Ai4science

Wu Hualong

HARBIN INSTITUTE OF TECHNOLOGY,SHEN ZHEN

Outline

background

TFN-Tensor field networks

SE(3)-Transformer

eSCN

background

problem

- We focus on continuous $SE(3)$ transformations in 3D structures of chemical compounds, including translations and 3D rotations, where $SE(3)$ stands for the special Euclidean group in 3D space.
- Let $C = [\mathbf{c}_1, ..., \mathbf{c}_n] \in \mathbb{R}^{3\times n}$ be the coordinate matrix of a 3D point cloud with n nodes
- $f : \mathbb{R}^{3\times n} \to \mathbb{R}^{2\ell+1}$ be a function mapping coordinate matrices to $(2\ell + 1)$-dimensional property vector that is $SE(3)$ equivariant with order ℓ

order-ℓ equivariance requires f to be:

$$f\left(RC + \mathbf{t}1^T\right) = D^\ell(R)f(C) \tag{1}$$

- $\mathbf{t} \in \mathbb{R}^3$ is the translation vector, $R \in \mathbb{R}^{3\times 3}$ is the rotation matrix
- $D^\ell(R) \in \mathbb{R}^{(2\ell+1)\times(2\ell+1)}$ is the (real) Wigner-D matrix of R

Tensor and Wigner-D matrix

Tensor: What characterizes a tensor is the way it transform under rotation.A type-ℓ vector is $2\ell + 1$ dimension.
Wigner-D matrices are high-order rotation matrices for 3D rotation transformation in physics.,they map elements of $SO(3)$ to $(2\ell + 1) \times (2\ell + 1)$-dimensional matrices.

- A type-0 vector $v \in V_0$ is just a scalar that trivially transforms by a 1×1 dim "matrix".

$$\mathbf{D}^{(0)}(\mathbf{R})v = 1v = v$$

- A type-1 vector $\mathbf{v} \in \mathbf{V}_1$ is a 3D vector.(e.g. velocity,force,displacement) that transforms directly via the rotation matrix $\mathbf{R} \in SO(3)$

$$\mathbf{D}^{(1)}(\mathbf{R})\mathbf{v} = \mathbf{R}\mathbf{v}$$

Tensor Product and Clebsch-Gordan Coefficients

Mathematically, the tensor product is defined to represent bilinear maps,consider two 3D vectors.
let $f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ be a bilinear map,All such bilinear maps can be written as:

$$f(\mathbf{x}, \mathbf{y}) = \sum_{ij} c_{ij} x_i y_j$$

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \otimes \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 x_2 & x_1 y_2 & x_1 z_2 \\ y_1 x_2 & y_1 y_2 & y_1 z_2 \\ z_1 x_2 & z_1 y_2 & z_1 z_2 \end{bmatrix}$$

The Clebsch-Gordan coefficients are c_{ij} that can ensure equivariance.

Clebsch-Gordan Tensor Product

Tensor products can interact with different type-L vectors.The tensor product denoted as $\otimes$ uses Clebsch-Gordan coefficients to combine type-L_1 vector $f^{(L_1)}$ and type-L_2 vector $g^{(L_2)}$ and produces type-L_3 vector $h^{(L_3)}$:

$$h_{m_3}^{(L_3)} = (f^{(L_1)} \otimes g^{(L_2)})_{m_3} = \sum_{m_1=-L_1}^{L_1} \sum_{m_2=-L_2}^{L_2} C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)} f_{m_1}^{(L_1)} g_{m_2}^{(L_2)} \tag{2}$$

where m_1 denotes order and refers to the m_1-th element of $f^{(L_1)}$,Clebsch-Gordan coefficients $C_{(L_1,m_1)(L_2,m_2)}^{(L_3,m_3)}$ are non-zero only when $|L_1 - L_2| \leq L_3 \leq |L_1 + L_2|$.

Clebsch-Gordan Tensor Product

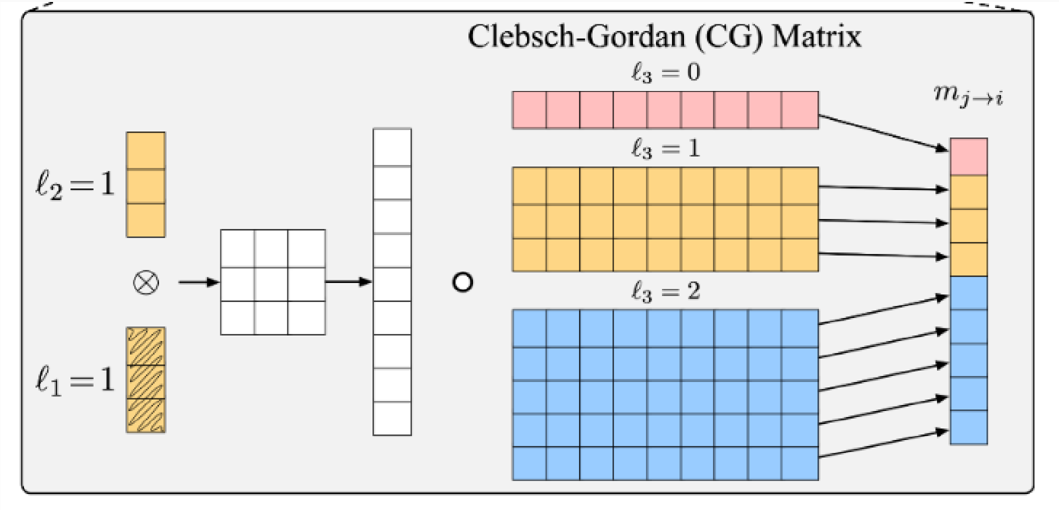


Figure 1: Clebsch-Gordan Tensor Product

Tensor Product

We call each distinct non-trivial combination of $L_1 \otimes L_2 \rightarrow L_3$ a path. Each path is independently equivariant, and we can assign one learnable weight to each path in tensor products, which is similar to typical linear layers.

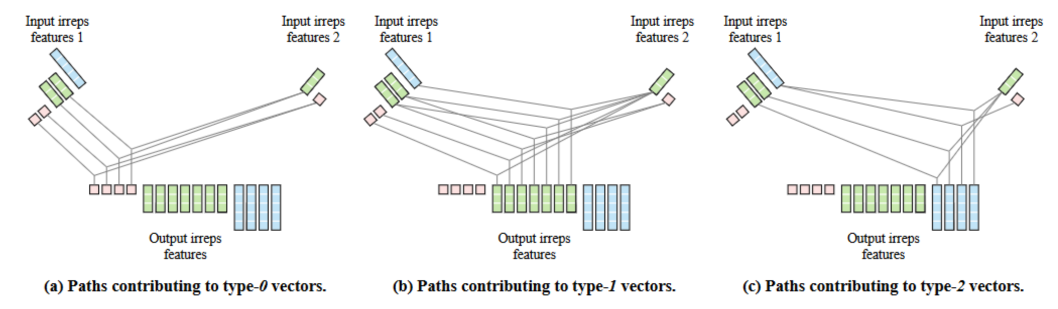


Figure 2: Tensor Product

Spherical Harmonics.

Euclidean vectors $\vec{r}$ in $\mathbb{R}^3$ can be projected into type-L vectors $f^{(L)}$ by using spherical harmonics (SH)

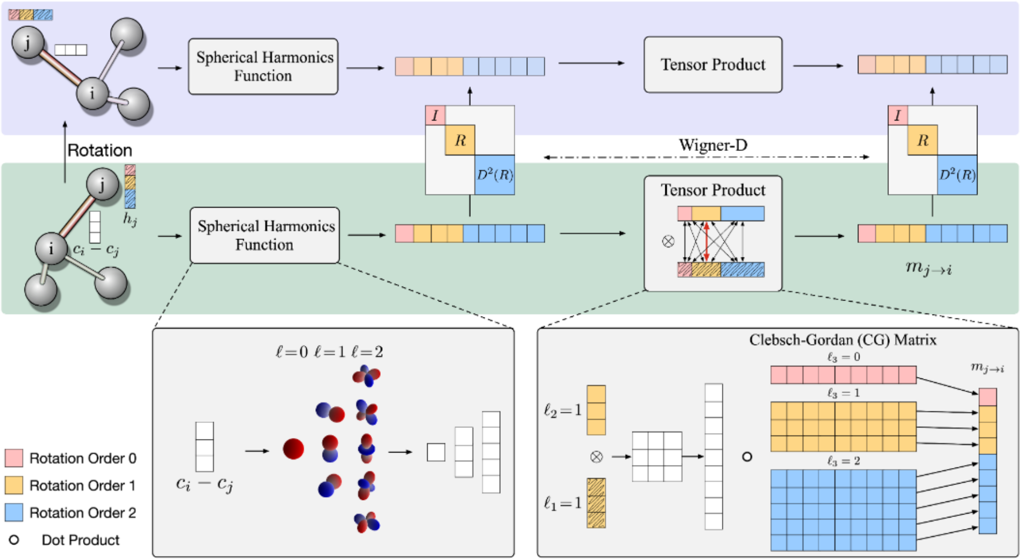
$$Y^{(L)}: f^{(L)} = Y^{(L)}(\frac{\vec{r}}{||\vec{r}||}). \tag{3}$$

SH are $E(3)$-equivariant with:

$$D_L(g)f^{(L)} = Y^{(L)}(\frac{D_L(g)\vec{r}}{||D_L(g)\vec{r}||}). \tag{4}$$

SH of relative position $\vec{r}_{ij}$ generates the first set of irreps features. Equivariant information propagates to other irreps features through equivariant operations like tensor products.

Equivariant Data Interactions via Tensor Product



TFN-Tensor field networks

Tensor field network layers

input embedding: every vertex(atom) is embedded to the a representation: V_{acm}

- V_{acm} has many irreducible representations: V^{ℓ}_{acm}, ℓ is the rotation order,and can be $0, 1, ..., \ell_{max}$
- there are multiple instances of each l-rotation-order irreducible representations,we call it channels

filters: For the filters to be rotation-equivariant, restrict them to the following form:

$$F^{(l_f,l_i)}_{cm}(\vec{r}) = R^{(l_f,l_i)}_{c}(r)Y^{(l_f)}_{m}(\hat{r}) \tag{5}$$

- $\ell_i and \ell_f$ are non-negative integers corresponding to the rotation order of the input and the filter
- $R^{(l_f,l_i)}_{c}(r) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ are learned functions

Layer definition

A given input inhabits one representation, a filter inhabits another, and together these produce outputs at possibly many rotation orders.

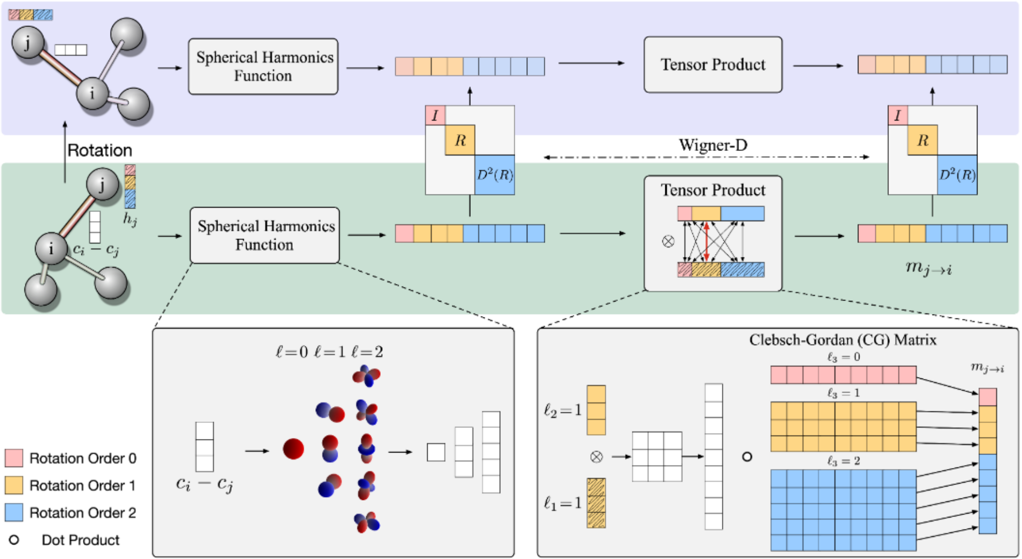
Layer definition:

$$\mathcal{L}_{acm_o}^{(l_o)}\left(\vec{r}_a, V_{acm_i}^{(l_i)}\right) := \sum_{m_f, m_i} C_{(l_f, m_f)(l_i, m_i)}^{(l_o, m_o)} \sum_{b \in S} F_{cm_f}^{(l_f, l_i)}(\vec{r}_{ab}) V_{bcm_i}^{(l_i)} \tag{6}$$

we get message:

$$M_a^{l_o} = \sum_{b \in S} TP_{l_i, l_f}^{l_o}(F_c^{(l_f, l_i)}(\vec{r}_{ab}), V_{bc}^{(l_i)}) \tag{7}$$

Layer definition



Tensor Field Network

Self-interaction: Self-interaction layers are analogous to 1x1 convolutions:

$$\sum_{c'} W^{(l)}_{cc'} V^{(l)}_{ac'm}$$

Nonlinearity:

$$\eta^{(0)}\left(V^{(0)}_{ac} + b^{(0)}_{c}\right) \quad \text{and} \quad \eta^{(l)}\left(\|V\|^{(l)}_{ac} + b^{(l)}_{c}\right) V^{(l)}_{acm} \quad \text{where} \quad \|V\|^{(l)}_{ac} := \sqrt{\sum_{m} |V^{(l)}_{acm}|^2}$$

SE(3)-Transformer

TFN in SE(3)

Each $\mathbf{f}_j$ is a concatenation of vectors of different types for node j, where a sub-vector of type-ℓ is written $\mathbf{f}_j^\ell$

The type-ℓ output of the TFN layers at position $\mathbf{x}_i$ is

$$\mathbf{f}_{\mathrm{out},i}^{\ell} = \sum_{k\geq 0} \underbrace{\int \mathbf{W}^{\ell k}(\mathbf{x}' - \mathbf{x}_i)\mathbf{f}_{\mathrm{in}}^{k}(\mathbf{x}')\mathrm{d}\mathbf{x}'}_{k\to\ell \text{ convolution}} = \sum_{k\geq 0}\sum_{j=1}^{n} \underbrace{\mathbf{W}^{\ell k}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}_{\mathrm{in},j}^{k}}_{\text{node } j\to \text{ node } i \text{ message}}, \tag{8}$$

furthermore:

$$\mathbf{W}^{\ell k}(\mathbf{x}) = \sum_{J=|k-\ell|}^{k+\ell} \varphi_J^{\ell k}(\|\mathbf{x}\|)\mathbf{W}_J^{\ell k}(\mathbf{x}), \quad \text{where } \mathbf{W}_J^{\ell k}(\mathbf{x}) = \sum_{m=-J}^{J} Y_{Jm}(\mathbf{x}/\|\mathbf{x}\|)\mathbf{Q}_{Jm}^{\ell k} \tag{9}$$

- $\mathbf{W}_J^{\ell k} : \mathbb{R}^3 \to \mathbb{R}^{(2\ell+1)\times(2k+1)}$,shape of Clebsch-Gordan matrices $\mathbf{Q}_{Jm}^{\ell k}(2\ell+1)\times(2k+1), Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$

TFN in SE(3)

put it together:

$$\mathbf{f}^{\ell}_{\text{out},i} = \sum_{k \geq 0} \sum_{j=1}^{n} \sum_{J=|k-\ell|}^{k+\ell} \varphi^{\ell k}_{J}(\|\mathbf{r}\|) \sum_{m=-J}^{J} Y_{Jm}(\mathbf{r}) \mathbf{Q}^{\ell k}_{Jm} \mathbf{f}^{k}_{\text{in},j} \tag{10}$$

$\mathbf{W}^{\ell k}_{J} : \mathbb{R}^3 \to \mathbb{R}^{(2\ell+1)\times(2k+1)}$, shape of $\quad \mathbf{Q}^{\ell k}_{Jm} (2\ell + 1) \times (2k + 1)$, $Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$

The SE(3)-Transformer

$$\mathbf{f}^{\ell}_{\text{out},i} = \underbrace{\mathbf{W}^{\ell\ell}_{V}\mathbf{f}^{\ell}_{\text{in},i}}_{\text{self-interaction}} + \sum_{k\geq 0}\sum_{j\in\mathcal{N}_i\setminus i} \underbrace{\alpha_{ij}}_{\text{attention}} \underbrace{\mathbf{W}^{\ell k}_{V}(\mathbf{x}_j - \mathbf{x}_i)\mathbf{f}^{k}_{\text{in},j}}_{\text{value message}}. \tag{11}$$

The SE(3)-Transformer itself consists of three components:

- edge-wise attention weights α_{ij},constructed to be SE(3)-invariant on each edge ij
- edge-wise SE(3)-equivariant value messages, propagating information between nodes
- a linear/attentive self-interaction layer

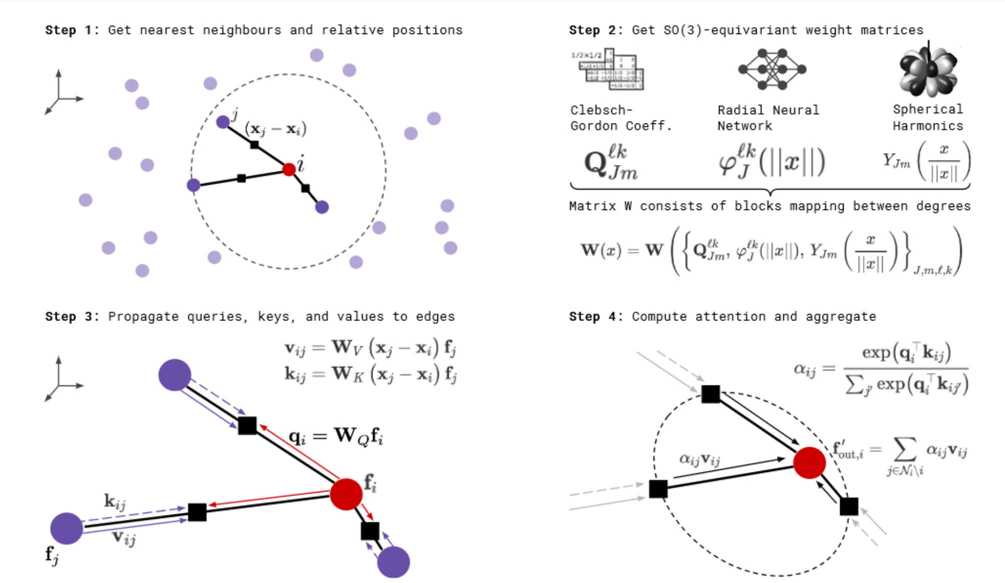
attention weights α_{ij}

The attention weights α_{ij} is calculated via a normalised inner product between a query vector $\mathbf{q}_i$ at node i and a set of key vectors $\mathbf{k}_{ij}{}_{j \in \mathcal{N}_i}$

$$\alpha_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_{ij})}{\sum_{j' \in \mathcal{N}_i \setminus i} \exp(\mathbf{q}_i^\top \mathbf{k}_{ij'})}, \quad \mathbf{q}_i = \bigoplus_{\ell \geq 0} \sum_{k \geq 0} \mathbf{W}_Q^{\ell k} \mathbf{f}_{\mathrm{in},i}^k, \quad \mathbf{k}_{ij} = \bigoplus_{\ell \geq 0} \sum_{k \geq 0} \mathbf{W}_K^{\ell k} (\mathbf{x}_j - \mathbf{x}_i) \mathbf{f}_{\mathrm{in},j}^k. \tag{12}$$

- $\bigoplus$ is the vector concatenation.
- The linear embedding matrices $\mathbf{W}_Q^{\ell k}$ and $\mathbf{W}_K^{\ell k}(\mathbf{x}_j - \mathbf{x}_i)$ are of TFN type.
- The attention weights α_{ij} is because of the invariance of the inner product of two SO(3)-equivariant vectors

Updating the node features



eSCN

view of Spherical Harmonics

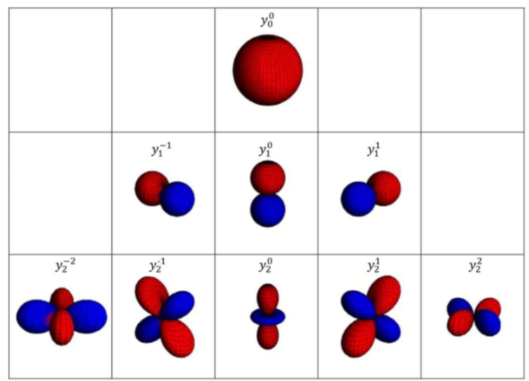


Figure 6: first 3 bands of SH function image (red for positive, blue for negative.)

eSCN

We consider the message m_{ts} sent from source node s to target node t in a SO(3) convolution. The L_o-th degree of m_{ts} can be expressed as:

$$m_{ts}^{(L_o)} = \sum_{L_i, L_f} w_{L_i, L_f, L_o} \left(x_s^{(L_i)} \otimes_{L_i, L_f}^{L_o} Y^{(L_f)}(\hat{r}_{ts}) \right) \tag{13}$$

Therefore,By choosing a specific R, we can reduce the cost of computing equation above substantially. Specifically, if select a rotation matrix $\mathbf{R}_{ts}$ so that $\mathbf{R}_{ts} \cdot \hat{\mathbf{r}}_{ts} = (0, 0, 1)$, the $\mathbf{Y}(\mathbf{R}_{st} \cdot \hat{\mathbf{r}}_{st})$ become sparse:

$$\mathbf{Y}_m^{(l)}(\mathbf{R}_{ts} \cdot \hat{\mathbf{r}}_{ts}) \propto \delta_m^{(l)} = \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{if } m \neq 0 \end{cases} \tag{14}$$

$$
\begin{aligned}
m_{ts}^{(L_o)} &= \sum_{L_i, L_f} w_{L_i, L_f, L_o} \left(x_s^{(L_i)} \otimes_{L_i, L_f}^{L_o} Y^{(L_f)}(\hat{r}_{ts}) \right) \\
m_{ts}^{(L_o)} &= \left(D^{(L_o)}(R_{ts}) \right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \left(D^{(L_i)}(R_{ts}) x_s^{(L_i)} \otimes_{L_i, L_f}^{L_o} Y^{(L_f)}(R_{ts} \hat{r}_{ts}) \right) \\
&= \left(D^{(L_o)} \right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\sum_{m_i, m_f} \left(D^{(L_i)} x_s^{(L_i)} \right)_{m_i} C_{(L_i, m_i),(L_f, m_f)}^{(L_o, m_o)} \left(Y^{(L_f)}(R_{ts} \hat{r_{ts}}) \right)_{m_f} \right) \\
&= \left(D^{(L_o)} \right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\sum_{m_i} \left(\tilde{x}_s^{(L_i)} \right)_{m_i} C_{(L_i, m_i),(L_f, 0)}^{(L_o, m_o)} \right)
\end{aligned} \tag{15}
$$

Additionally, given $m_f = 0$ Clebsch-Gordan coefficients $C^{(L_o,m_o)}_{(L_i,m_i),(L_f,0)}$ are sparse and are non-zero only when $m_i = \pm m_o$.

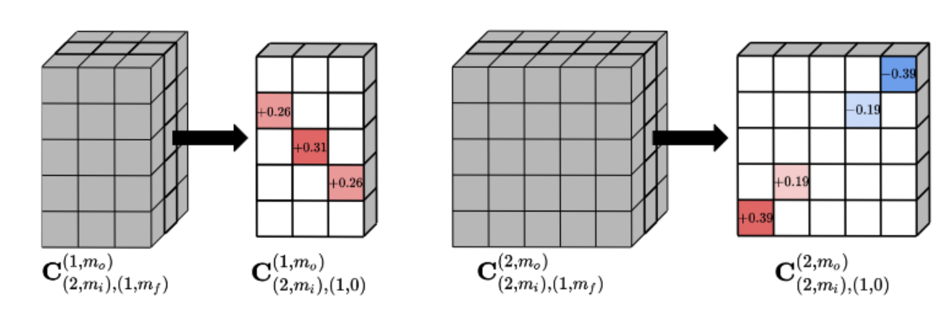


Figure 7: Visual representation of the Clebsch-Gordan matrices $\mathbf{C}^{(1,m_o)}_{(2,m_i),(1,m_f)} \in \mathbb{R}^{5\times3\times3}$ and $\mathbf{C}^{(2,m_o)}_{(2,m_i),(1,m_f)} \in \mathbb{R}^{5\times3\times5}$

eSCN

Therefore,

$$
\begin{aligned}
m_{ts}^{(L_o)} &= \left(D^{(L_o)}\right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\sum_{m_i} \left(\tilde{x}_s^{(L_i)}\right)_{m_i} C_{(L_i, m_i), (L_f, 0)}^{(L_o, m_o)} \right) \\
&= \left(D^{(L_o)}\right)^{-1} \sum_{L_i, L_f} w_{L_i, L_f, L_o} \bigoplus_{m_o} \left(\left(\tilde{x}_s^{(L_i)}\right)_{m_o} C_{(L_i, m_o), (L_f, 0)}^{(L_o, m_o)} + \left(\tilde{x}_s^{(L_i)}\right)_{-m_o} C_{(L_i, -m_o), (L_f, 0)}^{(L_o, m_o)} \right) \\
&= \left(D^{(L_o)}\right)^{-1} \sum_{L_i} \bigoplus_{m_o} \left(\left(\tilde{x}_s^{(L_i)}\right)_{m_o} \sum_{L_f} \left(w_{L_i, L_f, L_o} C_{(L_i, m_o), (L_f, 0)}^{(L_o, m_o)} \right) \right. \\
&\left. + \left(\tilde{x}_s^{(L_i)}\right)_{m_o} \sum_{L_f} \left(w_{L_i, L_f, L_o} C_{(L_i, -m_o), (L_f, 0)}^{(L_o, m_o)} \right) \right)
\end{aligned}
$$

eSCN

Instead of using learnable parameters for w_{L_i,L_f,L_o},eSCN proposes to parameterize $\tilde{w}_{m_o}^{(L_i,L_o)}$ and $\tilde{w}_{-m_o}^{(L_i,L_o)}$

$$\tilde{w}_{m_o}^{(L_i,L_o)} = \sum_{L_f} w_{L_i,L_f,L_o} C_{(L_i,m_o),(L_f,0)}^{(L_o,m_o)} = \sum_{L_f} w_{L_i,L_f,L_o} C_{(L_i,-m_o),(L_f,0)}^{(L_o,-m_o)} \quad \text{for } m >= 0$$

$${}_{-m_o}^{(L_i,L_o)} = \sum_{L_f} w_{L_i,L_f,L_o} C_{(L_i,m_o),(L_f,0)}^{(L_o,-m_o)} = -\sum_{L_f} w_{L_i,L_f,L_o} C_{(L_i,-m_o),(L_f,0)}^{(L_o,m_o)} \quad \text{for } m > 0$$

eSCN

Finally,we have:

$$m_{ts}^{(L_o)} = \left(D^{(L_o)}\right)^{-1} \sum_{L_i} \bigoplus_{m_o} \left(y_{ts}^{(L_i,L_o)}\right)_{m_o} \tag{16}$$

$$\begin{aligned}
\left(y_{ts}^{(L_i,L_o)}\right)_{m_o} &= \tilde{w}_{m_o}^{(L_i,L_o)} \left(\tilde{x}_s^{(L_i)}\right)_{m_o} - \tilde{w}_{-m_o}^{(L_i,L_o)} \left(\tilde{x}_s^{(L_i)}\right)_{-m_o} \quad \text{for } m_o > 0 \\
\left(y_{ts}^{(L_i,L_o)}\right)_{-m_o} &= \tilde{w}_{-m_o}^{(L_i,L_o)} \left(\tilde{x}_s^{(L_i)}\right)_{m_o} + \tilde{w}_{m_o}^{(L_i,L_o)} \left(\tilde{x}_s^{(L_i)}\right)_{-m_o} \quad \text{for} m_o > 0 \\
\left(y_{ts}^{(L_i,L_o)}\right)_{m_o} &= \tilde{w}_{m_o}^{(L_i,L_o)} \left(\tilde{x}_s^{(L_i)}\right)_{m_o} \quad \text{for} m_o = 0
\end{aligned}$$