CrystalFramer: Rethinking the role of frames for SE(3)-invariant crystal structure modeling

TL;DR: To make a GNN invariant to rotations, let's standardize the orientations of local atomic environments represented by internal self-attention weights!

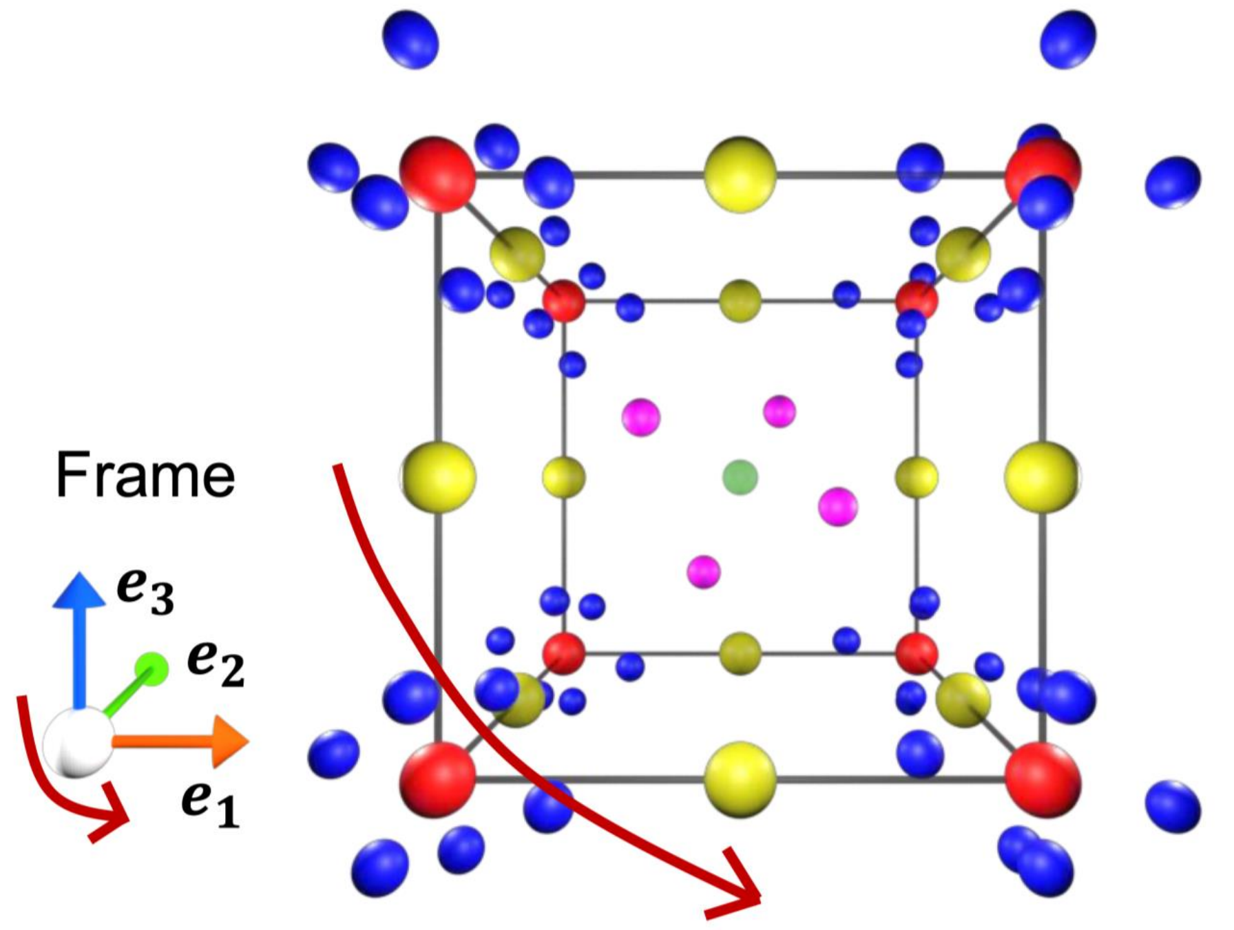
Yusei Ito[1,2*] Tatsunori Taniai[1*] Ryo Igarashi[1] Yoshitaka Ushiku[1] Kanta Ono[2]

[1] OMRON SINIC X Corporation [2] Osaka University [*] Equal contribution

Frame-based SE(3)-invariant crystal structure modeling

The key to modeling crystal structures lies in learning **SE(3)-invariant (*i.e.*, rotation & translation invariant) representations.**

Why not normalize the orientation of the input structure?



That's the "frame" (*i.e.*, structure-aligned coordinate system)

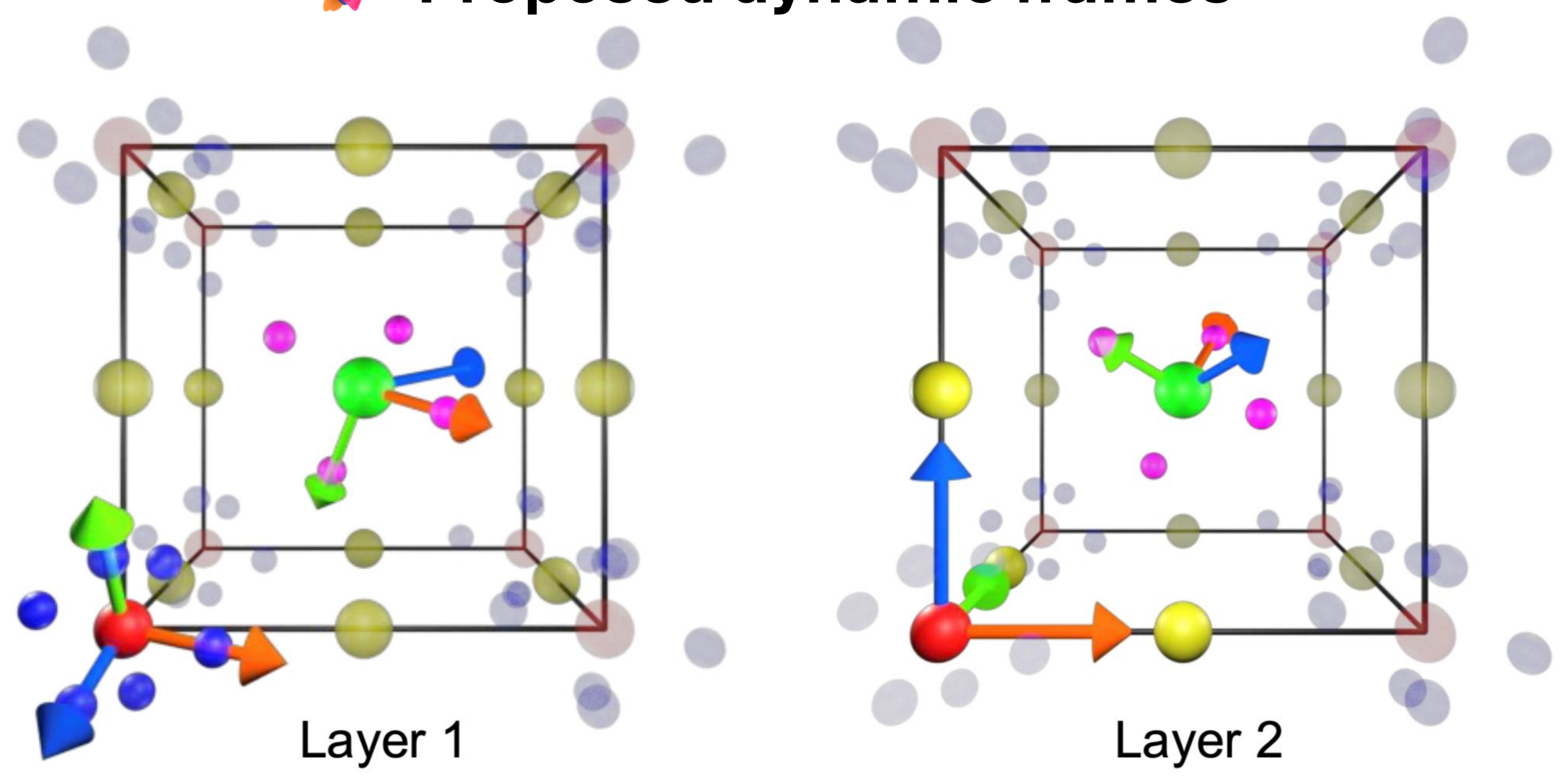
Ex 1) PCA for the positions of atoms in a unit cell (PCA frame)

Ex 2) Lattice vectors of a unit cell (lattice frame)

- Can use richer information than interatomic distances
- No restriction on the network architectures

- **Is it enough to just align with the structure?**
- **Is there no need to adapt the frame based on the task?**

Proposed dynamic frames



The fundamental role of the frame is *to effectively incorporate relative positional information* into the message-passing layers for **modeling interatomic interactions**:

$$\boldsymbol{x}'_i = \sum_j w_{ij} \boldsymbol{f}_{i\leftarrow j}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{r}_{ij})$$

Let's consider **a frame aligned with the interatomic interactions** rather than with the structure itself.

Dynamic frames: attention-based local frames

Suppose a message-passing layer in a general form: $\boldsymbol{x}'_i = \sum_j w_{ij} \boldsymbol{f}_{i\leftarrow j}$.

Interaction weights w_{ij} can be interpreted as **a "mask" representing the local environment of the structure viewed from target atom i.**

Weighted PCA frame

For each atom i, compute the weighted covariance matrix of the direction vectors $\bar{\boldsymbol{r}}_{ij}$ pointing toward the surrounding atoms j.

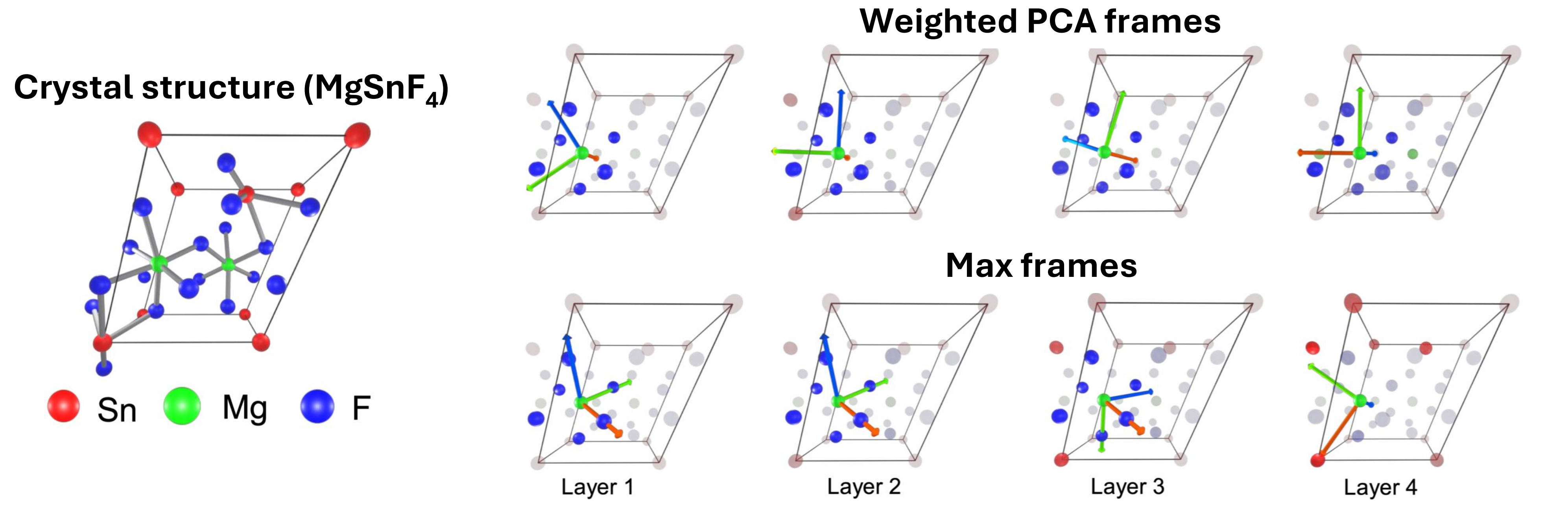
$$\Sigma_i = \sum_j w_{ij} \bar{\boldsymbol{r}}_{ij} \bar{\boldsymbol{r}}_{ij}^T$$

Set the orthonormal eigenvectors $[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$ of the matrix as the three axes of the frame.

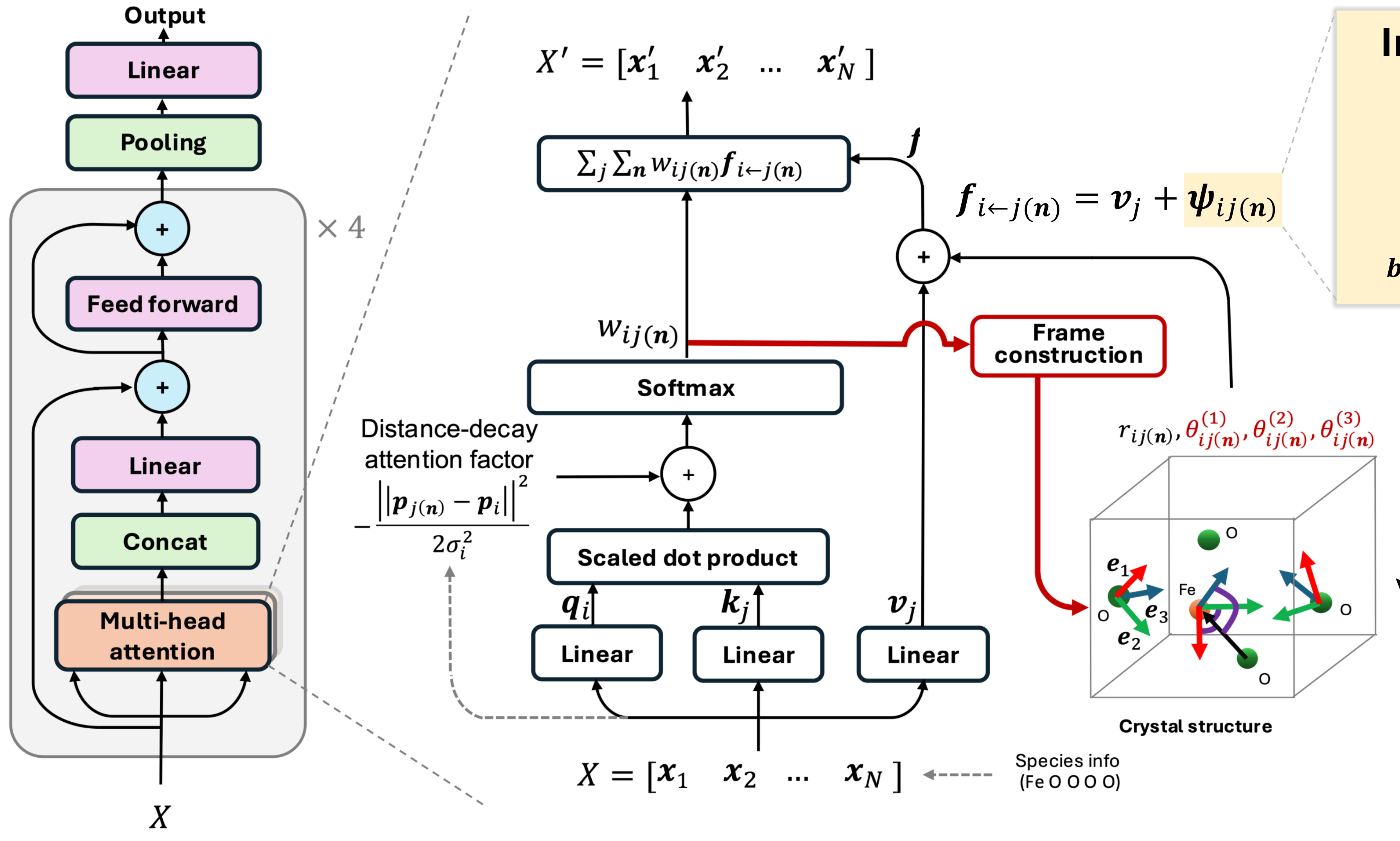
Max frame

(Weight-sorted selection & orthogonalization)

1. Set the first axis $\boldsymbol{e}_1$ to the direction $\bar{\boldsymbol{r}}_{ij}$ pointing toward the atom j with the highest weight w_{ij}.
2. To ensure diversity, select the direction $\bar{\boldsymbol{r}}_{ij}$ for the second axis candidate $\hat{\boldsymbol{e}}_2$ by maximizing the adjusted weight $(1 - |\boldsymbol{e}_1 \cdot \bar{\boldsymbol{r}}_{ij}|) w_{ij}$, which penalizes alignment with $\boldsymbol{e}_1$.
3. Apply Gram–Schmidt orthogonalization to obtain $\boldsymbol{e}_2 = \hat{\boldsymbol{e}}_2 - (\boldsymbol{e}_1 \cdot \hat{\boldsymbol{e}}_2)\boldsymbol{e}_1$.
4. Set $\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2$ to form a right-handed orthonormal system.



CrystalFramer: Crystalformer (Taniai et al., 2024) + Dynamic frames


Output
Linear
Pooling
Feed forward
Linear
Concat
Multi-head attention
× 4
X
$X' = [\boldsymbol{x}'_1 \ \boldsymbol{x}'_2 \ \dots \ \boldsymbol{x}'_N]$
$\sum_j \sum_n w_{ij(n)} \boldsymbol{f}_{i\leftarrow j(n)}$
$\boldsymbol{f}$
$\boldsymbol{f}_{i\leftarrow j(n)} = \boldsymbol{v}_j + \boldsymbol{\psi}_{ij(n)}$
$w_{ij(n)}$
Softmax
Frame construction
Distance-decay attention factor
$-\frac{\|\boldsymbol{p}_{j(n)} - \boldsymbol{p}_i\|^2}{2\sigma_i^2}$
Scaled dot product
$\boldsymbol{q}_i$
$\boldsymbol{k}_j$
$\boldsymbol{v}_j$
Linear
Linear
Linear
$r_{ij(n)}, \theta^{(1)}_{ij(n)}, \theta^{(2)}_{ij(n)}, \theta^{(3)}_{ij(n)}$
e_1
e_2
e_3
Fe
O
O
O
O
Crystal structure
$X = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \dots \ \boldsymbol{x}_N]$
Species info (Fe O O O O)

Invariant edge feature

Projection onto the frame coordinate system

$$\boldsymbol{\psi}_{ij(n)} = \underbrace{\boldsymbol{W}_0 \boldsymbol{b}_{\text{dist}}(r_{ij(n)})}_{\text{Distance encoding}} + \underbrace{\sum_{k=1,2,3} \boldsymbol{W}_k \boldsymbol{b}_{\text{angl}}(\boldsymbol{e}_k^T \bar{\boldsymbol{r}}_{ij(n)})}_{\text{Angle encoding}}$$

$\boldsymbol{b}(x)$: Gaussian basis functions that convert a scalar value into a soft one-hot vector

In each layer, a frame is dynamically constructed based on the atoms that the network is currently attending to. The normalized relative positions $\bar{\boldsymbol{r}}_{ij(n)}$ are projected onto the frame coordinate system as cosines of angles between frame axes $\boldsymbol{e}_k$ and $\bar{\boldsymbol{r}}_{ij(n)}$, and used as **position encodings $\boldsymbol{\psi}_{ij(n)}$ for the Value vectors $\boldsymbol{v}_j$.**

Physical property prediction tasks

	Materials Project (MEGNET's snapshot)				JARVIS-DFT 3D				
	E form	Bandgap	Bulk mod.	Shear mod.	E form	E total	Bandgap(opt)	Bandgap(mbj)	E hull
Method	eV/atom	eV	log (GPa)	log (GPa)	eV/atom	eV/atom	eV	eV	eV
Matformer (Yan et al., 2023)	0.021	0.211	0.043	0.073	0.0325	0.035	0.137	0.30	0.064
PotNet (Lin et al., 2023)	0.0188	0.204	0.040	0.065	0.0294	0.032	0.127	0.27	0.055
eComformer (Yan et al., 2024)	0.0182	0.202	0.0417	0.0729	0.0284	0.032	0.124	0.28	**0.044**
iComformer (Yan et al., 2024)	0.0183	0.193	0.0380	**0.0637**	0.0272	0.0288	0.122	0.26	0.047
Crystalformer (Taniai et al., 2024)	0.0186	0.198	0.0377	0.0689	0.0306	0.0320	0.128	0.274	0.0463
– w/ PCA frames (Duval et al., 2023)	0.0197	0.217	0.0424	0.0719	0.0325	0.0334	0.144	0.292	0.0568
– w/ lattice frames (Yan et al., 2024)	0.0194	0.212	0.0389	0.0720	0.0302	0.0323	0.125	0.274	0.0531
– w/ static local frames	0.0178	0.191	0.0354	0.0708	0.0285	0.0292	0.122	0.261	**0.0444**
– w/ weighted PCA frames (**proposed**)	0.0197	0.214	0.0423	0.0715	0.0287	0.0305	0.126	0.279	**0.0444**
– w/ max frames (**proposed**)	**0.0172**	**0.185**	**0.0338**	0.0677	**0.0263**	**0.0279**	**0.117**	**0.242**	0.0471
CrystalFramer (default)	**0.0172**	**0.185**	**0.0338**	0.0677	**0.0263**	**0.0279**	**0.117**	**0.242**	0.0471
CrystalFramer (lightweight)	0.0176	0.191	0.0341	**0.0654**	0.0268	**0.0279**	**0.117**	0.262	**0.0467**

Method	Time/epoch	Test/mater.	# Params.
Matformer	60 s	20.4 ms	2.9 M
PotNet	43 s	313 ms	1.8 M
iComFormer	59 s	54.8 ms	5.0 M
Crystalformer	32 s	6.6 ms	853 K
CrystalFramer (default)	74 s	16.8 ms	952 K
CrystalFramer (lightweight)	43 s	15.2 ms	878 K

✓ **Incorporating dynamic frame-based edge features substantially enhances the prediction performance.**

✓ **High model efficiency with only a small increase in parameters compared to the baseline Crystalformer.**

The default ver encodes an angle into a 64-D vector using 64 Gaussian basis functions, while the lightweight ver uses a 16-D vec.