Stony Brook University

Equivariant GNNs

Wenhan Gao

Ph.D. student at Stony Brook, supervised by Prof. Yi Liu

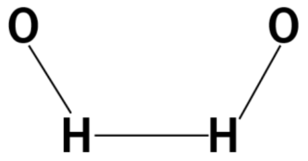
Department of Applied Mathematics and Statistics

Department of Computer Science

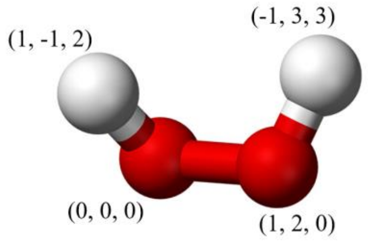
FAR BEYOND

Geometric Representation of Atomistic Systems

Representation of Atomistic Systems:



Molecules as 2D (planar) graphs



Molecules as 3D (geometric) graphs

- ➢ 3D geometric configuration (coordinates) is crucial in determining properties.
- ➢ 3D representations outperforms their 2D counterparts by a large margin.

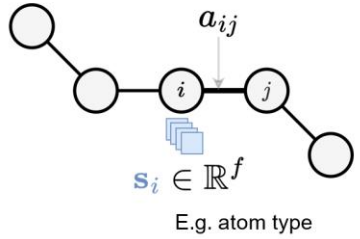
Model	MAE **(Lower, better)**
GIN-Virtual	0.2371
SchNet	0.1511
DimeNet++	0.1214
SphereNet	0.1182
ComENet	0.1273

Best 2D GNN

3D GNNs outperform 2D GNNs by a large margin

Graphs and Geometric Graphs

Graphs are purely topological objects.



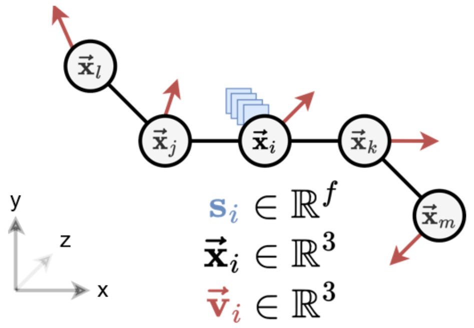
$G = (A, S)$

$A \in \mathbb{R}^{n \times n}$: Adjacency matrix

$S \in \mathbb{R}^{n \times a}$: Scalar features

a is the dimension or number of scalar feature channels.

Geometric graphs are a type of graphs where nodes are additionally endowed with geometric information.



$\vec{\mathbf{x}}_l$
$\vec{\mathbf{x}}_j$
$\vec{\mathbf{x}}_i$
$\vec{\mathbf{x}}_k$
$\vec{\mathbf{x}}_m$
y
z
x
$\mathbf{s}_i \in \mathbb{R}^f$
$\vec{\mathbf{x}}_i \in \mathbb{R}^3$
$\vec{\mathbf{v}}_i \in \mathbb{R}^3$

$G = (A, S, X, V)$

$A \in \mathbb{R}^{n \times n}$: Adjacency matrix

$S \in \mathbb{R}^{n \times a}$: Scalar features

$X \in \mathbb{R}^{n \times d}$: Coordinates

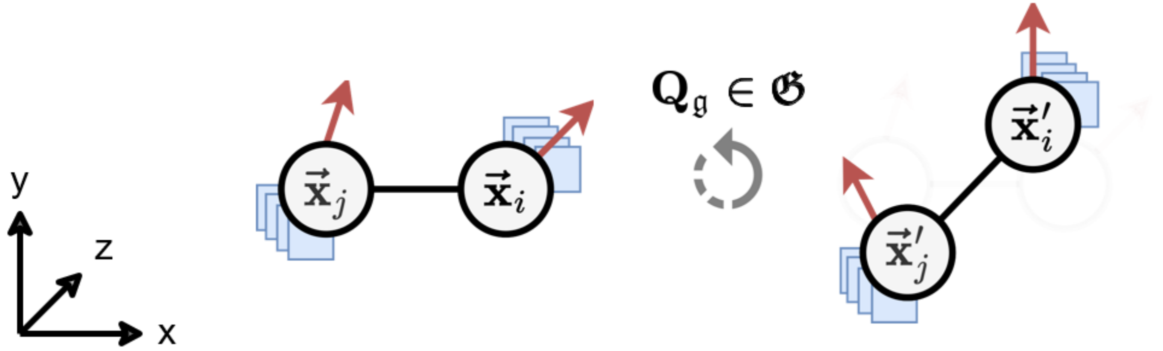
$V \in \mathbb{R}^{n \times b \times d}$: Geometric features

b is the dimension or number of geometric feature channels.

Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Physical Symmetries

We have two types of features: **Scalar features** and **Geometric features**. **Geometric features transform with Euclidean transformations** of the system (equivariance); **Scalar features remain unchanged** (invariance).



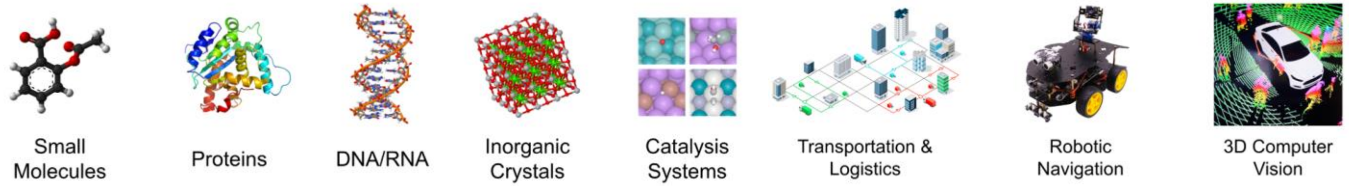
$$\mathbf{s}_i' = \mathbf{s}_i$$

$$\vec{\mathbf{x}}_i' = \mathbf{Q}_\mathfrak{g}\vec{\mathbf{x}}_i$$

$$\vec{\mathbf{v}}_i' = \mathbf{Q}_\mathfrak{g}\vec{\mathbf{v}}_i$$

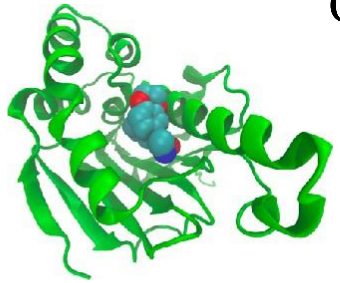
Geometric Graph Neural Networks

Why we care about geometric GNNs? There are many systems with geometric & relational structures.



Geometric GNNs is a fundamental tool for machine learning on geometric (3D) graphs.

Geometric Graph → Geometric GNN → Prediction



- Functional properties?
- Ligand binding affinity?
- Ligand efficacy?

Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Review: Equivariance and Invariance

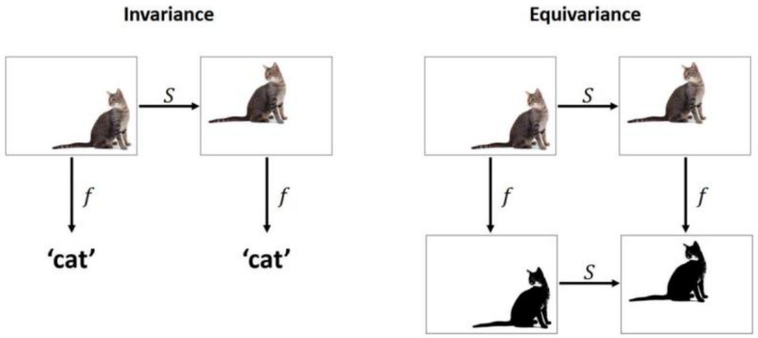
Equivariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it commutes with the group action:

$$\Phi \circ \rho^X(g) = \rho^Y(g) \circ \Phi,$$

Invariance is a property of an operator $\Phi : X \to Y$ (such as a neural network layer) by which it remains unchanged after the group action:

$$\Phi \circ \rho^X(g) = \Phi,$$

- $\rho^X(g)$: group representation action on X
- $\rho^Y(g)$: group representation action on Y
- Invariance is a special case of equivariance when $\rho^Y(g)$ is the identity.



Geometric GNNs should account for physical symmetries

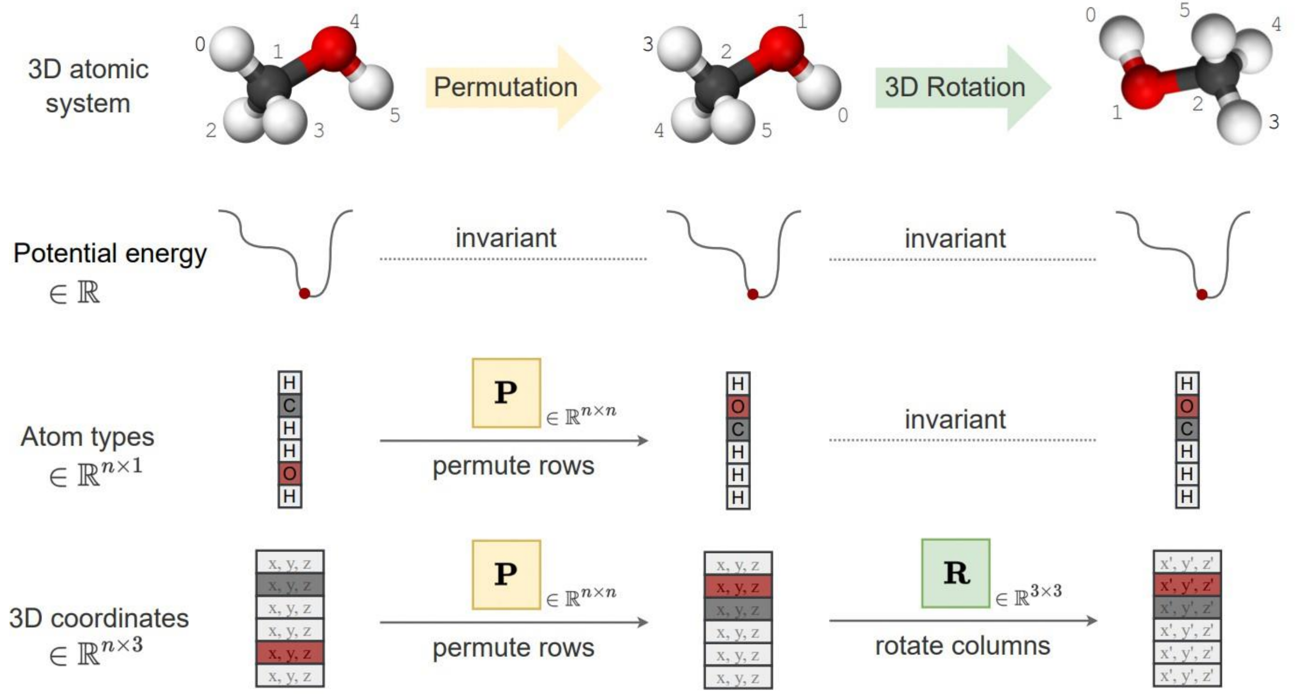
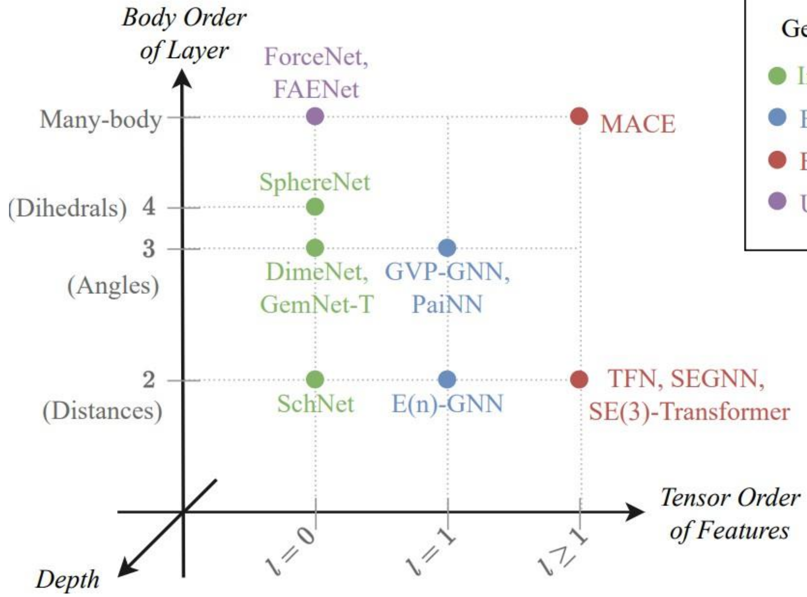

3D atomic system
Permutation
3D Rotation
0 1 2 3 4 5
Potential energy $\in \mathbb{R}$
invariant
invariant
Atom types $\in \mathbb{R}^{n\times 1}$
H C H H O H
$\mathbf{P} \in \mathbb{R}^{n\times n}$
permute rows
H O C H H H
invariant
3D coordinates $\in \mathbb{R}^{n\times 3}$
x, y, z
permute rows
$\mathbf{R} \in \mathbb{R}^{3\times 3}$
rotate columns
x', y', z'

Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Building blocks of Geometric GNNs

- **Scalar features must be updated in an invariant manner.**
- **Geometric features must be updated in an equivariant manner.**



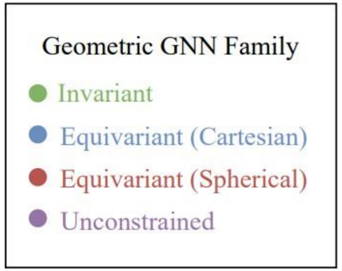
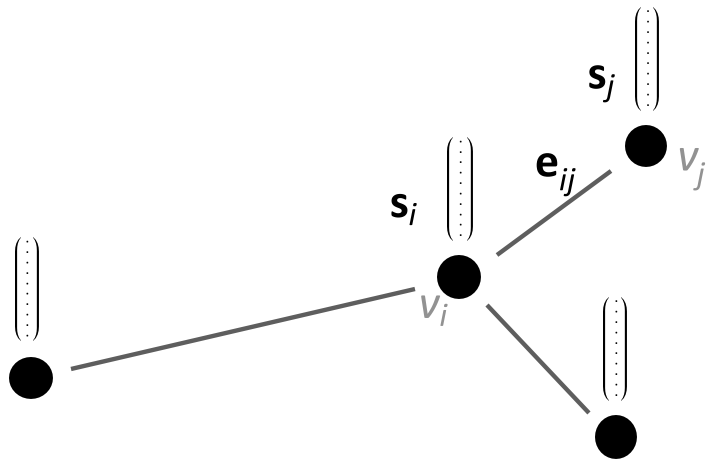


Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Review: The Message Passing Framework



$G = (A, S, E)$

$A \in \mathbb{R}^{n \times n}$: Adjacency matrix

$S \in \mathbb{R}^{n \times a}$: Node features

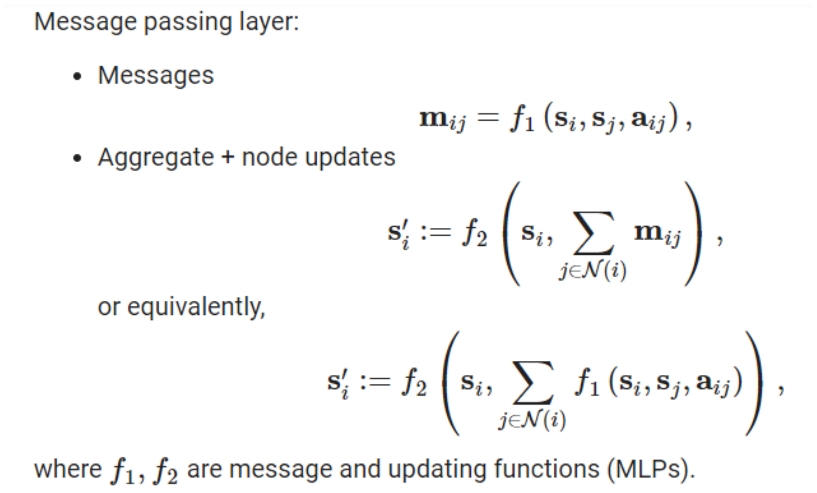
$E \in \mathbb{R}^{n \times n \times b}$: Edges features

Goal of Message Passing: iteratively update node features to obtain useful hidden representations

Stony Brook University

Review: The Message Passing Framework

$$\begin{tikzcd}
\text{Input Graph} \arrow[d, "\text{Node-wise transformation (MLP)}"] \\
\text{Initial Node Embeddings} \arrow[d, "\text{Message Passing}"] \arrow[d, loop right] \\
\text{Final Node Embeddings} \arrow[d, "\text{Pooling}"] \\
\text{Graph Embedding (Optional)} \arrow[d, "\text{Transformation (MLP)}"] \\
\text{Final Predictions}
\end{tikzcd}$$



Message passing layer:

- Messages

$$\mathbf{m}_{ij} = f_1\left(\mathbf{s}_i, \mathbf{s}_j, \mathbf{a}_{ij}\right),$$

- Aggregate + node updates

$$\mathbf{s}'_i := f_2\left(\mathbf{s}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}\right),$$

or equivalently,

$$\mathbf{s}'_i := f_2\left(\mathbf{s}_i, \sum_{j \in \mathcal{N}(i)} f_1\left(\mathbf{s}_i, \mathbf{s}_j, \mathbf{a}_{ij}\right)\right),$$

where f_1, f_2 are message and updating functions (MLPs).

FAR BEYOND

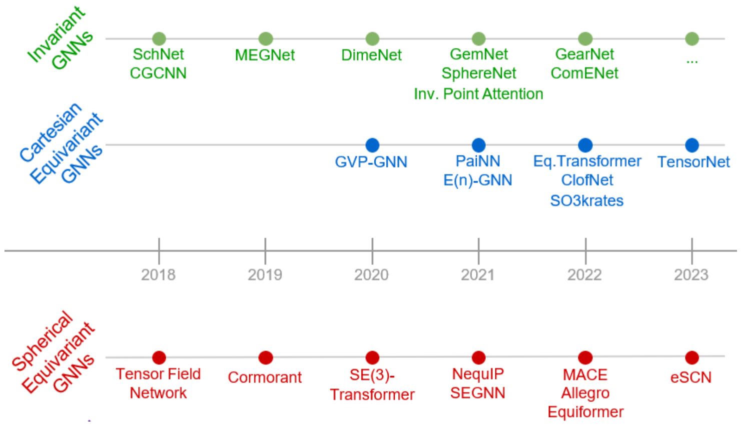
Stony Brook University

Geometric Message Passing

For geometric message passing, we condition on geometries. As an illustrative example, assume we have the coordinate information and let a_{ij} contain geometric information, we can have the following message passing schemes:

$$\mathbf{m}_{ij} = f_1\left(\mathbf{s}_i, \mathbf{s}_j, x_j - x_i\right)$$

To make it equivariant (invariant) to E(3), there are in general two directions: Scalarization and Using Steerable Tensor Features. We term them as invariant GNNs (scalarization) and equivariant GNNs (Tensor Operations). *Invariant GNNs constraint the geometric information that can be utilized, while the other constraints the model operations.*



FAR BEYOND

Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Stony Brook University

Invariant GNNs

1. Using relative distances (1-hop; body order 2): $\mathbf{m}_{ij} = f_1\left(\mathbf{s}_i, \mathbf{s}_j, d_{ij}\right)$, where $d_{ij} = ||x_j - x_i||$. This is $E(3)$-equivariant, but we limit the expressivity of the model as we cannot distinguish different local geometries. We cannot distinguish two local neighbourhoods apart using the unordered set of distances only.
2. Using relative distances and bond angles (2-hop; body order 3): $\mathbf{m}_{ij} = f_1\left(s_i, s_j, d_{ij}, \sum_{k \in \mathcal{N}_j \backslash \{i\}} f_3\left(s_j, s_k, d_{ij}, d_{jk}, \measuredangle ijk\right)\right)$. This is $E(3)$-equivariant, but again we limit the expressivity of the model.
3. Using relative distances, bond angles, and torsion angles (3-hop; body order 4):

$$\boldsymbol{m}_{ij} = f_1\left(s_i, s_j, d_{ij}, \sum_{k \in \mathcal{N}_j \backslash \{i\}, l \in \mathcal{N}_k \backslash \{i,j\}} f_3\left(s_k, s_l, d_{kl}, d_{ij}, d_{jk}, \measuredangle ijk, \measuredangle jkl, \measuredangle ijkl\right)\right).$$

This is $E(3)$-equivariant and complete.

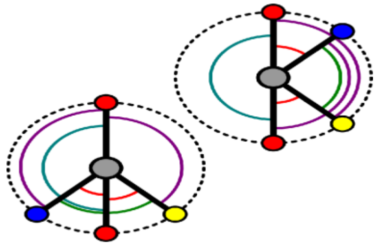


Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Invariant GNNs

In summary, invariant GNNs update latent representations by scalarizing local geometry information. This is efficient, and we can achieve invariance with simple MLP without specific constraints on the operations or activations we can take.

Pros:
- Simple usage of non-linearities on many-body scalars.
- Great performance on some use-cases (e.g. GemNet on OC20).

Cons:
- Scalability of scalar’s pre-computation. The accounting of higher-order tuples is expensive.
- Making invariant predictions may still require solving equivariant sub-tasks.
- May lack generalization capabilities (equivariant tasks, multi-domain).

GemNet (Torsion) in theory is complete; however, it requires a complete graph and a certain discretization scheme to be universal*.

So far, the precise body order of scalars at which all geometric graphs can be uniquely identified remains an open question *.

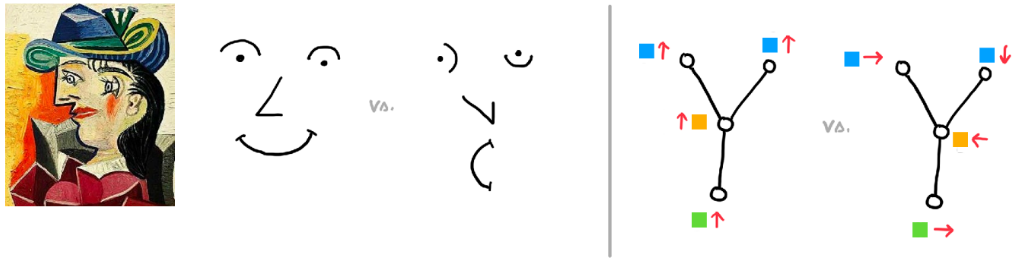
*: Quote from A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Equivariant GNNs

- In invariant GNNs, invariants are 'fixed' prior to message passing.
- Equivariant GNNs build up invariants 'on the go' during message passing. More layers of message passing can lead to more complex invariants being built up. Furthermore, equivariant quantities remain available.

Intuition: **The Picasso Problem**

Making invariant predictions may still require solving equivariant sub-tasks



Relative orientation of eyes, nose, mouth is important (**orientation of sub-graphs w.r.t. one another**), not just their presence (**invariant quantities**)!

Image Credit: A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems, Duvel et al.. and their presentation slides

Scalar-Vector GNNs

- ❑ In invariant GNNs, we work with only scalars $f\ (s_1, s_2, \dots, s_n)$.
- ❑ In equivariant GNNs, we work with vectors $f\ (s_1, s_2, \dots s_n, v_1, \dots, v_m)$.

Example: “Scalar-vector” GNNs

Scalar message:

$$\mathbf{m}_i := \quad f_1\left(\mathbf{s}_i, \|\mathbf{v_i}\|\right) + \sum_{j \in \mathcal{N}_i} f_2\left(\mathbf{s}_i, \mathbf{s}_j, \left\|\vec{\boldsymbol{x}}_{ij}\right\|, \left\|\boldsymbol{v}_j\right\|, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_j, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_i, \mathbf{v}_i \cdot \mathbf{v}_j\right)$$

Vector message:

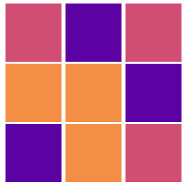
$$\begin{aligned}\vec{\mathbf{m}}_i := f_3\left(\mathbf{s}_i, \|\mathbf{v_i}\|\right) \odot \mathbf{v}_i &+ \sum_{j \in \mathcal{N}_i} f_4\left(\mathbf{s}_i, \mathbf{s}_j, \left\|\vec{\boldsymbol{x}}_{ij}\right\|, \left\|\boldsymbol{v}_j\right\|, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_j, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_i, \mathbf{v}_i \cdot \mathbf{v}_j\right) \odot \mathbf{v}_j \\ &+ \sum_{j \in \mathcal{N}_i} f_5\left(\mathbf{s}_i, \mathbf{s}_j, \left\|\vec{\boldsymbol{x}}_{ij}\right\|, \left\|\boldsymbol{v}_j\right\|, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_j, \vec{\boldsymbol{x}}_{ij} \cdot \mathbf{v}_i, \mathbf{v}_i \cdot \mathbf{v}_j\right) \odot \vec{\boldsymbol{x}}_{ij},\end{aligned}$$

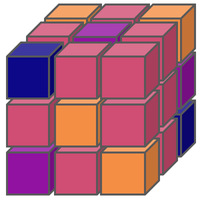
Tensors

The high-level ideas in equivariant GNNs is that we keep track of the "types" of the objects and apply equivariant operations. As of now, we are constrained to have only scalar or vector features. *What about tensors?*

- A tensor is a multi-dimensional array with directional information.
- A rank-n Cartesian tensor T can be viewed as a multidimensional array with n indices, i.e., $(T)_{\mathrm{i}_1\,\mathrm{i}_2\cdots\,\mathrm{i}_\mathrm{n}}$ with $i_k \in \{1,2,3\}$ for $\forall k \in \{1, \cdots, n\}$. Furthermore, each index of $(T)_{i_1 i_2 \cdots i_n}$ transforms independently as a vector under rotation.

Rank 1


Rank 2


Rank 3

Tensor Products

A vector v in 3D Euclidean space $\mathbb{R}^3$ can be expressed in the familiar Cartesian coordinate system in the standard basis e_x, e_y, e_z, and coordinates A_x, A_y, A_z, where $\mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ $\mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ $\mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$.

When you perform the tensor (or outer) product of two vectors in $\mathbb{R}^3$, you obtain a matrix (or a rank2 tensor). If you have two vectors $\mathbf{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$ and $\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$, their tensor product $\mathbf{u} \otimes \mathbf{v}$ is given by:

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \otimes \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} u_x v_x & u_x v_y & u_x v_z \\ u_y v_x & u_y v_y & u_y v_z \\ u_z v_x & u_z v_y & u_z v_z \end{pmatrix}.$$

This is easier to understand or remember with the definition of outer product of two functions: $(f \otimes g)(x, y) = f(x)g(y)$.

Stony Brook University

Basis of Tensors

In terms of basis, if $\mathbf{u}$ and $\mathbf{v}$ are expressed in the standard basis $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$, the resulting tensor product $\mathbf{u} \otimes \mathbf{v}$ can be viewed as a linear combination of the outer products of the basis vectors:

$$\mathbf{u} \otimes \mathbf{v} = u_x v_x (\mathbf{e}_x \otimes \mathbf{e}_x) + u_x v_y (\mathbf{e}_x \otimes \mathbf{e}_y) + u_x v_z (\mathbf{e}_x \otimes \mathbf{e}_z) + u_y v_x (\mathbf{e}_y \otimes \mathbf{e}_x) + u_y v_y (\mathbf{e}_y \otimes \mathbf{e}_y) + u_y v_z (\mathbf{e}_y \otimes \mathbf{e}_z) + u_z v_x (\mathbf{e}_z \otimes \mathbf{e}_x) + u_z v_y (\mathbf{e}_z \otimes \mathbf{e}_y) + u_z v_z (\mathbf{e}_z \otimes \mathbf{e}_z)$$

The basis are given by:

$$\mathbf{e}_x \otimes \mathbf{e}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{e}_x \otimes \mathbf{e}_y = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{e}_x \otimes \mathbf{e}_z = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{e}_y \otimes \mathbf{e}_x = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ldots$$

FAR BEYOND

Stony Brook University

Change of Basis

Let $\vec{v} \in V$ be a vector. Fix a basis $\{e_1, \ldots, e_n\}$, whence you have $\vec{v} = \sum_{i=1}^{n} e_i v^i = (e_1, \ldots e_n) \cdot (v^1, \ldots, v^n)^T$.

Then a change of basis is equivalent to the choice of an invertible $n \times n$ matrix M via

$\vec{v} = (e_1, \ldots, e_n)\, M M^{-1} (v^1, \ldots, v^n)^T = (\epsilon_1, \ldots, \epsilon_n) \cdot (\nu^1, \ldots, \nu^n)^T$, where $\{\epsilon_1, \ldots, \epsilon_n\}$ is the new basis and $\nu^1, \ldots, \nu^n$ are the new coefficients.

Example:

To change the basis of the vector $\mathbf{v} = [1, 2, 3]$ from the standard basis to the new basis $(1, 1, 0)$, $(0, 1, 1)$, $(0, 0, 1)$, we need to express $\mathbf{v}$ in terms of the new basis vectors.

1. Construct M: For new basis vectors $(1, 1, 0)$, $(0, 1, 1)$, and $(0, 0, 1)$, $M = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$.
2. $M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}$, $\mathbf{v}_{\text{new}} = B^{-1}\mathbf{v} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$.

FAR BEYOND

Stony Brook University

Representations

A representation $\rho : G \rightarrow GL(V)$ is a group homomorphism from G to the general linear group $GL(V)$. That is, $\rho(g)$ is a linear transformation parameterized by group elements $g \in G$ that transforms some vector $\mathbf{v} \in V$ (e.g. an image or a tensor) such that

$$\rho\left(g'\right) \circ \rho(g)[\mathbf{v}] = \rho\left(g' \cdot g\right)[\mathbf{v}].$$

Example: The representation of $SO(3)$ acting on a geometric 3D vector is a 3×3 orthogonal matrices with determinant 1.

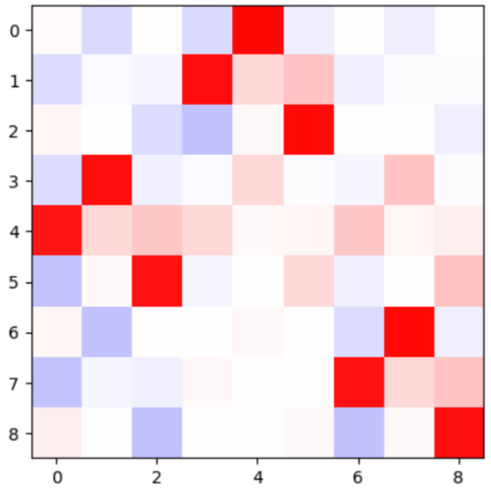
Irreducible Representation: A representation $\rho : G \rightarrow GL(V)$ is said to be irreducible if there are no proper non-zero subspaces W of V that are invariant under all group actions, i.e., $\rho(g)W \subseteq W$ for all $g \in G$. In other words, V cannot be split into smaller subspaces that are individually invariant under the group action.

Decomposition: If a representation is reducible, it can be decomposed into a direct sum of irreducible representations (irreps). A block diagonal matrix can be written as the direct sum of the matrices that lie along the diagonal. In contrast, an irreducible representation cannot be decomposed further in this way.

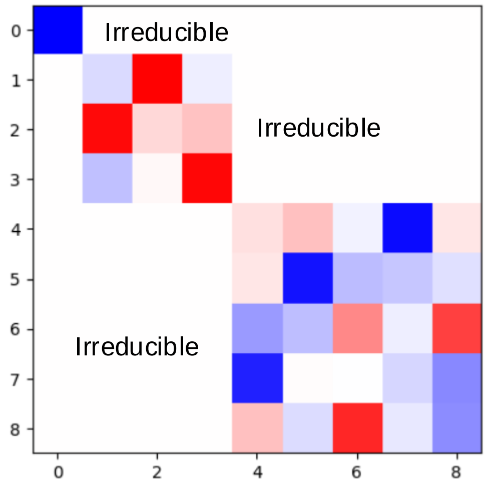
Block-Diagonal Form: If a representation is reducible (i.e., can be decomposed into simpler parts), it can be rewritten (decomposed) into block-diagonal form (each of the blocks are the simpler parts, i.e., irreps).

FAR BEYOND

Representations



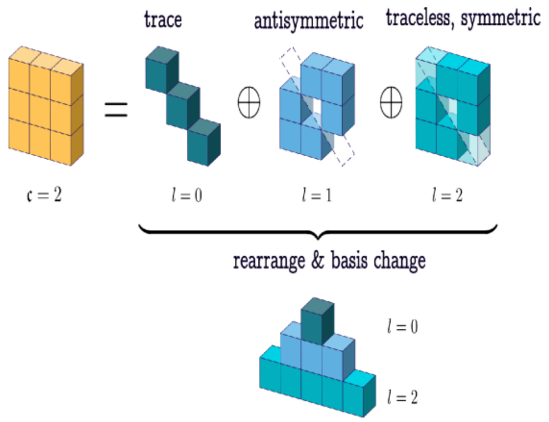
Reducible
Representation



Decomposed
into Irreps

Decomposition of Rank 1 Cartesian Tensors

Decompose the Cartesian tensor space into a direct sum of some subspaces:



Why?

- Each subspace acts independently under the actions of the rotation group (irreducible representations).
- Tensors in each subspace have the same "type".
- Like scalar-vector networks, we apply equivariant operations to each type.

Stony Brook University

Decomposition of Rank 2 Cartesian Tensors

$$T^{[2]} = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix}$$

$$= \frac{\sqrt{3}}{3}(T_{xx}+T_{yy}+T_{zz}) \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \qquad (l=0)$$

$$+ \begin{bmatrix} & \frac{T_{xy}-T_{yx}}{2} & \frac{T_{xz}-T_{zx}}{2} \\ \frac{T_{yx}-T_{xy}}{2} & & \frac{T_{yz}-T_{zy}}{2} \\ \frac{T_{zx}-T_{xz}}{2} & \frac{T_{zy}-T_{yz}}{2} & \end{bmatrix} \qquad (l=1)$$

$$+ \begin{bmatrix} T_{xx} & \frac{T_{xy}+T_{yx}}{2} & \frac{T_{xz}+T_{zx}}{2} \\ \frac{T_{yx}+T_{xy}}{2} & T_{yy} & \frac{T_{yz}+T_{zy}}{2} \\ \frac{T_{zx}+T_{xz}}{2} & \frac{T_{zy}+T_{yz}}{2} & T_{zz} \end{bmatrix} - \frac{\sqrt{3}}{3}(T_{xx}+T_{yy}+T_{zz}) \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \qquad (l=2)$$

$$= \overbrace{\frac{\sqrt{3}}{3}(T_{xx}+T_{yy}+T_{zz})}^{\lambda_1} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \qquad (l=0)$$

$$+ \overbrace{\frac{\sqrt{2}}{2}(T_{yz}-T_{zy})}^{\lambda_2} \begin{bmatrix} & & \\ & & \frac{\sqrt{2}}{2} \\ & -\frac{\sqrt{2}}{2} & \end{bmatrix} + \overbrace{\frac{\sqrt{2}}{2}(T_{zx}-T_{xz})}^{\lambda_3} \begin{bmatrix} & & -\frac{\sqrt{2}}{2} \\ & & \\ \frac{\sqrt{2}}{2} & & \end{bmatrix} + \overbrace{\frac{\sqrt{2}}{2}(T_{xy}-T_{yx})}^{\lambda_4} \begin{bmatrix} & \frac{\sqrt{2}}{2} & \\ -\frac{\sqrt{2}}{2} & & \\ & & \end{bmatrix} \qquad (l=1)$$

$$+ \overbrace{\frac{\sqrt{2}}{2}(T_{xz}+T_{xz})}^{\lambda_5} \begin{bmatrix} & & \frac{\sqrt{2}}{2} \\ & & \\ \frac{\sqrt{2}}{2} & & \end{bmatrix} + \overbrace{\frac{\sqrt{2}}{2}(T_{xy}+T_{yx})}^{\lambda_6} \begin{bmatrix} & \frac{\sqrt{2}}{2} & \\ \frac{\sqrt{2}}{2} & & \\ & & \end{bmatrix} + \overbrace{\frac{\sqrt{6}}{6}(2\cdot T_{yy}-T_{xx}-T_{zz})}^{\lambda_7} \begin{bmatrix} -\frac{\sqrt{2}}{2} & & \\ & \sqrt{2} & \\ & & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

$$+ \overbrace{\frac{\sqrt{2}}{2}(T_{yz}+T_{zy})}^{\lambda_8} \begin{bmatrix} & & \\ & & \frac{\sqrt{2}}{2} \\ & \frac{\sqrt{2}}{2} & \end{bmatrix} + \overbrace{\frac{\sqrt{2}}{2}(T_{zz}-T_{xx})}^{\lambda_9} \begin{bmatrix} -\frac{\sqrt{6}}{2} & & \\ & & \\ & & \frac{\sqrt{6}}{2} \end{bmatrix} \qquad (l=2)$$

Inner product: $\vec{v}\cdot\vec{w} = v_x w_x + v_y w_y + v_z w_z = \sqrt{3}\cdot\lambda_1$

Cross product: $\vec{v}\times\vec{w} = \begin{bmatrix} v_y w_z - v_z w_y \\ v_z w_x - v_x w_z \\ v_x w_y - v_y w_x \end{bmatrix} = \sqrt{2}\cdot\begin{bmatrix} \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}$

- Inner product is *invariant*
- Cross product is *equivariant*
- Scaling does not change invariance/equivariance
- Decomposition to irreps is not unique

FAR BEYOND

Stony Brook University

Rotation of Spherical Tensor

To summarize, we have seen that the 9-dimensional rank-2 Cartesian tensor can be decomposed into a 1D, 3D and 5D parts which correspond to the $l = 0, 1, 2$ irreps respectively

$$\mathbf{3} \otimes \mathbf{3} = \mathbf{1} \oplus \mathbf{3} \oplus \mathbf{5}.$$

These tensors are called spherical tensors.

The $\ell = 0$ part is invariant, $\ell = 1$ part rotates as a normal vector in $\mathbb{R}^3$. For types of higher l, the transformation behaviour under rotations is also known and given by the so-called Wigner D-matrices. The D here stands for Darstellung, the German word for representation. The Wigner D-matrix is how one can represent a rotation in the $2l + 1$ dimensional space of a degree l spherical tensor.

$$\vec{T}^{(l)} \rightarrow \mathcal{D}^{(l)}(\mathbf{R})\vec{T}^{(l)}$$

where $\vec{T}^{(l)} \in \mathbb{R}^{(2\ell+1)}$ and $D^{\ell}(R) \in \mathbb{R}^{(2\ell+1)\times(2\ell+1)}$.

FAR BEYOND

Stony Brook University

Tensor Products of Spherical Tensors

Unfortunately, the tensor product of two spherical tensors $\vec{S}^{(l_1)}$ and $\vec{T}^{(l_2)}$ is generally not a spherical tensor anymore.

Example: Suppose we have a $\ell_1 = 1$ (3 elements) tensor and a $\ell_2 = 2$ (5 elements), the tensor product would have $3 \times 5 = 15$ elements. However, this may not be a $\ell_3 = 7$ (15 elements) tensor.

However, we can decompose the product $\vec{S}^{(l_1)} \otimes \vec{T}^{(l_2)}$ back into spherical tensors.

As a rule, the $(l_1 l_2)$-dimensional tensor product of two spherical tensors of ranks l_1 and l_2 decomposes into:

$$l_1 \otimes l_2 = |l_1 - l_2| \oplus |l_1 - l_2 + 1| \oplus \cdots \oplus (l_1 + l_2 - 1) \oplus (l_1 + l_2)\,.$$

This means the $l_1 l_2$-dimensional product decomposes into exactly one spherical tensor for each rank between the absolute difference $|l_1 - l_2|$ and the sum $l_1 + l_2$.

Example: $|1 - 2| = 1$ and $1 + 2 = 3$. The 15 elements in the tensor product can be decomposed into a $l = 1$ (3 elements) tensor, a $l = 2$ (5 elements) tensor, and a $l = 3$ (7 elements). In some not so rigorous notation:$1 \otimes 2 = 1 \oplus 2 \oplus 3$.

FAR BEYOND

CG Decomposition

The coefficients of the decomposition are given by the Clebsch-Gordan coefficients.

Example: Suppose we with to get the $l = 1$ tensor resulted from the tensor product of $\vec{S}^{(l_1)} \otimes \vec{T}^{(l_2)}$. Each of these three elements is a weighted sum of the 15 elements. So in total, we have $3 \times 5 \times 3 = 45$ coefficients. We denote this change of basis weights by $C^{(l_1,l_2,l_3)}_{(m_1,m_2,m_3)}$, where $-\ell_i \leq m_i \leq \ell_i$.

For example: $C^{(l_1=1,l_2=2,l_3=1)}_{(m_1=1,m_2=2,m_3=1)}$ means the coefficient of $t_1 \times s_2$ in order to get u_1 in the resulting tensor (We have 15 coefficients for u_1). That is

$$u_1 = \sum_{i=-1}^{1} \sum_{j=-2}^{2} C^{(l_1=1,l_2=2,l_3=1)}_{(m_1=i,m_2=j,m_3=1)} t_i s_j$$

$$u_2 = \sum_{i=-1}^{1} \sum_{j=-2}^{2} C^{(l_1=1,l_2=2,l_3=1)}_{(m_1=i,m_2=j,m_3=2)} t_i s_j$$

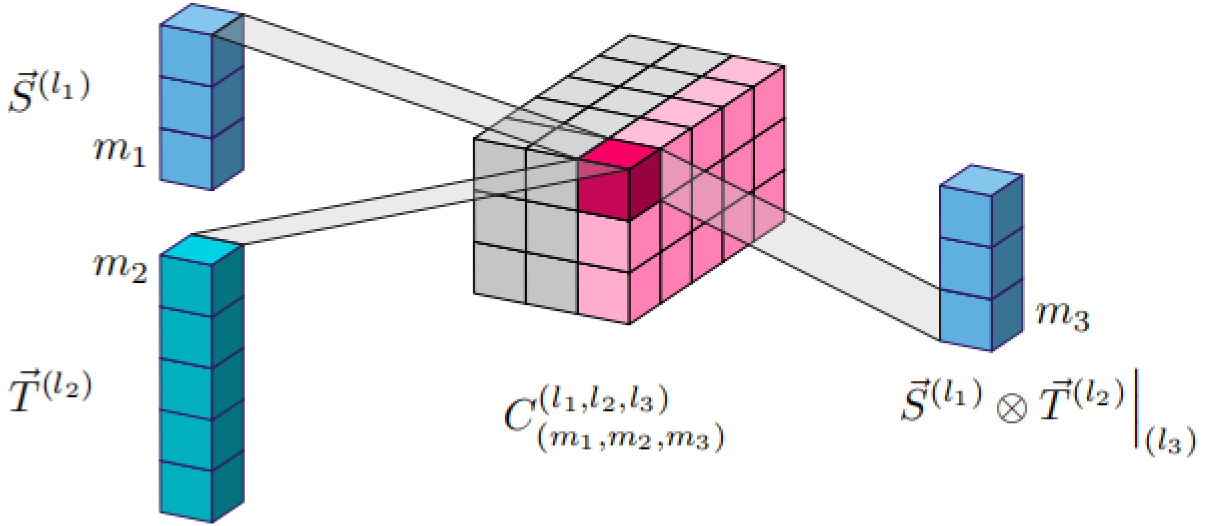
$$u_3 = \sum_{i=-1}^{1} \sum_{j=-2}^{2} C^{(l_1=1,l_2=2,l_3=1)}_{(m_1=i,m_2=j,m_3=3)} t_i s_j.$$

Similarly, $C^{(l_1=1,l_2=2,l_3=2)}_{(m_1,m_2,m_3)}$ will give the resulting $l = 2$ tensor, etc..

$$\mathbf{T} = \begin{pmatrix} t_{-1} \\ t_0 \\ t_1 \end{pmatrix}, \mathbf{S} = \begin{pmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{pmatrix}, \mathbf{T} \otimes \mathbf{S} = \left(\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ \hline u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ \hline u_9 \\ u_{10} \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{14} \\ u_{15} \end{array}\right)$$

Stony Brook University

CG Decomposition



$\vec{S}^{(l_1)}$
m_1
m_2
$\vec{T}^{(l_2)}$
$C^{(l_1,l_2,l_3)}_{(m_1,m_2,m_3)}$
m_3
$\vec{S}^{(l_1)} \otimes \vec{T}^{(l_2)}\Big|_{(l_3)}$

FAR BEYOND

Spherical Harmonics

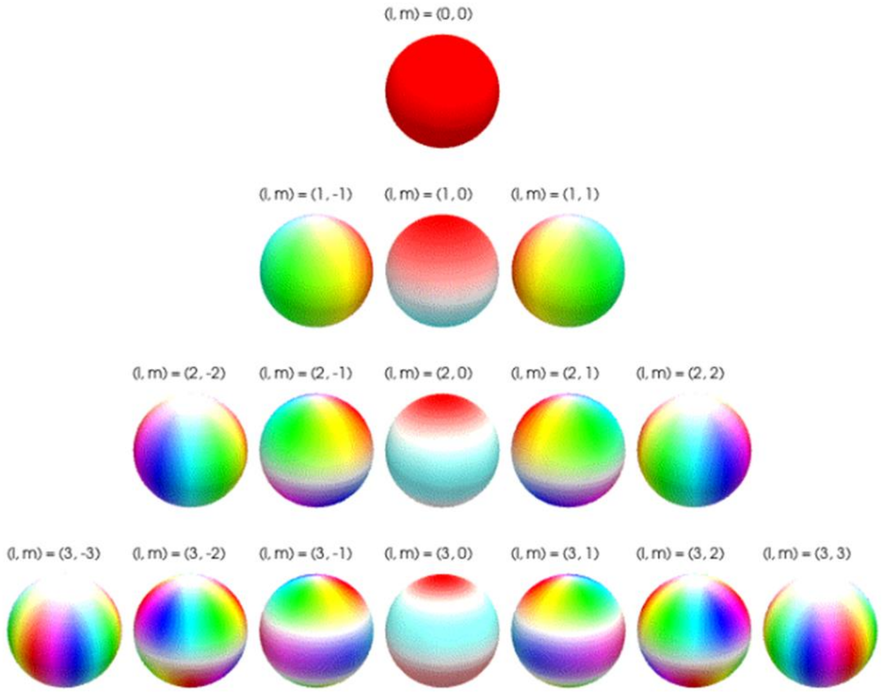
Real spherical harmonics $Y_l^m(\theta, \phi) : S^2 \to \mathbb{R}$ are real-valued functions defined on the surface of a sphere.

$$Y_\ell^m(\theta, \varphi) = (-1)^m \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)}} P_\ell^m(\cos\theta) e^{im\varphi}$$

Each real spherical harmonic is indexed by two integers: l (degree) and m (order), where $l \geq 0$ and $-l \leq m \leq l$. They are used as an orthonormal basis for representing functions on the sphere. Under fairly general condition (square-integrable on the sphere), any function can be written as a linear combination of spherical harmonics as follows:

$$f(\theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_\ell^m Y_\ell^m(\theta, \varphi).$$

Spherical Harmonics



Spherical Tensors From Spherical Harmonics

Generally, we can stack all the values from the degree-l spherical harmonics together to get a order-ℓ spherical tensor.

Example: Given a 3D point $v = (x, y, z)$, we can write it as a radial part $||v||$ and a directional part $v/||v||$. The directional part is now defined on S^2, write it as (θ, ϕ). We can get a order-1 tensor with spherical harmonics as: $V^{(1)} = \begin{pmatrix} Y_{l=1}^{m=-1}(\theta,\phi) \\ Y_{l=1}^{m=0}(\theta,\phi) \\ Y_{l=1}^{m=1}(\theta,\phi) \end{pmatrix}$.

For simplicity, we can rewrite (real) spherical harmonics as a vector-valued function for order-ℓ. That is $Y^{\ell}(\cdot) : \mathbb{R}^3 \to \mathbb{R}^{2\ell+1}$ maps an input 3D vector to a $(2\ell + 1)$-dimensional vector representing the coefficients of order- ℓ spherical harmonics bases.

Spherical harmonics function is equivariant to order-ℓ rotations, or so-called order-ℓ $SO(3)$ transformations:

$$Y^{\ell}(R\boldsymbol{c}) = D^{\ell}(R)Y^{\ell}(\boldsymbol{c}),$$

where $\boldsymbol{c}$ is a 3D point.

Note:

Spherical harmonics are a set of orthonormal functions defined on the surface of a sphere ($[0, \pi] \times [0, 2\pi]$) just like Fourier Basis. In fact, Fourier basis is called circular harmonics.

Stony Brook University

Preliminary: Spherical Harmonics

We fix the terms we use:

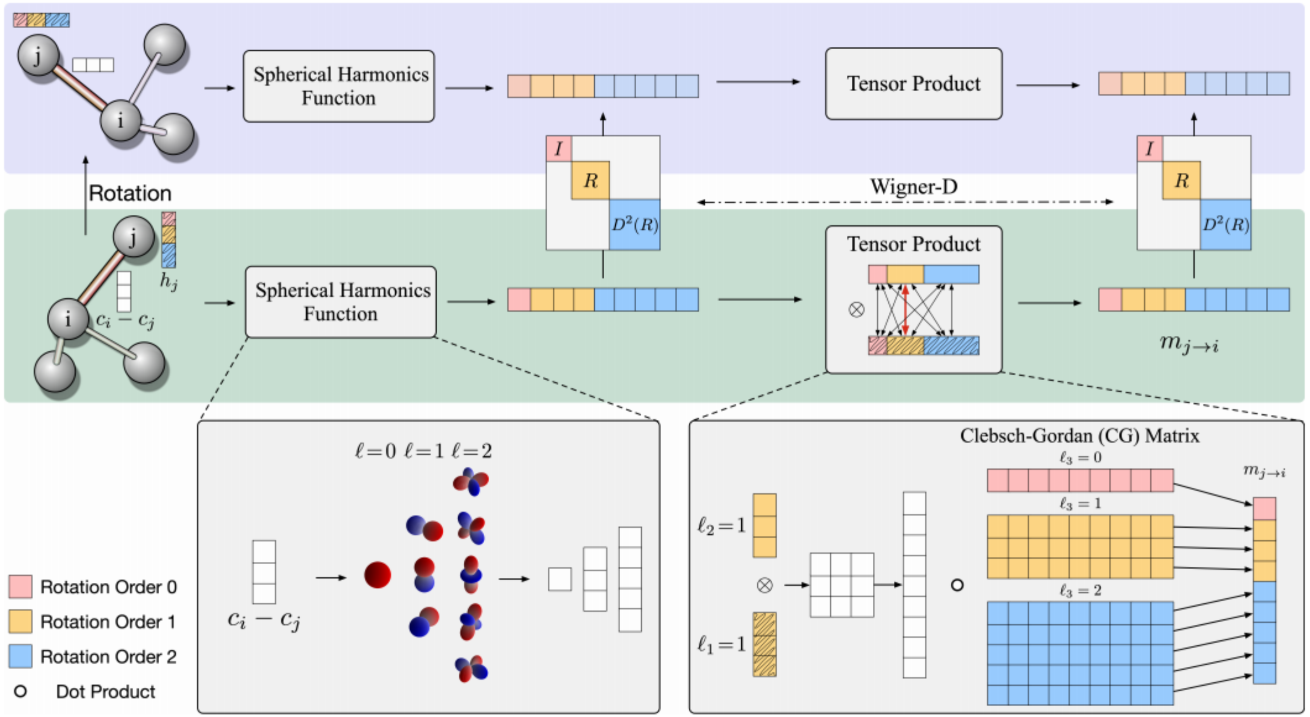
1. **Rank** k Cartesian Tensors: $T^{[k]}$
2. **Order**-ℓ Spherical Tensors: $T^{(l)}$
3. Spherical Harmonics with **degree** ℓ and **order** m: Y_l^m
4. **Order**-ℓ Spherical Harmonics Function that gives the **Order**-ℓ Spherical Harmonics Coefficients: $Y^{\ell}(\cdot)$

It is weird that they use order for degree, but this is the convention.

FAR BEYOND

Stony Brook University

Equivariant Interactions via Spherical TP



FAR BEYOND

Image Credit: Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems, Xuan Zhang el al.