PASM 2026 Lecture 4

Recipe for equivariant neural network:

1. Choose symmetry group.

2. Convert input with feature type $h^{(\ell)}$. → irrep

3. use equivariant linear map — no mixing between different ℓ.

4. CG nonlinearity added when mixing happens.

   (Add scalar gating)

5. Stack another equivariant layer

6. Read out invariant/equivariant output from different irrep channels.

Example $E \in V_0$, $\vec{F} \in V_1$

Scalar channel, $\ell=1$ channel (vector)

The symmetry structure is fixed by representation theory,

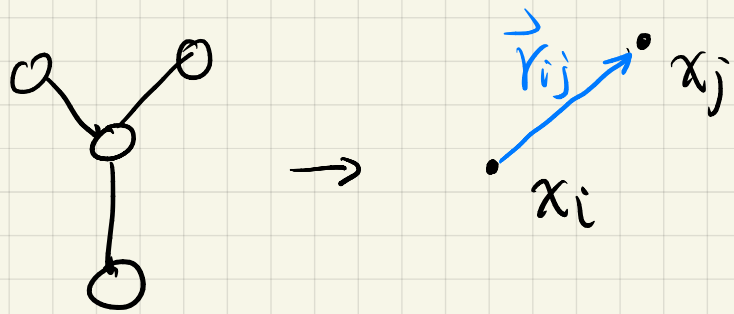
the learnable lives in channel mixing!

Input features for atomic systems:

Scalar: $\left\{\begin{array}{l}\text{element type} \\ \text{charge} \\ \text{local scalar descriptors}\end{array}\right.$ $\in V_0$.

Vector: spin orientation $\in V_1$.

Geometric:



x_i, x_j: Cartesian coordinates

$$\vec{r}_{ij} = x_j - x_i \quad : \text{relative position}$$

$$= \underbrace{|\vec{r}_{ij}|}_{\text{related to weight}}\, \hat{r}_{ij} \qquad \hat{r}_{ij} \in V_1$$

Equivariant NN for graphs

In a graph, each node i has features:

$$h_i = \bigoplus_{\ell} h_i^{(\ell)}$$

Between node i and neighbor nodes: $\vec{r}_{ij} = x_j - x_i$

Message passing:

$$\begin{cases} \overset{\text{message}}{m_{ij}} = \phi_m(h_i, h_j, \vec{r}_{ij}) \\ h_i' = \underset{\text{aggregation}}{\mathrm{AGG}} \left(h_i, \sum_{j \in N(i)} m_{ij}\right) \end{cases}$$

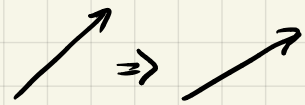
h_i, h_j: node feature

$e_{ij} = \vec{r}_{ij}$: edge feature

$N(i)$: neighbors of node i.

Under rotation R:

$$\vec{r}_{ij} \rightarrow R\vec{r}_{ij}$$



$$\phi_{nlm}(\vec{r}_{ij}) = R_n(|\vec{r}_{ij}|)\, Y_l^m(\hat{r}_{ij})$$

$R_n(|\vec{r}_{ij}|)$: radial basis

$Y_l^m(\hat{r}_{ij})$: angular basis (described by irrep l).

Equivariant message construction:

① The message:

$$m_{ij}^{(\ell)} = \sum_{\ell_1, \ell_2} W_{\ell_1, \ell_2, \ell} \, CG_{\ell_1, \ell_2 \to \ell} \left(h_j^{(\ell_1)} \otimes \phi(r_{ij})^{(\ell_2)} \right)$$

irrep type ℓ

Notation: $h_j^{(\ell_1)}$: feature of neighbor node j with irrep type ℓ_1

$\phi(r_{ij})^{(\ell_2)}$: edge with type ℓ_2.

$W_{\ell_1, \ell_2, \ell}$: learnable channel weights.

$\otimes$: tensor product.

② Aggregation:

Similar to regular GNN:

$$\sigma^{(\ell)} \left(h_i^{(\ell)}, \sum_{j \in N(i)} m_{ij}^{(\ell)} \right) \longrightarrow h_i^{(\ell)}$$

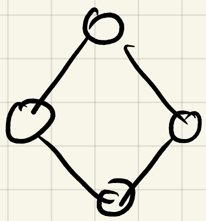
$\downarrow$

Nonlinearity:

① tensor product

② scalar gating.

The workflow for a equivariant GNN layer :



graph

→ input feature
neighbor feature
&
relative geometry
→ tensor product + CG decomposition

→ equivariant message → $\sigma(AGG(\cdot))$ → equivariant node update

---

Output of equivariant GNN :

Example :

Energy (scalar) : $E = \sum_i MLP(\underbrace{h_i^{(0)}}_{\text{scalar channel}})$

This is invariant !

Force (vector) :

Two ways : $\begin{cases} \vec{F} = \dfrac{dE}{d\vec{r}} & \text{differentiating energy wrt displacement} \\ \text{from } V_1 \text{ channel of each node.} & F_i \in V_1 \end{cases}$

Tensor :

V_2 channel or higher-order channel of whole graph.

Application of ENN for molecule and material properties

$E(3)$ / $SO(3)$ / $E(3)$+SOC

forces, tensors, tight-binding Hamiltonian elements

tensorial spectrum

The future:

- Combination with PINN.
- other groups
  ↳ point groups $\begin{pmatrix} \text{discrete, multiplicity} \\ \text{CG,} \end{pmatrix}$
  arXiv (2026).
- Combination with generative model.
- Elementary band representation theory