Visual Intuition for Local vs. Global Frames in AF2

Study Notes

December 9, 2025

1 Motivating Question

Why are local coordinates even needed in the AlphaFold 2 (AF2) structure module? Why do we go through the apparent "gymnastics" of local and global frames instead of working purely in one global coordinate system?

This note builds a concrete mental picture, then ties it directly to how AF2 uses local frames to achieve rotational invariance in invariant point attention (IPA).

2 Mental Picture: Two Residues in the Plane

Consider a protein sitting in ordinary 3D space with a global coordinate system (x, y, z). For visualization, imagine two residues:

- residue 1 (rs1) at position $x_1 \in \mathbb{R}^3$,
- residue 2 (rs2) at position $x_2 \in \mathbb{R}^3$.

For simplicity, assume both lie in the xy-plane ($z = 0$) in the first quadrant. Intuitively, rs2 is "down and to the right" of rs1 ("lower-right").

We want to compare:

- global description: "where is rs2 relative to rs1 in the global frame?",
- local description: "where is rs2 relative to rs1 as rs1 sees it in its own axes?".

3 Global vs. Local Description of rs2 Relative to rs1

3.1 Global Description

In global coordinates, the vector from rs1 to rs2 is

$$v_{\text{global}} = x_2 - x_1.$$

If we rotate the entire protein by a rotation matrix $G \in \mathrm{SO}(3)$:

$$x_1' = Gx_1, \qquad x_2' = Gx_2,$$

then the new global vector is

$$v_{\text{global}}' = x_2' - x_1' = Gx_2 - Gx_1 = G(x_2 - x_1) = Gv_{\text{global}}.$$

Thus in the global frame, v_{global} itself rotates; its coordinates change.

3.2 Local Description: rs1's Own Coordinate System

Now suppose rs1 carries its own local axes:

- three unit vectors e_1, e_2, e_3 forming an orthonormal basis attached to rs1,
- in AF2 these come from the N, Cα, C atoms via Gram–Schmidt.

Pack them as columns of a rotation matrix

$$R_1 = [e_1\ e_2\ e_3].$$

Then:

- R_1 maps local coordinates to global coordinates,
- $R_1^\top$ maps global coordinates to local coordinates.

The coordinates of v_{global} in rs1's local frame are

$$v_{\text{local}} = R_1^\top v_{\text{global}}.$$

This is the vector "from rs1 to rs2 as rs1 sees it", expressed in rs1's own axes.

4 Global Rotation of the Entire Protein

Apply a global rotation $G \in \mathrm{SO}(3)$ to everything:

$$x_1' = Gx_1, \qquad x_2' = Gx_2, \qquad v_{\text{global}}' = Gv_{\text{global}}.$$

The backbone of rs1 (and its local axes) also rotate:

$$e_1' = Ge_1, \quad e_2' = Ge_2, \quad e_3' = Ge_3$$

so the new local-to-global matrix is

$$R_1' = [e_1'\ e_2'\ e_3'] = G[e_1\ e_2\ e_3] = GR_1.$$

Compute the new local coordinates of rs2 relative to rs1 after the global rotation:

$$v_{\text{local}}' = (R_1')^\top v_{\text{global}}' = (GR_1)^\top (Gv_{\text{global}}) = R_1^\top G^\top G v_{\text{global}} = R_1^\top v_{\text{global}} = v_{\text{local}}.$$

So:

> The global vector from rs1 to rs2 rotates ($v_{\text{global}} \mapsto Gv_{\text{global}}$), but its coordinates expressed in rs1's local frame (v_{local}) stay exactly the same, provided rs1's frame is rotated by the same G.

This is the precise form of invariance that AF2 wants.

5 Why AF2 Needs Local Frames, Not Just Global Vectors

Suppose the model only used global coordinates and stored $x_2 - x_1$ directly in features:

- Rotating the protein changes those numbers.
- The network would have to learn that many rotated versions of the same neighborhood are equivalent.
- We would likely need heavy data augmentation over random orientations to encourage rotational invariance.

Instead, AF2 does two key things:

1. It associates each residue i with a local frame $T_i = (R_i, t_i)$:
   - R_i: orientation of residue i's axes in global space,
   - t_i: global position of the origin (Cα) of residue i.
2. It represents learned geometric information in local coordinates:
   - query points $\tilde{q}_i^{h,p} \in \mathbb{R}^3$ in residue-i's local frame,
   - key points $\tilde{k}_j^{h,p} \in \mathbb{R}^3$ in residue-j's local frame,
   - value points $\tilde{v}_j^{h,p} \in \mathbb{R}^3$ in residue-j's local frame.

Because these points are stored in local frames, they naturally encode relations like "rs2 is lower-right of rs1" in a way that is invariant under global rotations, as long as all frames transform consistently.

Whenever AF2 needs to compare positions, it:

- converts local points to global coordinates via T_i,
- computes distances or averages in global space,
- maps results back into local coordinates via T_i^{-1} when feeding them back into the network.

The IPA formulas are constructed so that if we apply the same global rigid motion T_{global} to all frames T_i, the attention logits and local outputs remain unchanged.

6 Link to the "rs1 Center, rs2 Lower-Right" Picture

Return to the cartoon:

- rs1 somewhere in the first quadrant,
- rs2 down and to the right of rs1.

In global coordinates:

- positions are x_1, x_2,

- the vector $v_{\text{global}} = x_2 - x_1$ captures "lower-right" in the global frame.

In rs1's local frame:

$$v_{\text{local}} = R_1^\top (x_2 - x_1)$$

captures the same "lower-right" relation relative to rs1's own axes.

After a global rotation G:

- v'_{global} points in a completely different direction in the global frame,
- $v'_{\text{local}} = v_{\text{local}}$; rs1 still sees rs2 in the same direction and at the same distance when measured in its own local frame.

Thus the abstract relation "rs2 is lower-right of rs1 by this much" is encoded by v_{local} in a way that does not depend on how we orient the entire protein in space.

This is exactly the invariance AF2 wants: local geometry around each residue should have a consistent representation, regardless of the arbitrary global orientation chosen when the structure is fed into the model.

7 Connecting Back to IPA

In invariant point attention, for each head h and residues i, j:

- query points $\tilde{q}_i^{h,p}$ live in the local frame of i,
- key points $\tilde{k}_j^{h,p}$ live in the local frame of j,
- frames T_i, T_j convert these to global coordinates for comparison.

The squared distances in the attention logits are

$$D_{ij}^h = \sum_p \left\| T_i \circ \tilde{q}_i^{h,p} - T_j \circ \tilde{k}_j^{h,p} \right\|^2.$$

Because T_i and T_j are rigid transforms, these distances behave like the v_{local} story above: if we apply a global transform T_{global} to all frames, D_{ij}^h is unchanged. Thus the attention weights do not depend on the global orientation of the protein.

The point-valued outputs

$$\tilde{o}_i^{h,p} = T_i^{-1} \circ \left(\sum_j a_{ij}^h (T_j \circ \tilde{v}_j^{h,p}) \right)$$

are likewise expressed back in the local frame of i, preserving the same invariance property.

8 Summary

- Global vectors (like $x_2 - x_1$) rotate when we rotate the entire protein; their coordinates depend on the arbitrary choice of global axes.
- Local vectors (like $v_{\text{local}} = R_1^\top (x_2 - x_1)$) measured in residue-specific frames remain invariant under global rigid motions, provided the frames transform with the structure.

- AF2 exploits this by:
  - attaching a local rigid frame to each residue,
  - expressing learned geometric quantities in those local frames,
  - and only using global coordinates transiently to measure distances and aggregate information.

This makes the per-residue geometric representation insensitive to the arbitrary global orientation of the protein and is the core reason local coordinates are crucial in the AF2 structure module.