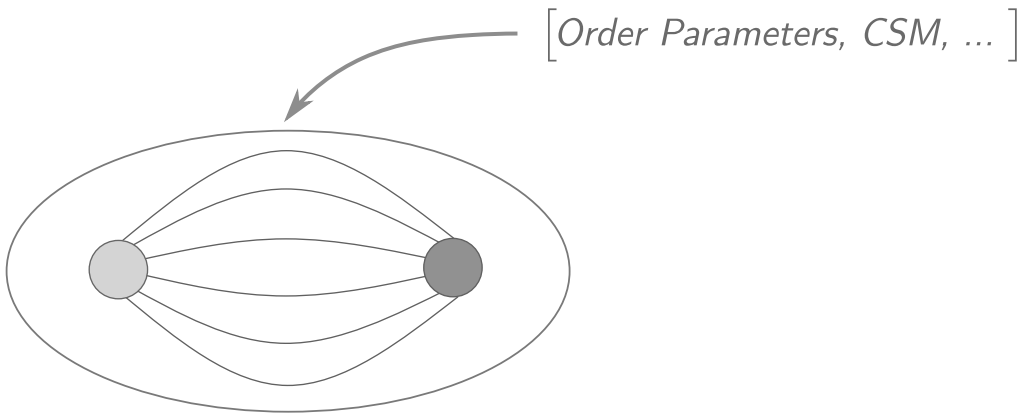


This gives us a natural way to encode features with higher order structure, where hypergraph nodes still represent atoms of underlying crystal structure.

Can treat all different order structures in crystals on equal footing: each has a corresponding hyperedge with a feature

Local Environments as Hyperedges

Considering our previous problem, we can now encode this lost geometrical information in larger hyperedges:



This geometrical information can be encoded quantitatively with continuous symmetry measures, local structure order parameters, etc., depending on task

Extending Message Passing to Hypergraphs

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

In this case, the neighborhood of nodes relevant to each message are now a set (instead of the single neighboring node feature of classic MPNNs)

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

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

Hypergraph Convolution

To deal with the variable sized neighborhoods of features, we simply aggregate the neighborhood features in the message passing phase, so that we essentially have the following form of convolution:

$$x_i \to x_i + f_b(x_i \oplus e_{ij} \oplus \text{AGG}(\{x_i, x_j\})) + f_m(x_i \oplus m_j \oplus \text{AGG}(\{n_w^t | n_w \in m_j\}))$$

where we only consider pair-wise and local environment hyperedges above (with other order structures, such as triplet and unit cell level hyperedges also fitting into this framework). This 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

An equivariant function $f : X \to Y$, where X, Y are vector spaces, is one that 'commutes' with a group's actions, satisfying:

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

Tensor products are equivariant with respect to their argument vector spaces.
Equivariance is always with respect to some group of transformations!

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!

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

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

$$v_i^{\alpha}$$

with indexes α denoting the irrep it transforms as, and i denoting the dimension of irrep space α.

Equivariant Networks (cont.)

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

- Self-interaction:

$$v'_{nc} = W^i_c v_{ni}$$

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

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

- Group correlation with filters:

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

- Tensor products with filters:

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

- Pooling over nodes or channels:

$$v_c = \sum_n v_{nc}$$

$SO(3)$ Properties

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

- 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 for the Euclidean group (add parity and translation invariance).

Tensor Field Networks

Tensor field networks [?] 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!

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

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):

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

So, define J_z basis:

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

so that $\hat{n} = a_+ Y_1^1 + a_0 Y_1^0 + a_- Y_1^{-1}$

Clebsch-Gordon Expansion

Build larger spherical harmonic tensors with CG expansion:

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

where Y_L represents a $2L+1$ dimensional symmetric tensor space of rank L.
This serves as a relation between symmetric tensor's J_z basis components and higher order 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

We can always reduce an arbitrary tensor T that transforms under a transformation as:

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

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

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

$SO(3)$ Invariants Ex.: Rank 2

Consider a rank-two tensor E_{ij}:
Decompose into symmetric S

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

Then take trace $t = g_{ij}S_{ij}$ and residue $R = S - tg$.
And, antisymmetric A:

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

Tensor Prediction

Add schematic

Tensor Prediction (cont.)

Add results

Tensor Prediction with $SO(3)$-Networks

- 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

Add point group schematic

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!

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

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

Coupling Coefficients

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

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

With these models we can predict:

- Scalar quantities that are invariant under material symmetry group (not just spherically symmetric)
- Tensor quantities that respect material symmetry
- Symmetry-adapted Hamiltonian elements

Hamiltonian Group

We define the operator $\hat{O}_G$ to be a representation of a group G that acts on H's input space $\vec{r}$ as:

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

The "group of the Hamiltonian" is the largest group of the form above that commutes with the Hamiltonian.

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

In such a case, the operators must have a simultaneous set of eigenvectors ψ_α^k that span the space of functions:

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

We have some freedom in choice of this set of basis functions on which the linear operators of the representations act.

Hydrogen Orbitals

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

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

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

In simplest form, consider H-like orbitals $|\psi^{\ell}_{nmp}\rangle$ localized at points p as basis for many particle system defined by $\hat{H}$ so:

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

Now, we can incorporate point group symmetry of crystal and molecular sites by directly learning features associated with basis functions that transform as irreducible representations.

Where to start? (O_h)

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

References