IPA Viewed as Ordinary Self-Attention at Its Core

Study Notes

December 8, 2025

1 Introduction

It is tempting to think that invariant point attention (IPA) departs fundamentally from ordinary self-attention: dot products suddenly coexist with distances, frames, local and global coordinates. The core message of this note is:

> IPA is still ordinary self-attention at its core. It simply adds a geometric bias term to the score, computed from local/global frames and learned points.

We build up this view in layers, starting from vanilla self-attention and gradually introducing pair biases, geometric terms, and frame-based invariance.

2 Standard Self-Attention: Only Dot Products

Forget proteins for a moment. Consider a vanilla Transformer layer acting on a sequence of tokens. You have:

- A sequence of vectors $(x_1, x_2, \ldots, x_N)$,
- For each token i and head h, you compute

$$\begin{aligned} q_i^h &= W_q^h x_i, \\ k_j^h &= W_k^h x_j, \\ v_j^h &= W_v^h x_j, \end{aligned}$$

  where W_q^h, W_k^h, W_v^h are learned matrices for head h.

  The attention scores (logits) before softmax are

$$\text{score}_{ij}^h = \frac{1}{\sqrt{d}} \big(q_i^h\big)^\top k_j^h.$$

The corresponding weights are

$$a_{ij}^h = \frac{\exp(\text{score}_{ij}^h)}{\sum_{j'} \exp(\text{score}_{ij'}^h)}.$$

The output for token i and head h is

$$o_i^h = \sum_j a_{ij}^h v_j^h.$$

So the overall pattern is:

1. Compute a scalar score for each pair (i, j) indicating how much i should listen to j,
2. Use softmax to convert scores into weights,
3. Take a weighted average of value vectors.

The only design choice here is the precise formula for the score score_{ij}^h. In vanilla attention this is just a scaled dot product.

3 Attention with Extra Bias Terms

Now imagine we want attention to also depend on additional information, for example:

- Distance between positions i and j along the sequence,
- A learned bias for each pair (i, j) from another network,
- Relative 2D position in a Vision Transformer.

We are free to modify the score formula. For example,

$$\text{score}_{ij}^h = \frac{1}{\sqrt{d}}(q_i^h)^\top k_j^h + b_{ij}^h,$$

where b_{ij}^h is some scalar bias that depends on (i, j).

This is still self-attention:

- We have not changed softmax,
- We have not changed the weighted sum over values.

We only changed the function that produces the scalar score before softmax.

Many modern Transformer architectures do this: they add learned relative positional bias, 2D bias, graph-structure bias, and so on.

We can adopt a more general mental model:

> Attention computes scores with some learned similarity function, not necessarily just a bare dot product. The pure dot product is a special case.

IPA fits cleanly into this picture.

4 AlphaFold's IPA: Adding Geometry to the Score

In IPA, AlphaFold uses an attention score of the form

$$\text{score}_{ij}^h = w_L\Big(\underbrace{\sqrt{\tfrac{1}{c}}(q_i^h)^\top k_j^h}_{\text{usual dot product}} + \underbrace{b_{ij}^h}_{\text{pair bias}} - \underbrace{\gamma_h w_C^2 D_{ij}^h}_{\text{geometric penalty}} \Big),$$

and then

$$a_{ij}^h = \text{softmax}_j\big(\text{score}_{ij}^h\big).$$

This is Algorithm 22, line 7 in the AF2 supplement. The only genuinely new ingredient is D_{ij}^h, the geometry-based term.

Structurally:

- We retain the standard dot-product similarity between features,
- Add a learned bias from the pair representation z_{ij},
- Subtract a nonnegative number D_{ij}^h that grows when i and j are geometrically far apart in 3D.

If we delete the last term, we recover standard attention with a pair bias. So IPA is most naturally read as

> Standard attention where the score function has been extended to include distances as well as dot products.

5 Why Introduce Distances in the Score?

Attention answers the question: "who should I listen to?".

In language, this is mostly driven by feature similarity. In geometry it is natural to also demand that:

- Residues close in 3D space are often more relevant to each other,
- Very distant residues may still matter occasionally, but less often.

For an evolving 3D structure, we want a strong locality bias in 3D space.

That is what the distance term does. For head h, the contribution

$$-\gamma_h w_C^2 D_{ij}^h$$

is always non-positive. If residues are close in 3D (in a sense defined by the learned points), D_{ij}^h is small and so is the penalty. If they are far apart, D_{ij}^h is large and the score is pushed down, reducing the attention weight.

The rest of the score function remains a dot product plus pair bias; the distance term is "just" a geometry-aware bias.

6 The Geometric Term Itself: How Distances Appear

We now unpack the definition of D_{ij}^h.

For head h and residues i, j, Algorithm 22 defines

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

6.1 Local Points

For each residue i and head h, the network predicts:

- Query points $\tilde{q}_i^{hp} \in \mathbb{R}^3$ for $p = 1, \ldots, P$,
- Key points $\tilde{k}_i^{hp} \in \mathbb{R}^3$ for $p = 1, \ldots, P$.

These live in the local frame of residue i, not in global coordinates. Each is like a point at some offset from the Cα atom of i in i's private coordinate system.

Thus, each head h carries a small constellation of P query points and P key points attached to each residue.

6.2 Frames T_i

For each residue i, there is a frame $T_i = (R_i, t_i)$ constructed from backbone geometry in the structure module.

Interpretation:

- Local to global: given a local point $u \in \mathbb{R}^3$, the global position is

$$T_i \circ u = R_i u + t_i.$$

- Global to local: given a global point x, the local coordinates are

$$T_i^{-1} \circ x = R_i^{-1}(x - t_i).$$

You can think of "apply T_i" as attaching a point physically to residue i in 3D space.

6.3 Distances in Global Space

To compare constellations of residues i and j, we need them expressed in the same coordinate system. For each point index p:

- Compute the global query point

$$g_i^{hp} = T_i \circ \tilde{q}_i^{hp} = R_i \tilde{q}_i^{hp} + t_i,$$

- Compute the global key point

$$g_j^{hp} = T_j \circ \tilde{k}_j^{hp} = R_j \tilde{k}_j^{hp} + t_j,$$

- Compute the squared Euclidean distance

$$d_{ij}^{hp,2} = \left\| g_i^{hp} - g_j^{hp} \right\|^2.$$

Then sum over p:

$$D_{ij}^h = \sum_p d_{ij}^{hp,2}.$$

This D_{ij}^h plays the role of a penalty in the score: large values reflect constellations that do not align well in global space.

In words:

> Head h attaches a small 3D pattern of query and key points to each residue. It uses the current backbone frames to map these to global 3D and measures squared distances between corresponding points. If two residues' constellations align well, the penalty is small; if not, the penalty is large.

7 Local and Global, Again: Why Frames Matter

We can now summarize the role of local/global coordinates and frames in this geometric score:

1. Points $\tilde{q}_i^{hp}$ and $\tilde{k}_j^{hp}$ live in local coordinates; they are offsets relative to each residue.
2. The frames T_i and T_j tell us where each residue sits in global 3D and how it is oriented.
3. Composing them, $T_i \circ \tilde{q}_i^{hp}$ gives a global coordinate for a query point of residue i; similarly for keys.
4. Subtracting these global coordinates produces a true Euclidean distance in 3D.

We cannot compute a meaningful distance directly between two local vectors such as $\tilde{q}_i$ and $\tilde{k}_j$, because they live in different coordinate systems. In local coordinates, each residue has its own origin and axes. Frames provide the bridge needed to compare them.

8 Invariance: Why Global Transforms Do Not Matter

There is a second reason for careful local/global bookkeeping:

> We want all IPA outputs to be invariant to where the protein is placed in space.

If we rotate and translate the entire protein, predictions should not change in an arbitrary way; this is a symmetry of the physical system.

Mathematically:

- A global rigid motion is another transform $T_{\text{global}} = (R_g, t_g)$,
- New frames become $T_i' = T_{\text{global}} \circ T_i$.

Then, for a term inside D_{ij}^h:

$$\left\| T_i' \circ \tilde{q}_i^{hp} - T_j' \circ \tilde{k}_j^{hp} \right\|^2 = \left\| T_{\text{global}} \circ T_i \circ \tilde{q}_i^{hp} - T_{\text{global}} \circ T_j \circ \tilde{k}_j^{hp} \right\|^2.$$

Rigid transforms preserve distances, so this equals

$$\left\| T_i \circ \tilde{q}_i^{hp} - T_j \circ \tilde{k}_j^{hp} \right\|^2.$$

Therefore D_{ij}^h and the geometric penalty are unchanged, and the entire score remains the same. Distances appear in the score because we want to combine:

- standard content similarity (dot product),
- geometric proximity in 3D that is invariant to the choice of coordinates.

Frames and local/global transforms are the mechanism that makes these distances meaningful and invariant.

9 Outputs: Local to Global and Back Again

We now turn to the IPA outputs that involve point features, focusing on the expression

$$\tilde{o}_i^{hp} = T_i^{-1} \circ \Big(\sum_j a_{ij}^h \big(T_j \circ \tilde{v}_j^{hp} \big) \Big).$$

At first sight this is opaque, but it follows the same pattern as the geometric term.

9.1 What This Output Represents

Beyond scalar attention weights, each head wants to produce geometric information:

- For each residue i, head h, and point index p, a vector in $\mathbb{R}^3$ describing the "average location" of neighbors' value points, expressed in residue i's local coordinates.

These outputs will be concatenated with other per-head outputs and passed through a linear layer to update the single representation.

9.2 Step-by-Step Interpretation

For a fixed head h, residue i, and point index p:

1. **Local to global for values:**
   - $\tilde{v}_j^{hp}$ is the value point in local coordinates of residue j,
   - The corresponding global point is $T_j \circ \tilde{v}_j^{hp}$.
2. **Attention-weighted average in global space:**

   $$y_i^{hp} = \sum_j a_{ij}^h \big(T_j \circ \tilde{v}_j^{hp} \big),$$

   a barycenter of neighbor points in global 3D.
3. **Global back to local for residue i:**

   $$\tilde{o}_i^{hp} = T_i^{-1} \circ y_i^{hp}.$$

   This expresses the averaged point in residue i's local frame.

So the flow is: local to global, global averaging, then global back to local.

9.3 Invariance of the Point Outputs

If we apply a global transform T_{global} to all frames:

- Global points for residue j become $T_{\text{global}} \circ (T_j \circ \tilde{v}_j^{hp})$,
- The global average becomes $T_{\text{global}} \circ y_i^{hp}$,
- The inverse of the new frame $T_i' = T_{\text{global}} \circ T_i$ is $T_i^{-1} \circ T_{\text{global}}^{-1}$.

Computing $\tilde{o}_i^{hp}$ under the new frames:

$$\tilde{o}_i^{hp,\prime} = (T_i')^{-1} \circ \Big(\sum_j a_{ij}^h \big(T_j' \circ \tilde{v}_j^{hp} \big) \Big) = T_i^{-1} \circ T_{\text{global}}^{-1} \circ T_{\text{global}} \circ y_i^{hp} = T_i^{-1} \circ y_i^{hp} = \tilde{o}_i^{hp}.$$

So these local outputs are also invariant to a global rigid motion. Transforming to global coordinates and back serves to:

- pool information from neighbors in physical 3D space, and
- express that pooled information in a coordinate-free way (in each residue's local frame).

10 Unified Mental Model: Dot Products and Distances Together

We can now give a compact conceptual model.

For each head h, the attention score from residue i to residue j can be written as

$$\text{score}_{ij}^h = \text{content\_similarity}(i,j) + \text{pair\_bias}(i,j) + \text{geometry\_similarity}(i,j),$$

where:

- content_similarity(i,j) is the scaled dot product of queries and keys,
- pair_bias(i,j) comes from the pair embedding z_{ij},
- geometry_similarity(i,j) is a negative squared distance between geometric features of i and j in 3D.

We still:

1. Plug scores into a softmax over j to get weights,
2. Use those weights to form weighted sums of scalar and point-valued outputs.

This is exactly what every attention head does, now with a richer scoring function.

Frames and local/global transforms enable:

- geometry-based terms to depend on true 3D distances, and
- invariance of all such terms to global rotations and translations.

If we keep this as the core picture, much of the notation in the AF2 supplement is just implementation detail for:

- how local points are mapped to global points and back, and
- how distances and averages are computed in a way that respects SE(3) symmetry.