

---

---

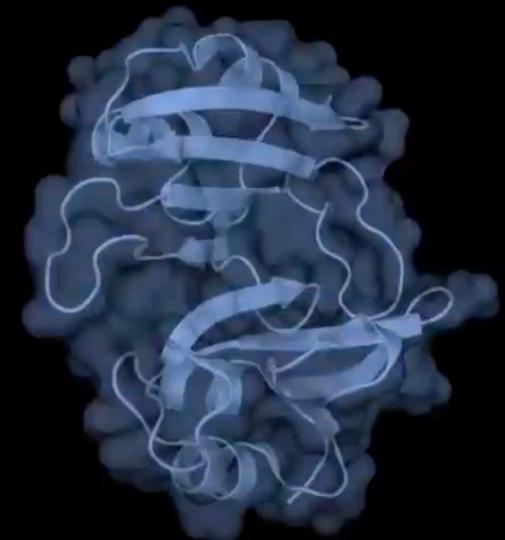
# Highly accurate protein structure prediction with AlphaFold (Part 1)

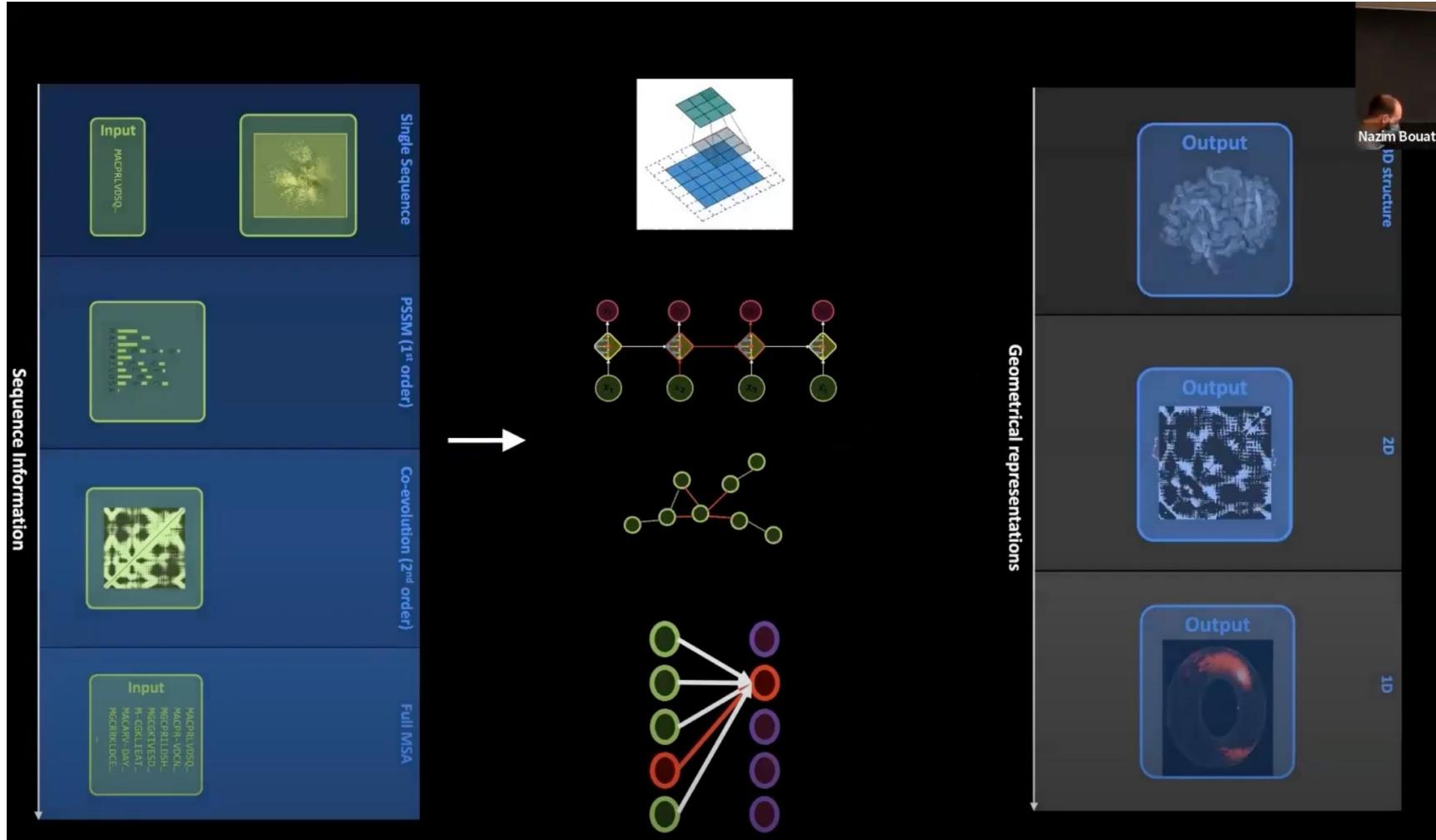
Jumper et al.

---

---

**FIQTRATIPIYYDQDIMFLECRG ...**



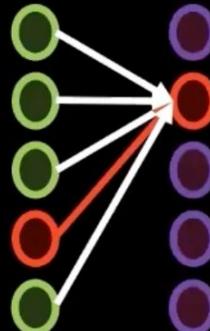


# Input

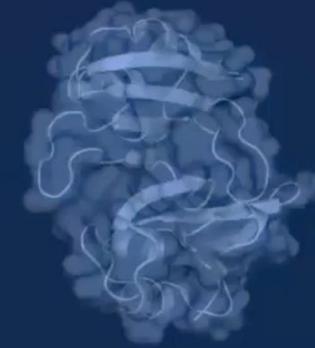
MACPRLVDSQ...  
MACPR-VDCN...  
MGCPRILDSH...  
MGCGKIVESD...  
M-CGKLIEAT...  
MACARV-DAY...  
MGCRRKLDCE...

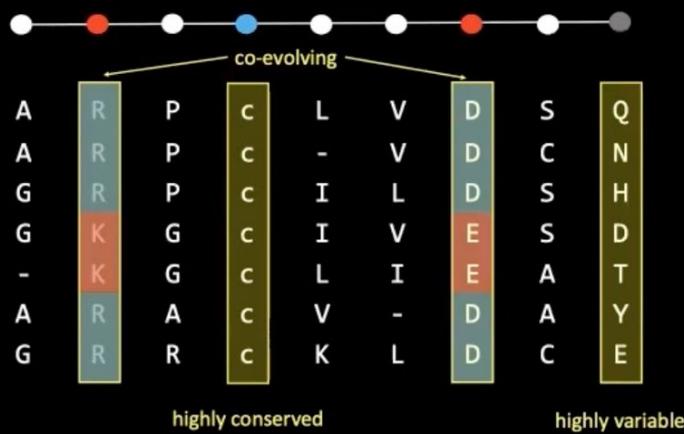


Transformer Networks



# Output

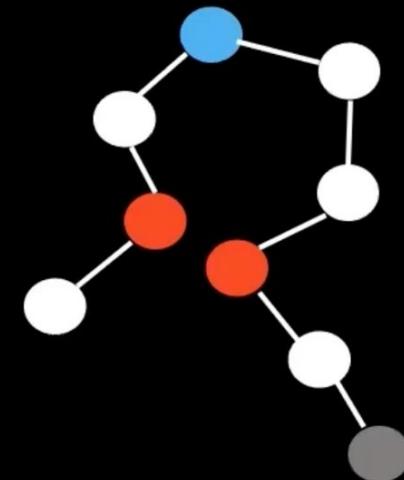


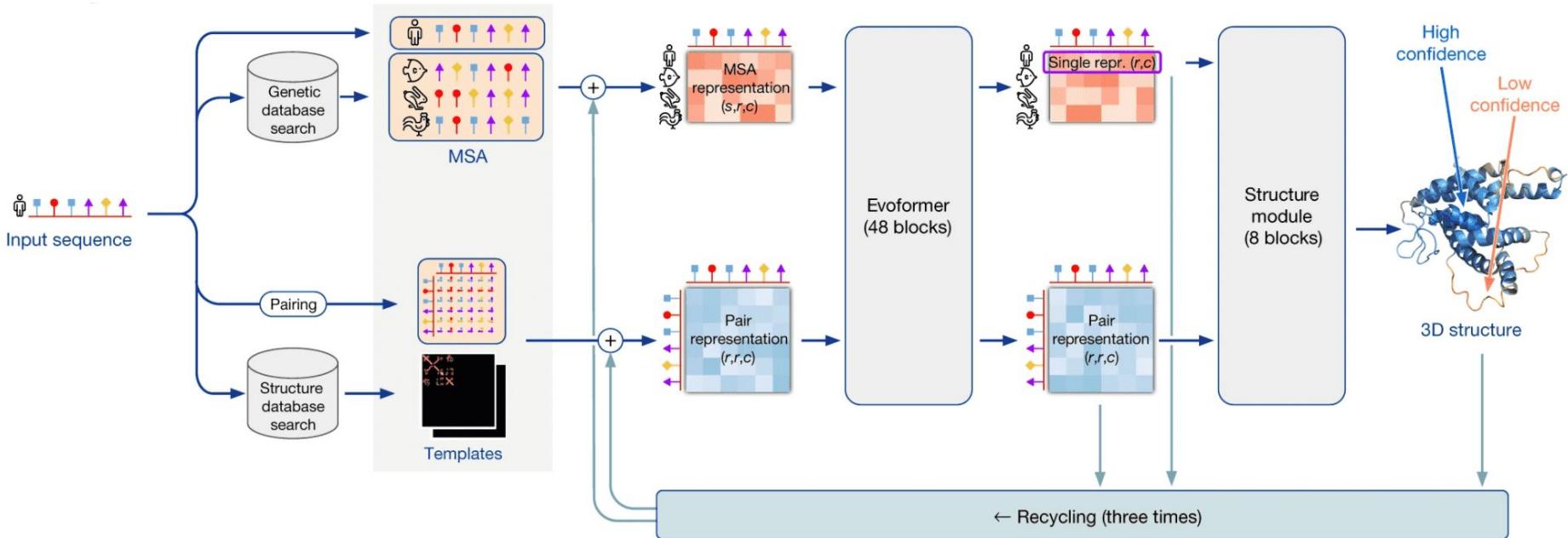


## Inference



## Constraint





Input:

Cogito

ergo

sum

Nazim Bouatta

Embeddings:

$x_1$



$x_2$



$x_3$



Queries:

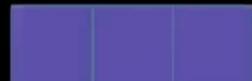
$q_1$



$q_2$

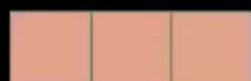


$q_3$

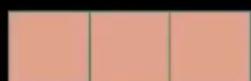


Keys:

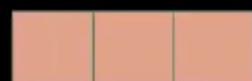
$k_1$



$k_2$

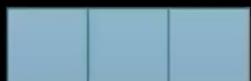


$k_3$



Values:

$v_1$



$v_2$



$v_3$



Input:

Cogito

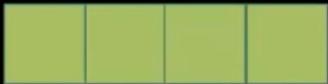
ergo

sum

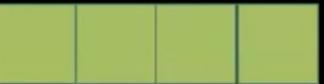
Nazim Bouatta

Embeddings:

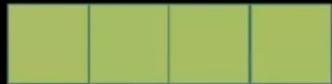
$x_1$



$x_2$



$x_3$



Queries:

$q_1$



What am I looking for?

$q_3$



Keys:

$k_1$



Who am I?

$k_3$



Values:

$v_1$



What do I have to say?

$v_3$



Queries:

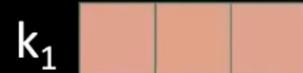


What am I looking for?

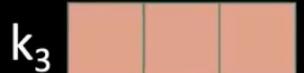


Nazim Bouatta

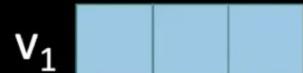
Keys:



Who am I?



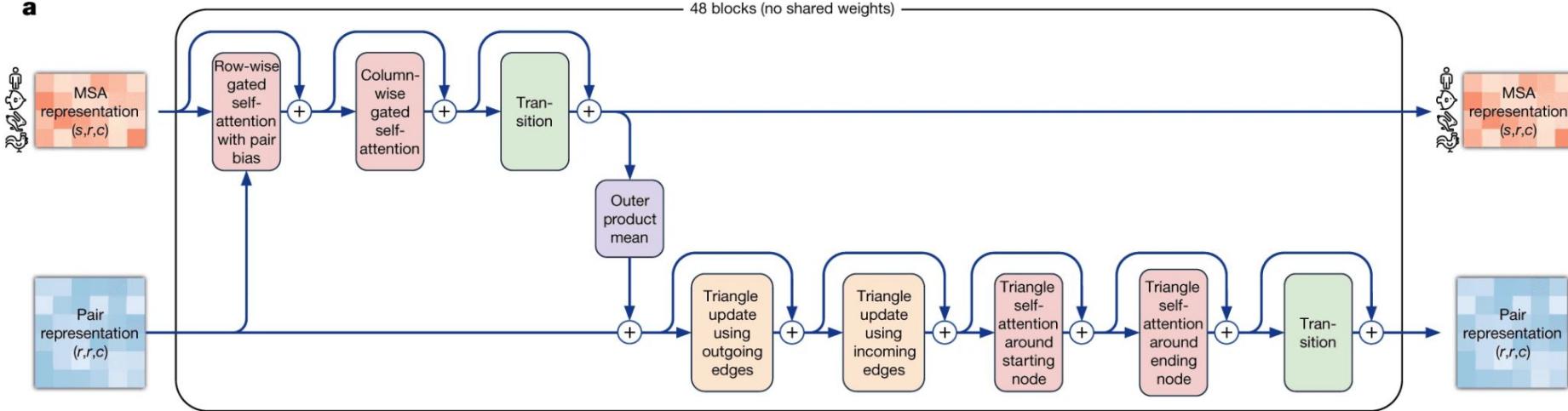
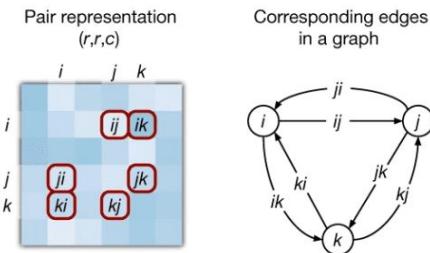
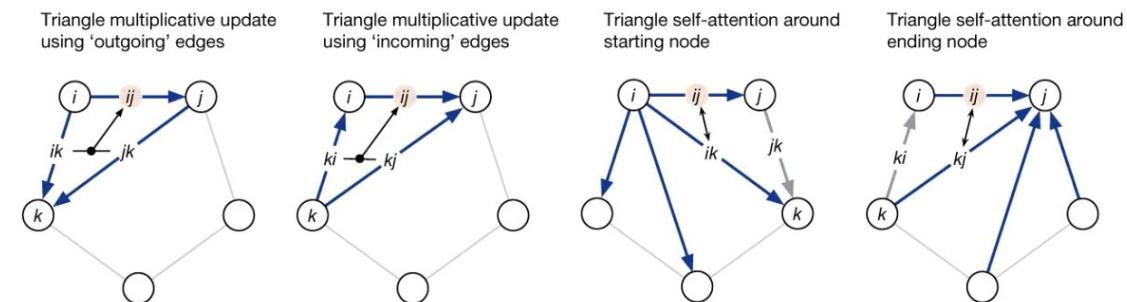
Values:

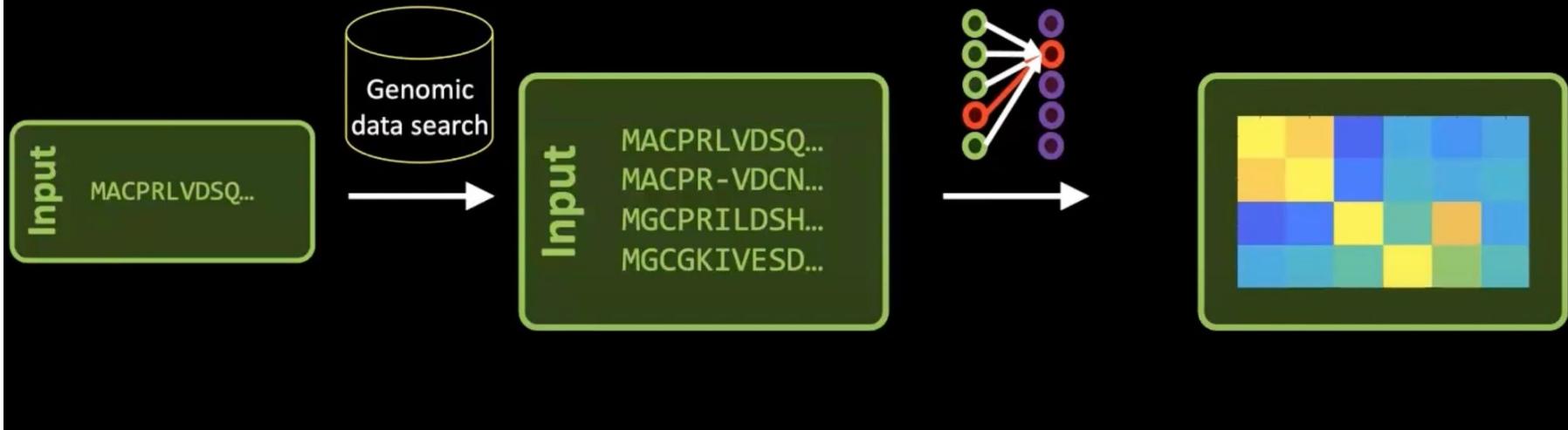


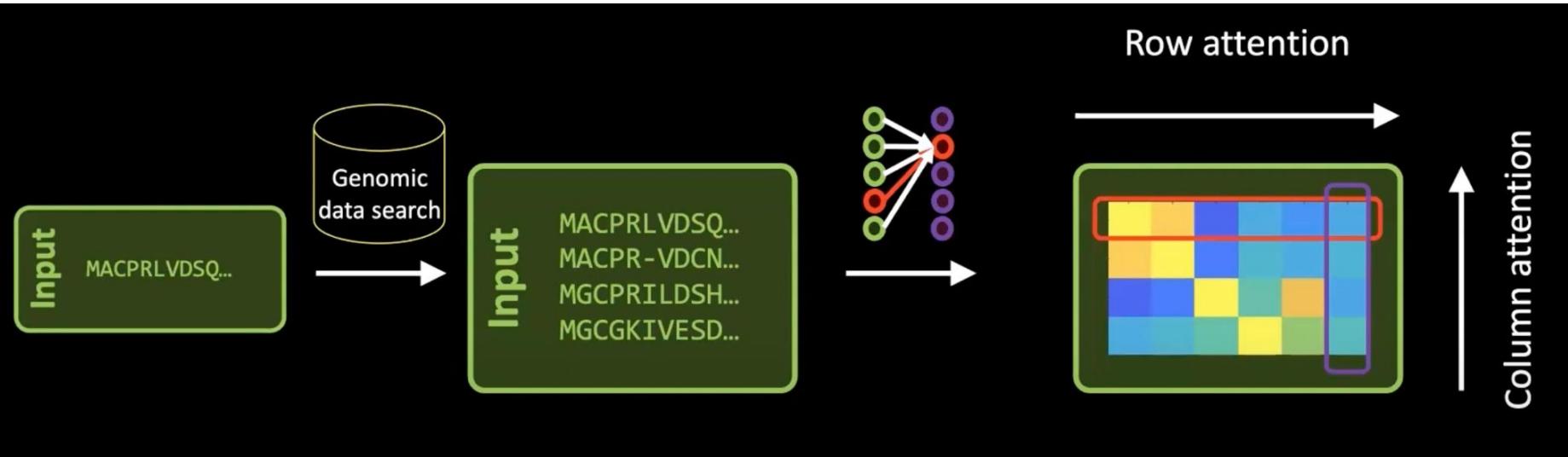
What do I have to say?



$$\text{softmax}\left(q_i k_j\right) v_j$$

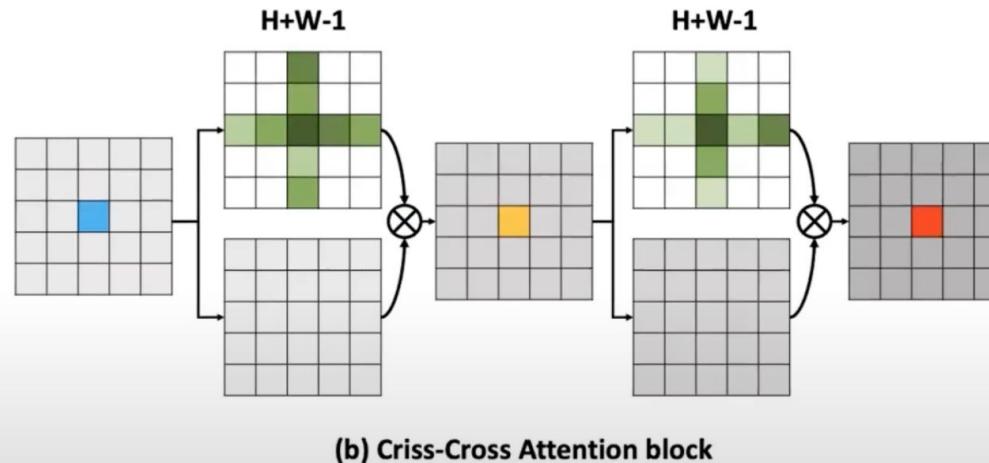
**a****b****c**



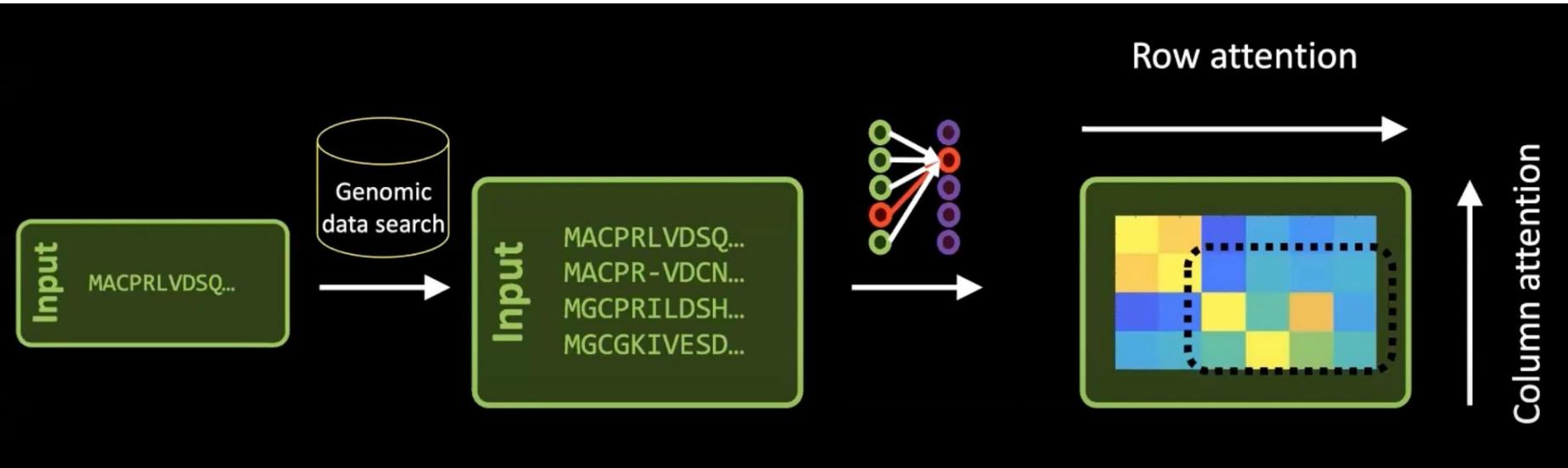


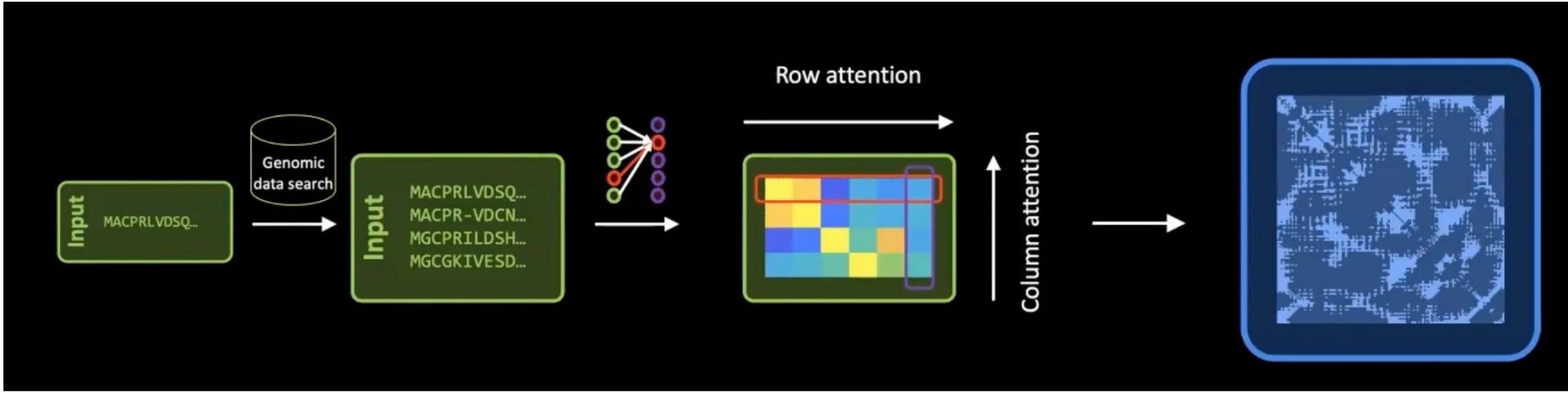
## CCNet: Criss-Cross Attention for Semantic Segmentation

Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, Thomas S. Huang



# Axial Attention

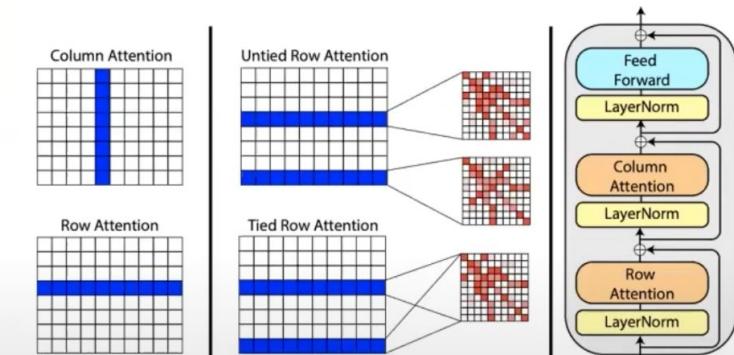




## MSA Transformer

✉ Roshan Rao, Jason Liu, Robert Verkuil, ✉ Joshua Meier, ✉ John F. Canny, Pieter Abbeel, ✉ Tom Sercu,  
✉ Alexander Rives

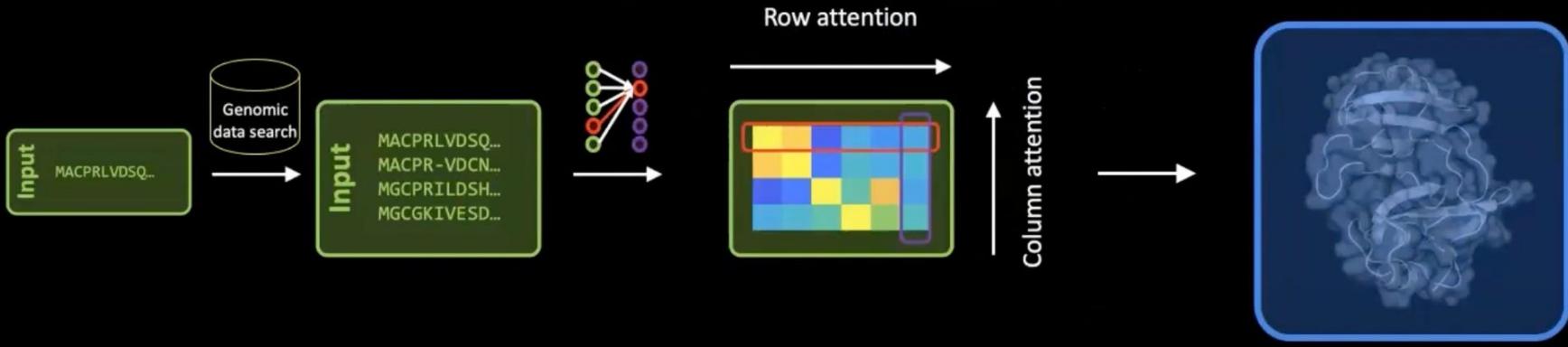
doi: <https://doi.org/10.1101/2021.02.12.430858>



# MSA Transformer

**Table 3.** Supervised contact prediction on CASP13 and CAMEO (long-range precision). \*Uses outer-concatenation of the query sequence representation as features. †Additionally uses the row attention maps as features.

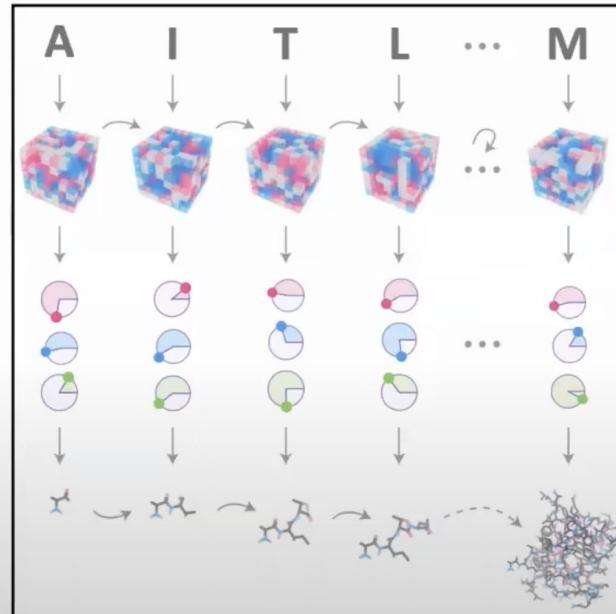
Model	CASP13-FM		CAMEO	
	L	L/5	L	L/5
trRosetta <sub>base</sub>	45.7	69.6	50.9	75.5
trRosetta <sub>full</sub>	51.8	80.1	53.2	77.5
Co-evolutionary	40.1	65.2	47.3	72.1
ProTrans-T5	25.0	41.4	40.8	63.3
ESM-1b	28.2	50.2	44.4	68.4
MSA Transformer*	54.5	<b>80.2</b>	53.6	78.0
MSA Transformer†	<b>54.6</b>	77.5	<b>55.8</b>	<b>79.1</b>



# Cell Systems

## End-to-End Differentiable Learning of Protein Structure

### Graphical Abstract



### Authors

Mohammed AlQuraishi

### Correspondence

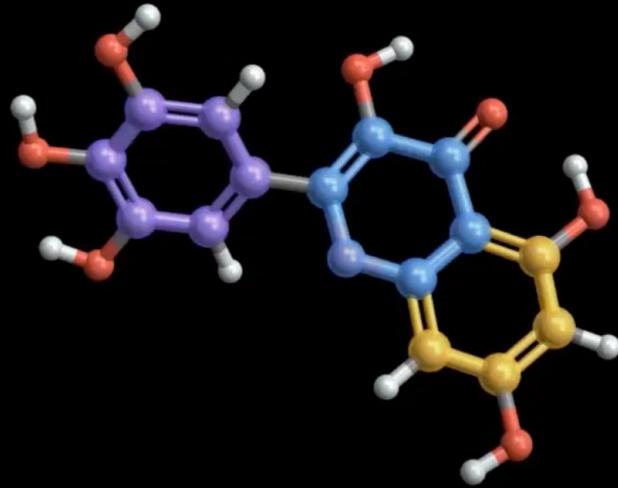
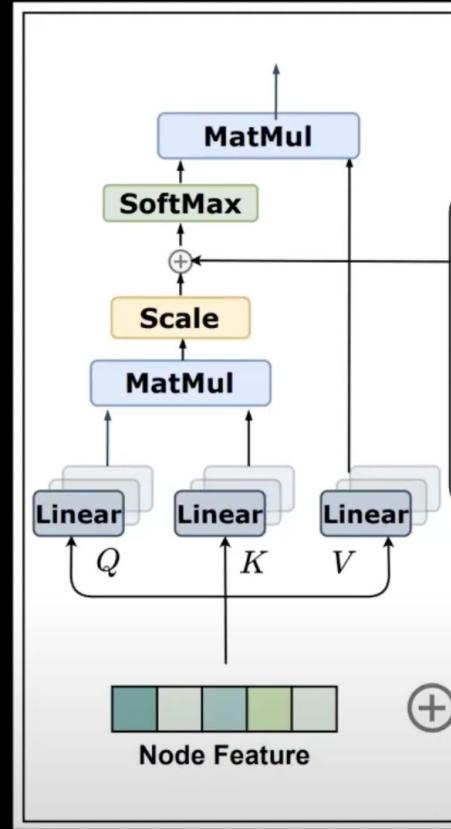
alquraishi@hms.harvard.edu

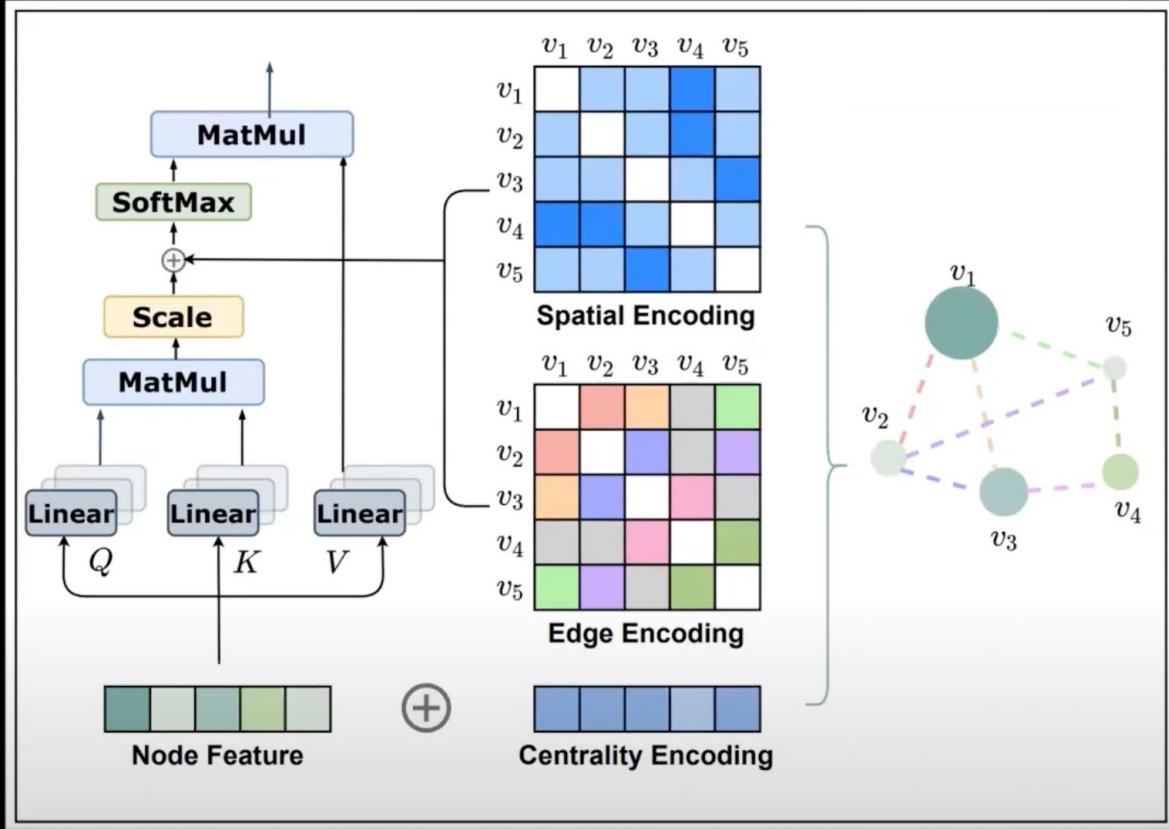
### In Brief

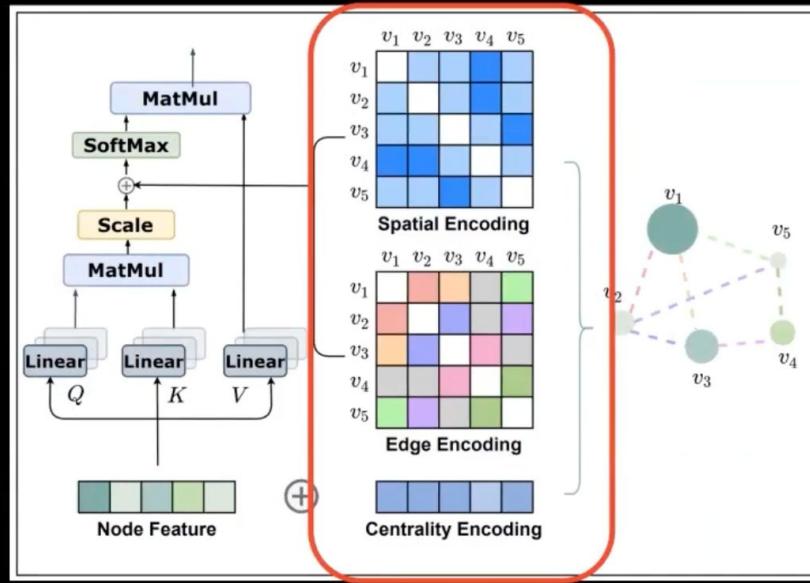
Prediction of protein structure from sequence is important for understanding protein function, but it remains very challenging, especially for proteins with few homologs. Existing prediction methods are human engineered, with many complex parts developed over decades. We introduce a new approach based entirely on machine learning that predicts protein structure from sequence using a single neural network. The model achieves state-of-the-art accuracy and does not require co-evolution information or structural homologs. It is also much faster, making predictions in milliseconds versus hours or days, which enables new applications in drug discovery and protein design.

### Highlights

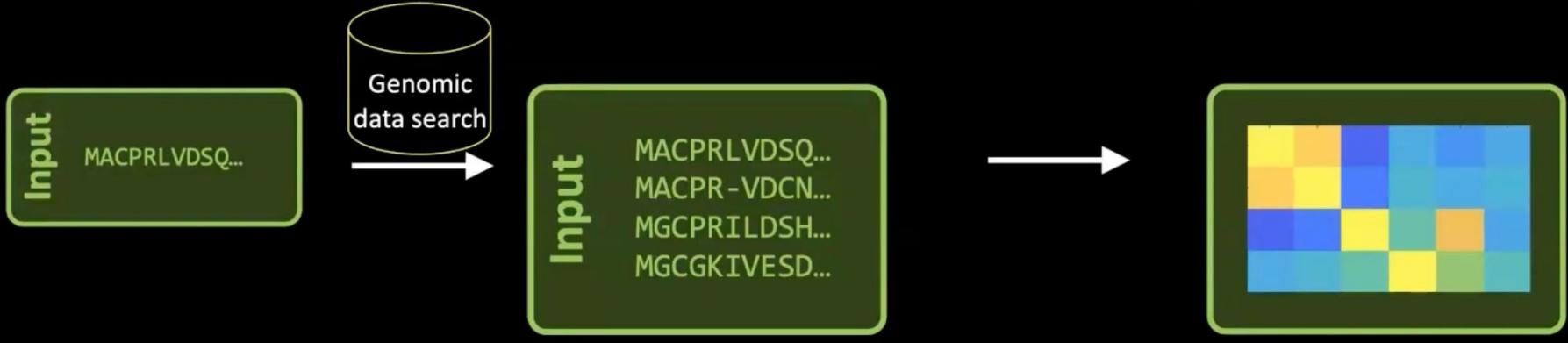
- Neural network predicts protein structure from sequence without using co-evolution

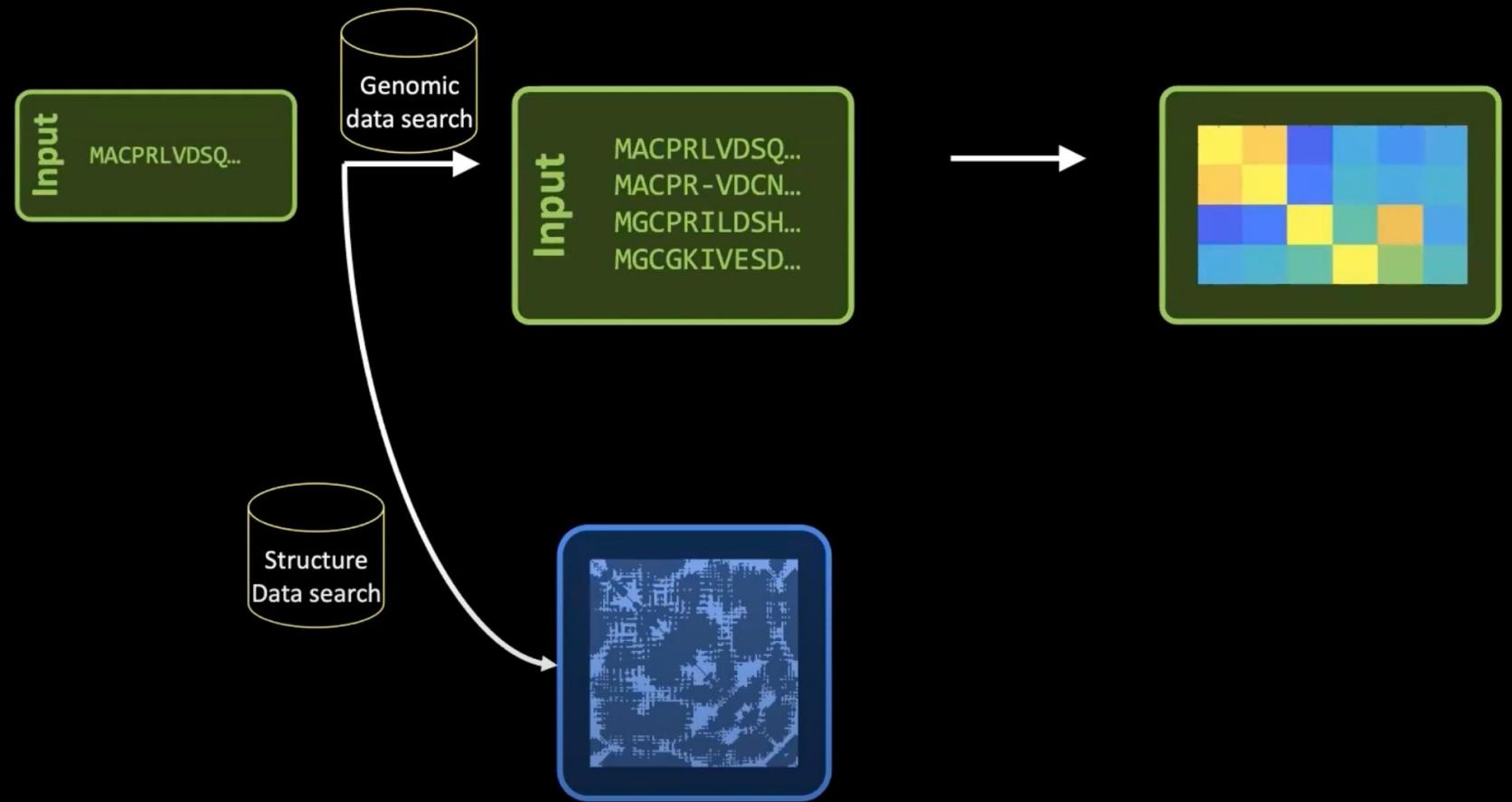


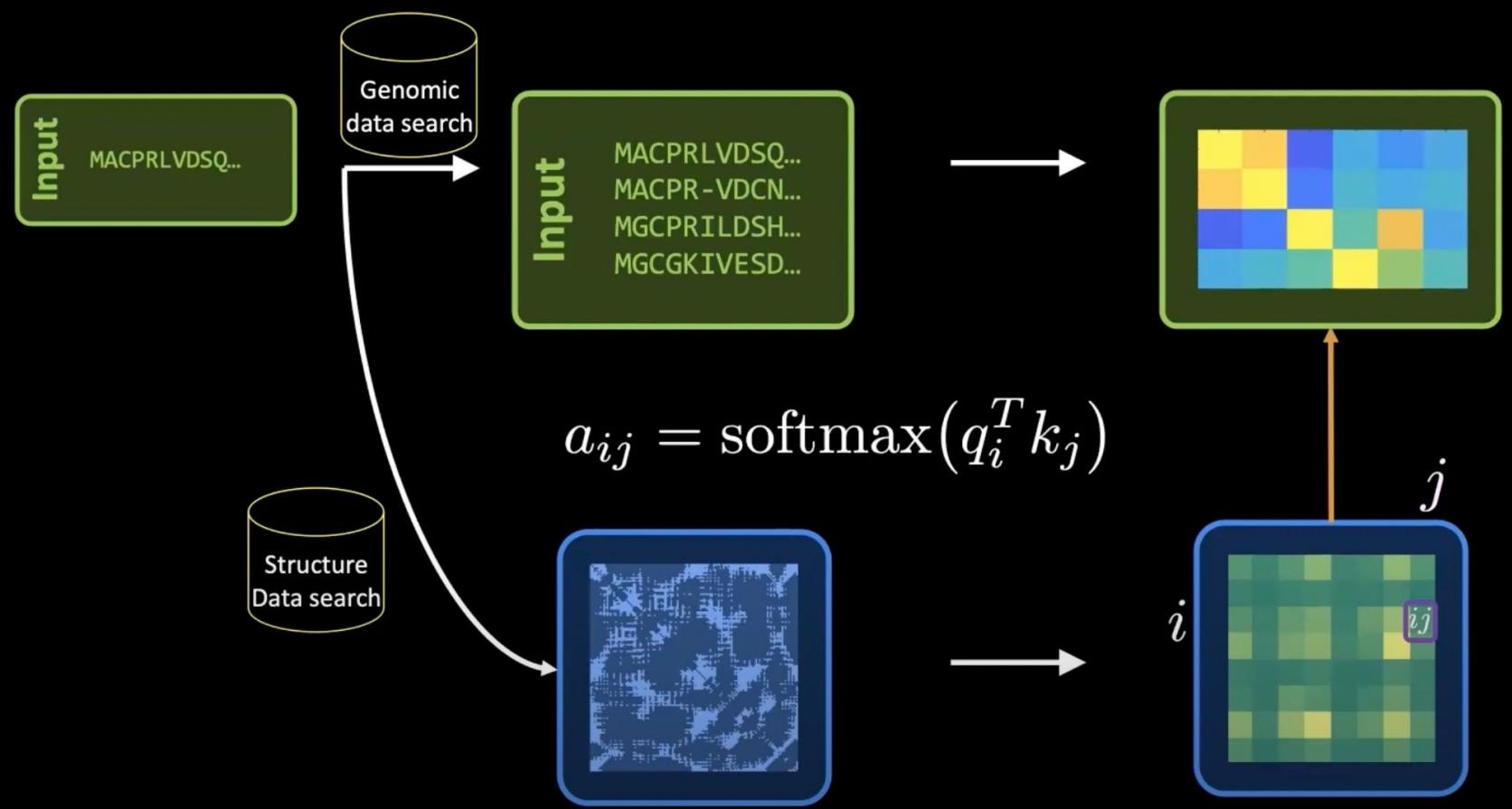


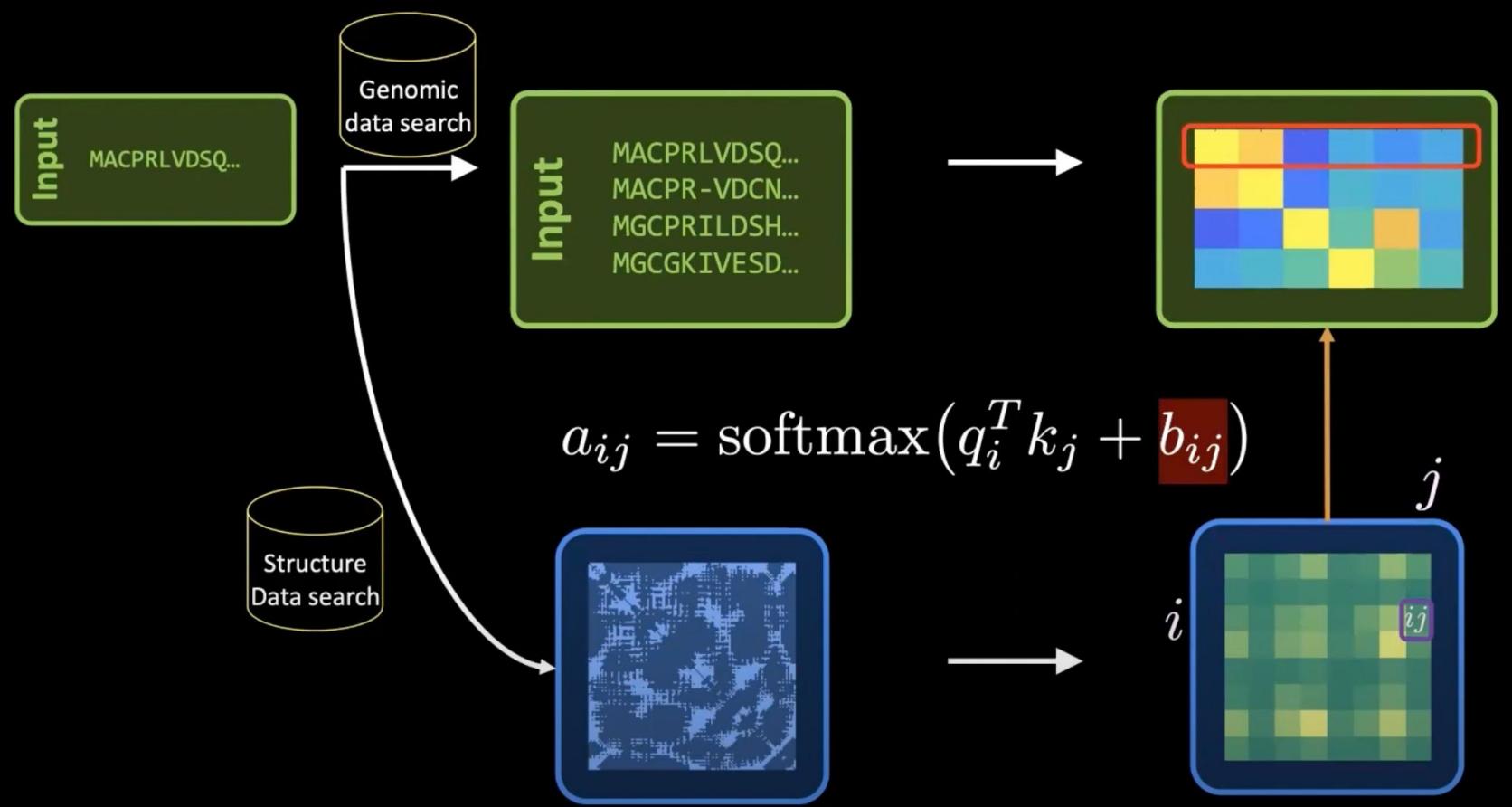


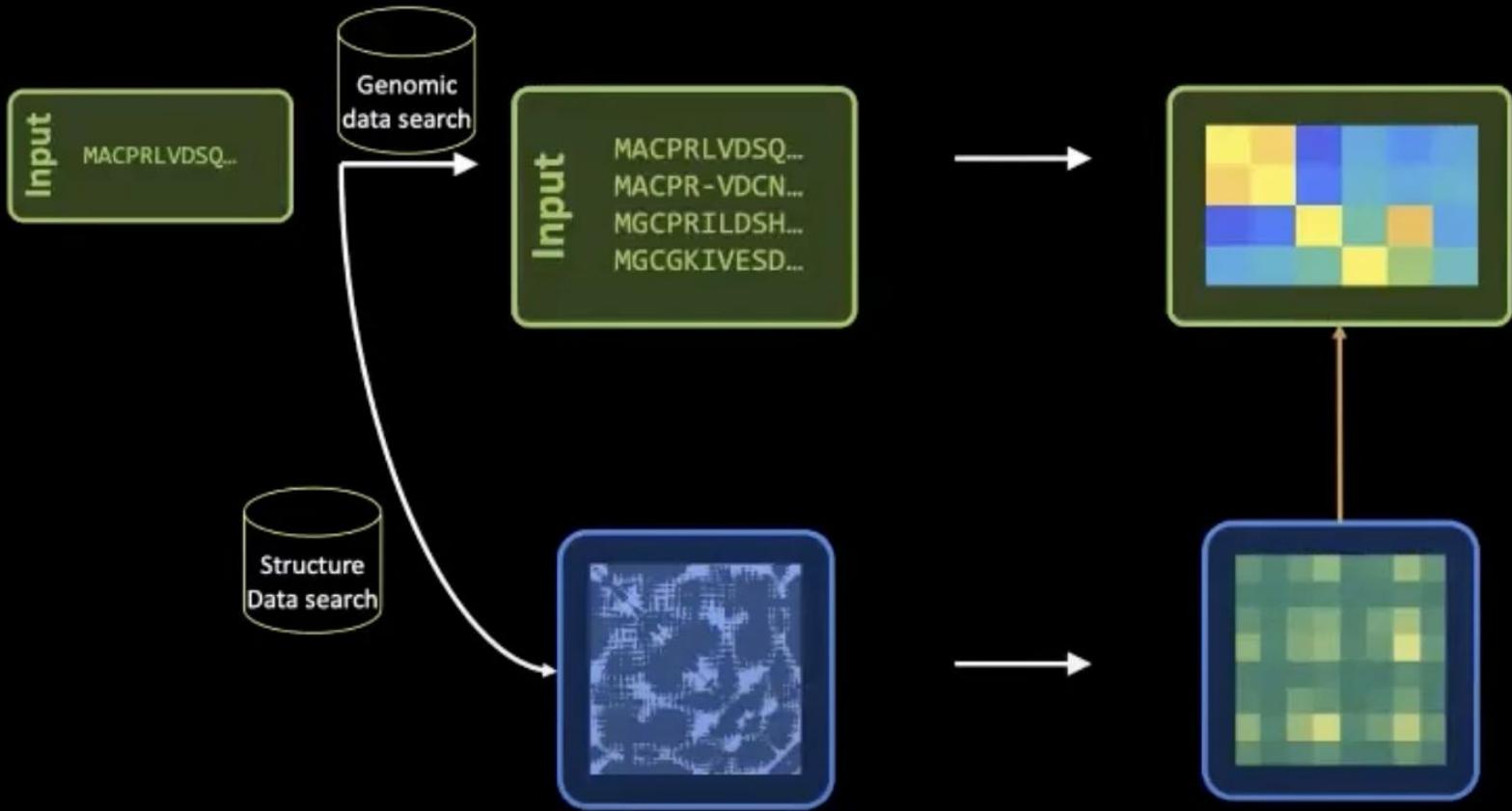
$$a_{ij} = \text{softmax}(q_i^T k_j + b_{ij})$$

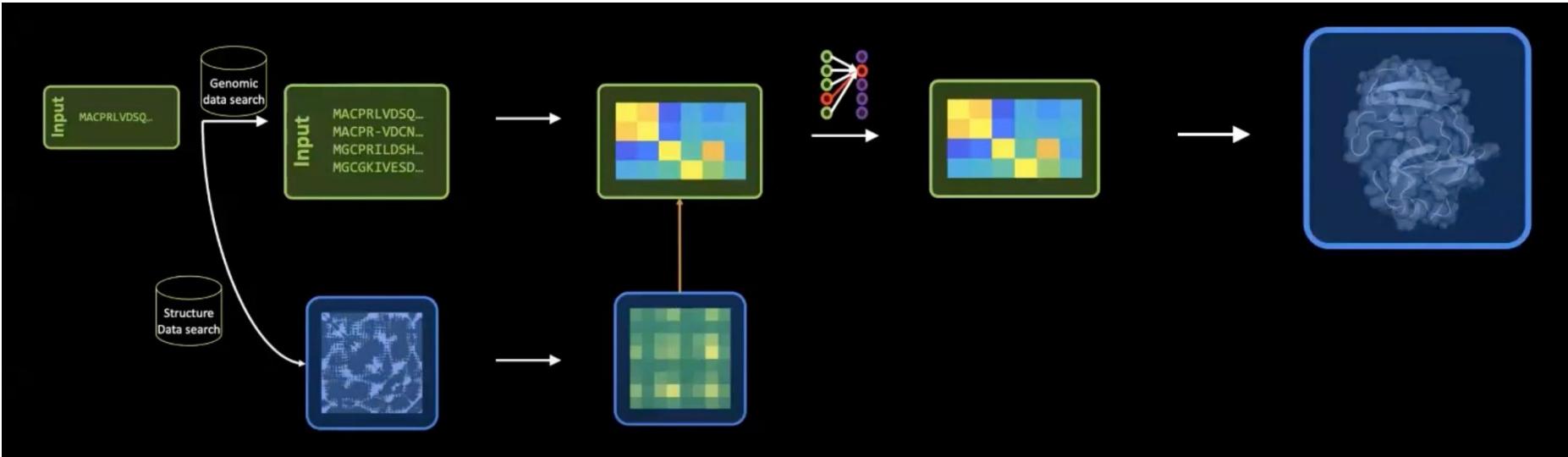


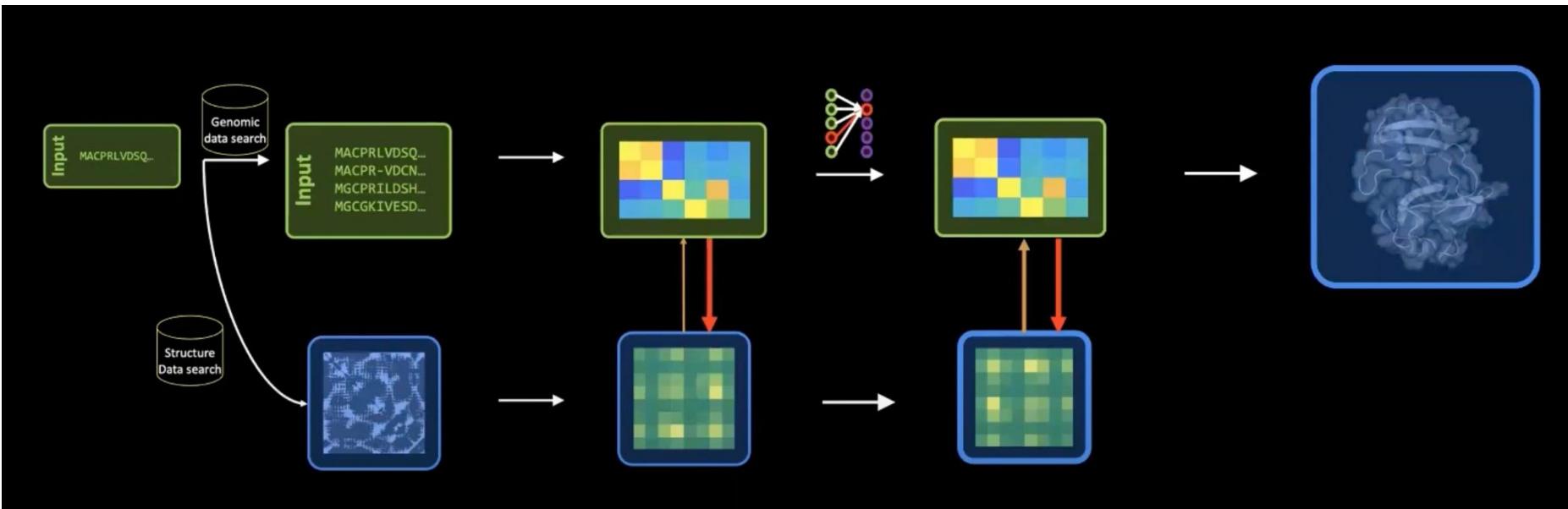


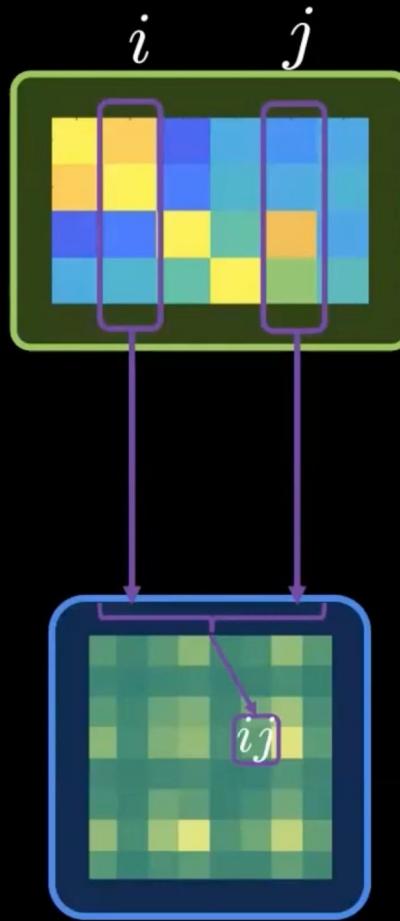




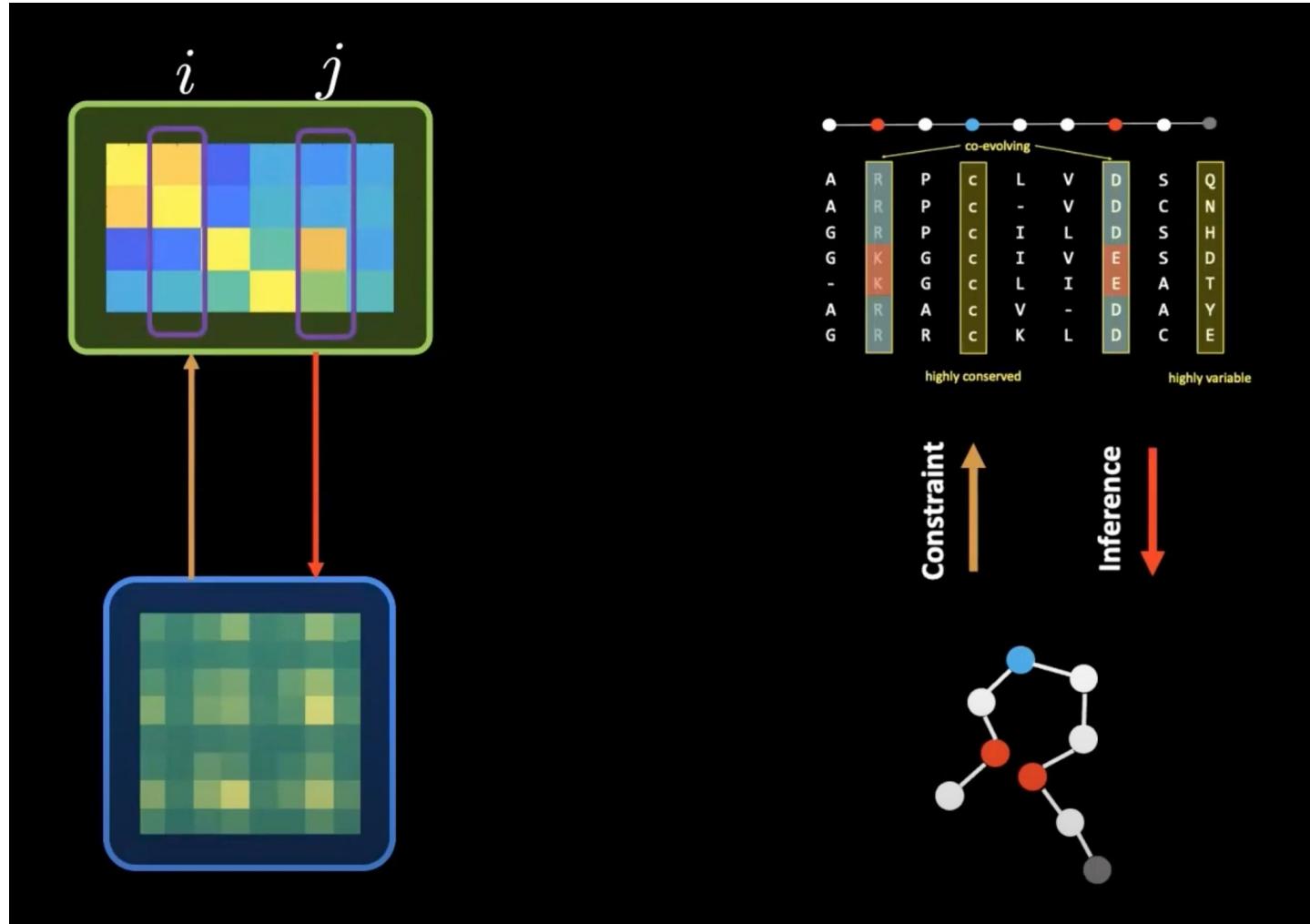


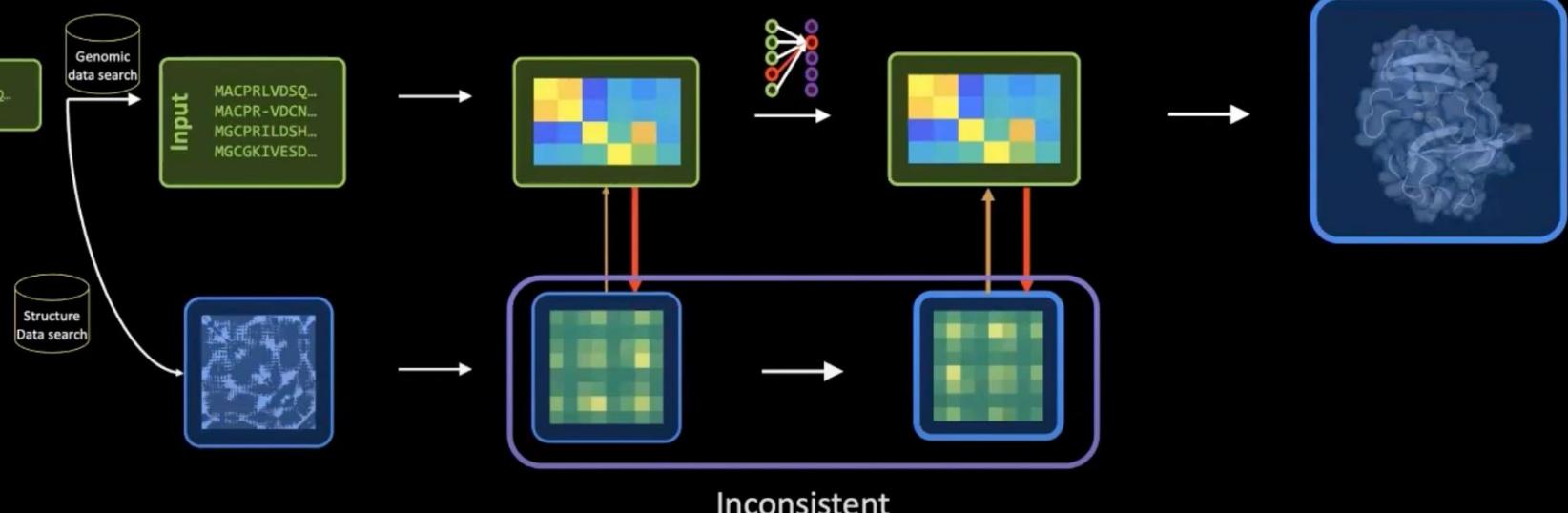


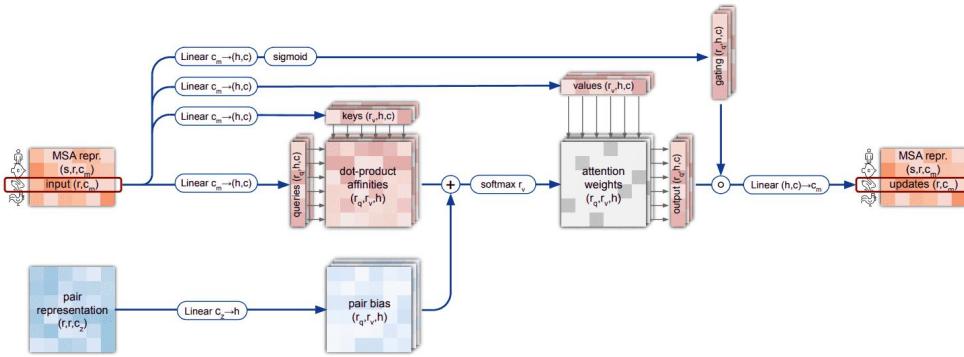




$$o_{ij} = \text{mean}(a_i \otimes b_j)$$







**Supplementary Figure 2** | MSA row-wise gated self-attention with pair bias. Dimensions: s: sequences, r: residues, c: channels, h: heads.

---

#### Algorithm 7 MSA row-wise gated self-attention with pair bias

---

**def** MSARowAttentionWithPairBias( $\{\mathbf{m}_{si}\}, \{\mathbf{z}_{ij}\}$ ,  $c = 32, N_{\text{head}} = 8$ ) :

*# Input projections*

1:  $\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

2:  $\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h = \text{LinearNoBias}(\mathbf{m}_{si})$

$$\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h \in \mathbb{R}^c, h \in \{1, \dots, N_{\text{head}}\}$$

3:  $b_{ij}^h = \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$

4:  $\mathbf{g}_{si}^h = \text{sigmoid}(\text{Linear}(\mathbf{m}_{si}))$

$$\mathbf{g}_{si}^h \in \mathbb{R}^c$$

*# Attention*

$$5: a_{sij}^h = \text{softmax}_j \left( \frac{1}{\sqrt{c}} \mathbf{q}_{si}^h \top \mathbf{k}_{sj}^h + b_{ij}^h \right)$$

$$6: \mathbf{o}_{si}^h = \mathbf{g}_{si}^h \odot \sum_j a_{sij}^h \mathbf{v}_{sj}^h$$

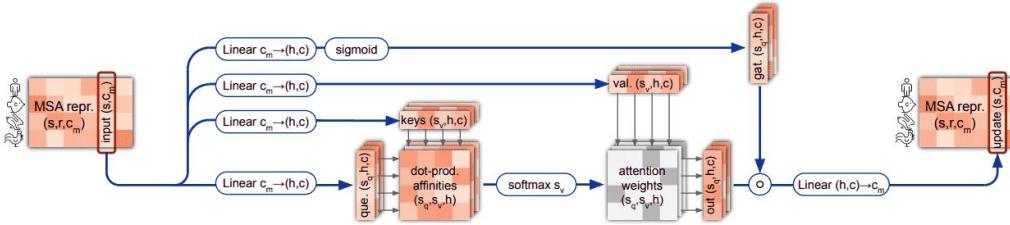
*# Output projection*

$$7: \tilde{\mathbf{m}}_{si} = \text{Linear} \left( \text{concat}_h(\mathbf{o}_{si}^h) \right)$$

$$\tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$$

8: **return**  $\{\tilde{\mathbf{m}}_{si}\}$

---



**Supplementary Figure 3 | MSA column-wise gated self-attention.** Dimensions: s: sequences, r: residues, c: channels, h: heads.

---

#### Algorithm 8 MSA column-wise gated self-attention

---

**def** MSAColumnAttention( $\{\mathbf{m}_{si}\}$ ,  $c = 32$ ,  $N_{\text{head}} = 8$ ) :

*# Input projections*

1:  $\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

2:  $\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h = \text{LinearNoBias}(\mathbf{m}_{si})$

$\mathbf{q}_{si}^h, \mathbf{k}_{si}^h, \mathbf{v}_{si}^h \in \mathbb{R}^c, h \in \{1, \dots, N_{\text{head}}\}$

3:  $\mathbf{g}_{si}^h = \text{sigmoid}(\text{Linear}(\mathbf{m}_{si}))$

$\mathbf{g}_{si}^h \in \mathbb{R}^c$

*# Attention*

4:  $a_{sti}^h = \text{softmax}_t\left(\frac{1}{\sqrt{c}} \mathbf{q}_{si}^{h\top} \mathbf{k}_{ti}^h\right)$

5:  $\mathbf{o}_{si}^h = \mathbf{g}_{si}^h \odot \sum_t a_{sti}^h \mathbf{v}_{st}^h$

*# Output projection*

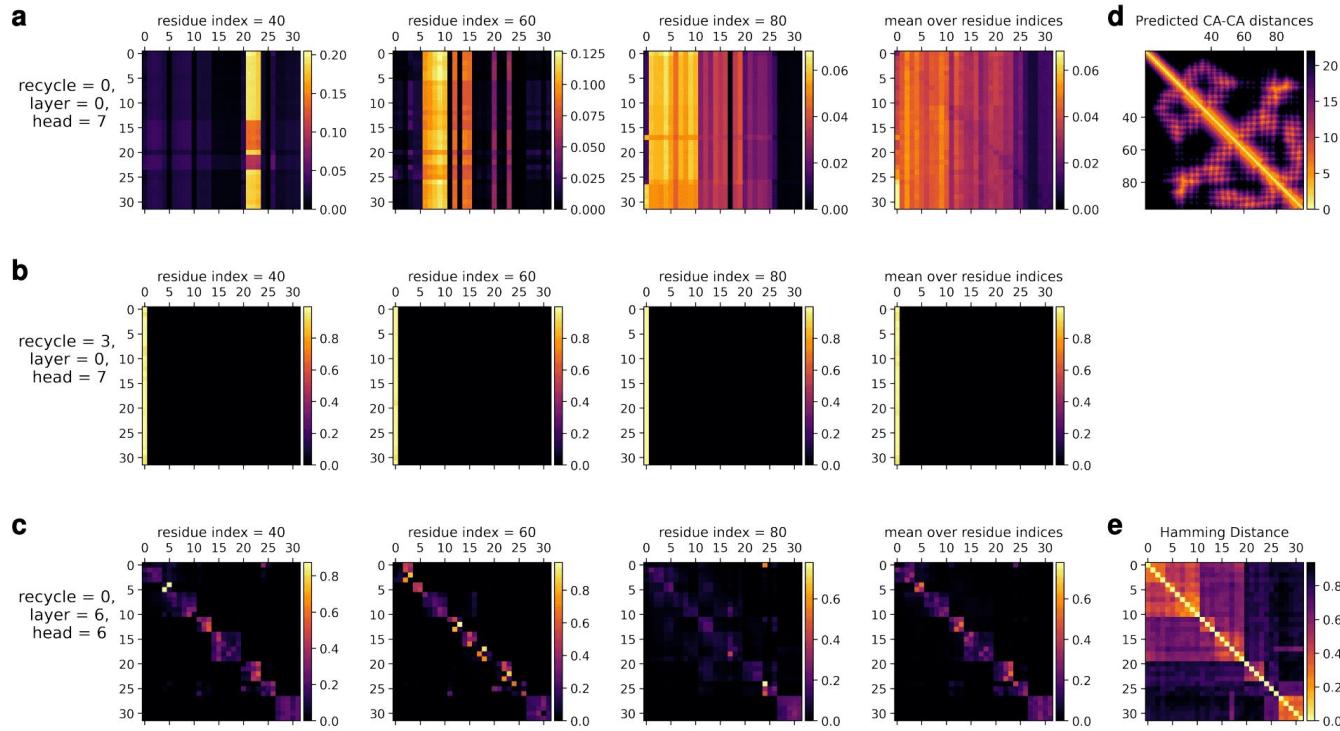
6:  $\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{si}^h)\right)$

$\tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

7: **return**  $\{\tilde{\mathbf{m}}_{si}\}$

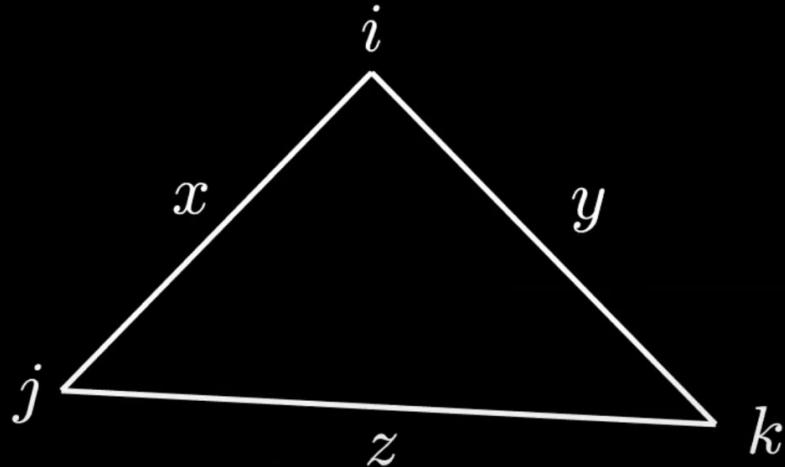
---

MSA Columnwise Attention: CASP14 target T1082

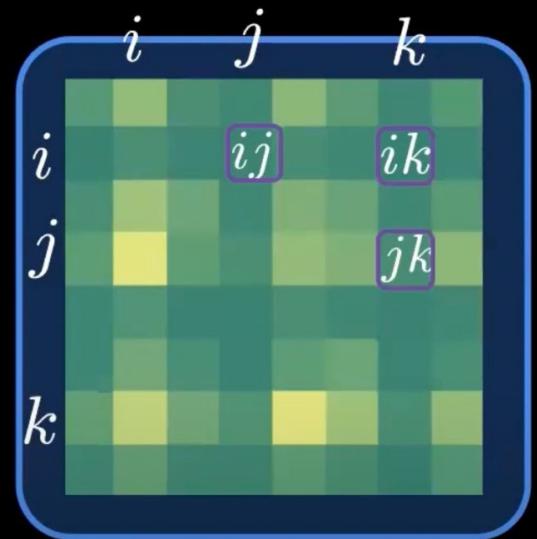


**Supplementary Figure 13 |** Visualization of attention in the MSA along sequences. **(a)** Attention patterns in layer 0, head 7. **(b)** Attention patterns in layer 0, head 7 in the last recycling iteration. **(c)** Attention patterns in layer 6, head 6. **(d)** Predicted  $C\alpha - C\alpha$  distances **(e)** Hamming distances between the sequences of the MSA.

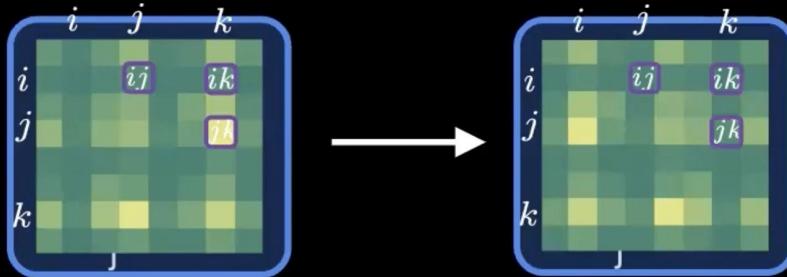
# Triangular inequality



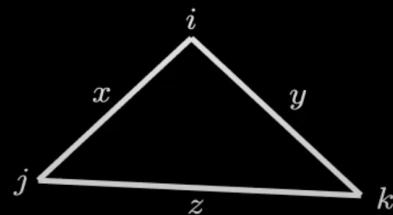
$$z \leq x + y$$



# Triangular inequality

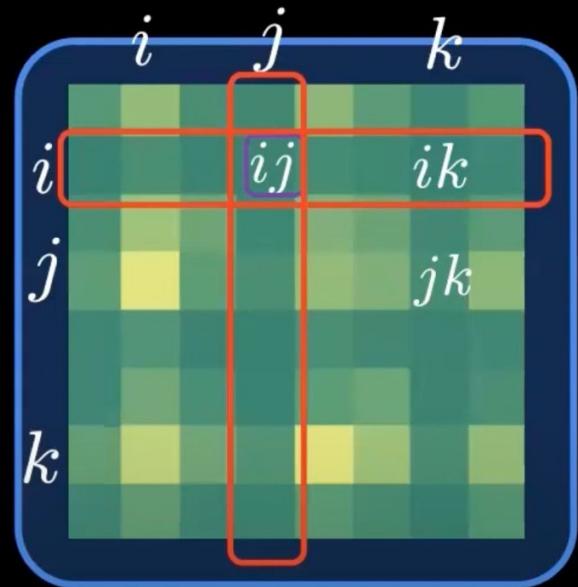


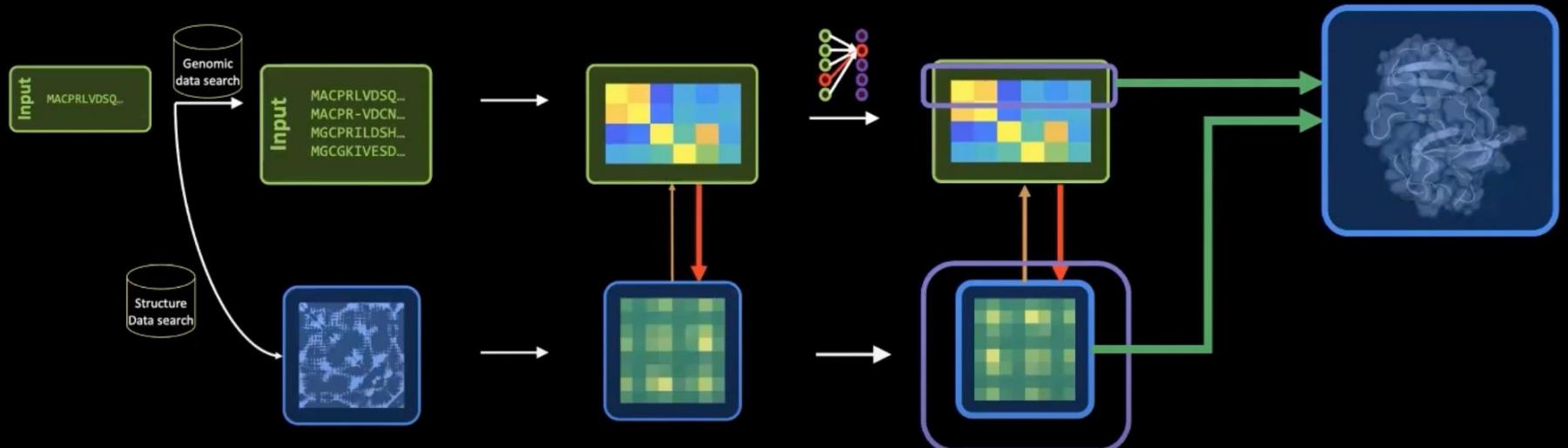
$$z > x + y$$

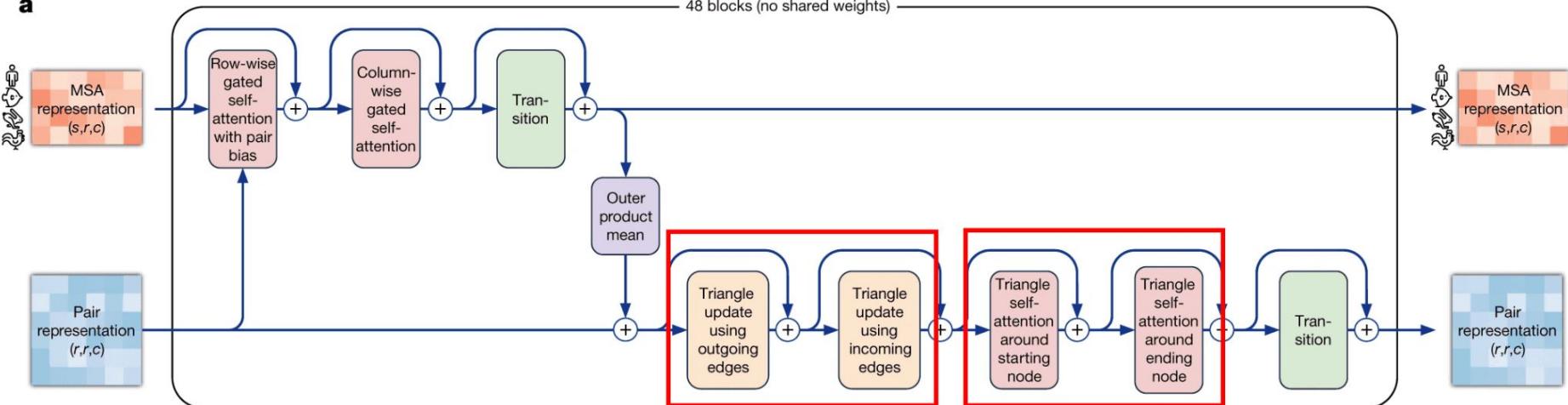


# Triangular inequality

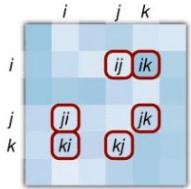
$$z_{ij} \leftarrow f\left(\sum_k a_{ik} b_{jk}\right)$$



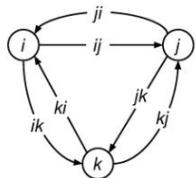


**a****b**

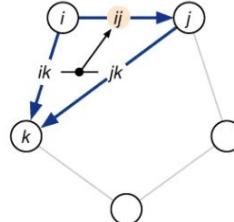
Pair representation  
( $r, r, c$ )



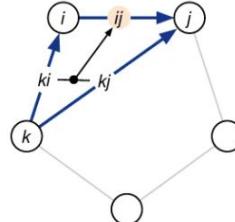
Corresponding edges  
in a graph

**c**

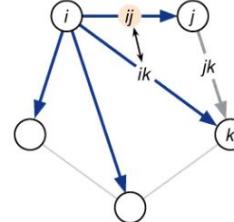
Triangle multiplicative update  
using 'outgoing' edges



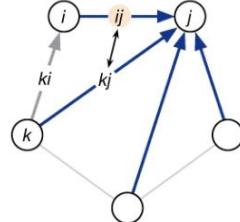
Triangle multiplicative update  
using 'incoming' edges

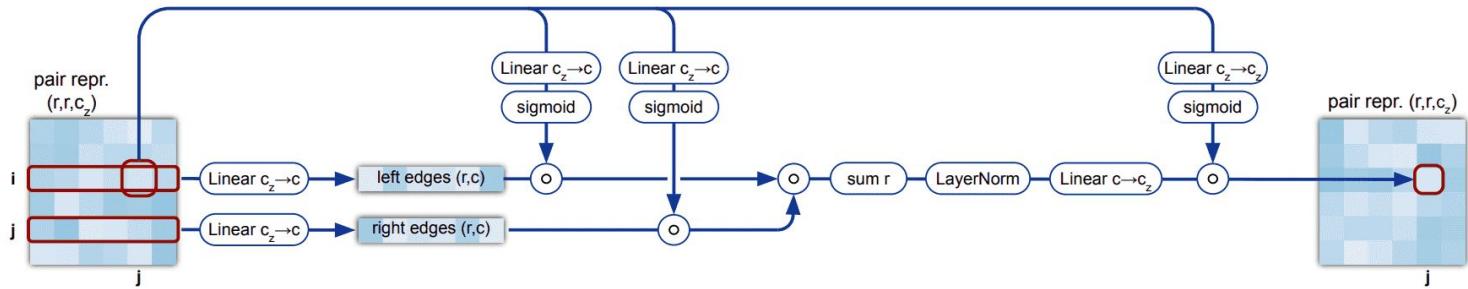


Triangle self-attention around  
starting node



Triangle self-attention around  
ending node





**Supplementary Figure 6** | Triangular multiplicative update using “outgoing” edges. Dimensions:  $r$ : residues,  $c$ : channels.

---

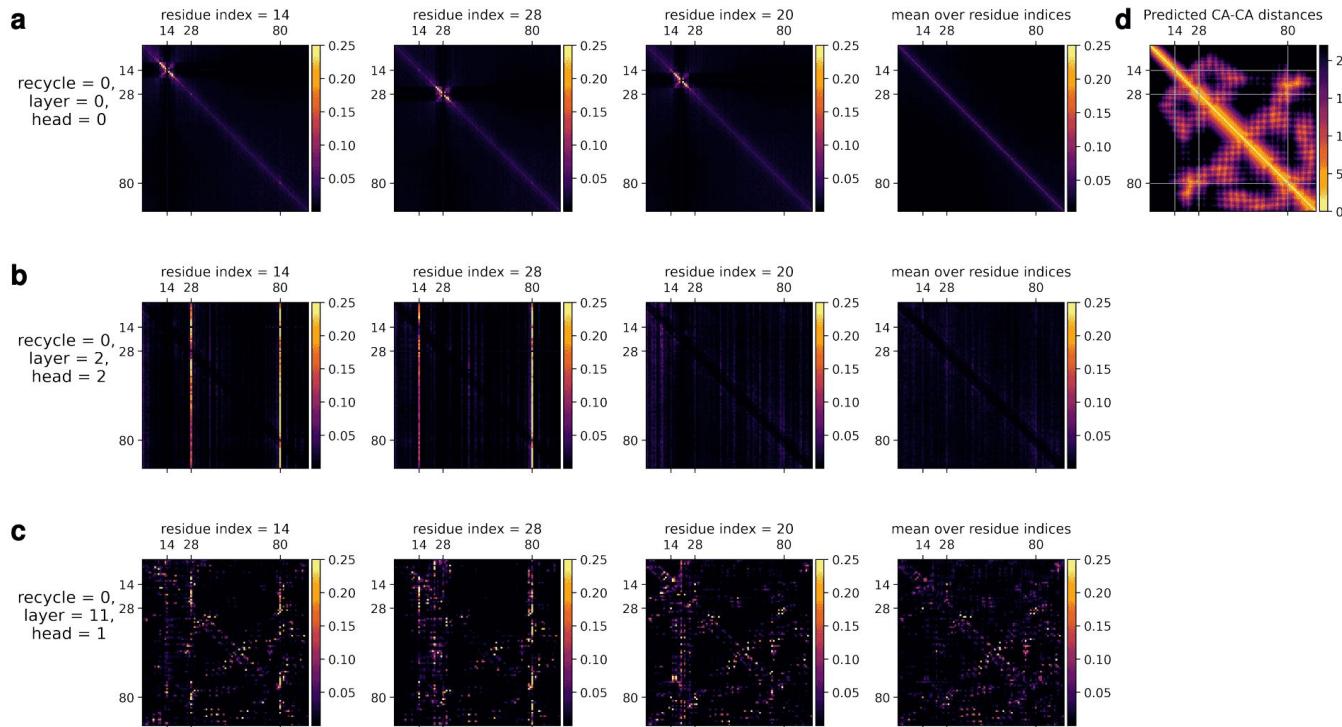
### Algorithm 11 Triangular multiplicative update using “outgoing” edges

---

**def** TriangleMultiplicationOutgoing( $\{\mathbf{z}_{ij}\}$ ,  $c = 128$ ) :

- 1:  $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$
  - 2:  $\mathbf{a}_{ij}, \mathbf{b}_{ij} = \text{sigmoid}(\text{Linear}(\mathbf{z}_{ij})) \odot \text{Linear}(\mathbf{z}_{ij})$   $\mathbf{a}_{ij}, \mathbf{b}_{ij} \in \mathbb{R}^c$
  - 3:  $\mathbf{g}_{ij} = \text{sigmoid}(\text{Linear}(\mathbf{z}_{ij}))$   $\mathbf{g}_{ij} \in \mathbb{R}^{c_z}$
  - 4:  $\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \text{Linear}(\text{LayerNorm}(\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$   $\tilde{\mathbf{z}}_{ij} \in \mathbb{R}^{c_z}$
  - 5: **return**  $\{\tilde{\mathbf{z}}_{ij}\}$
-

### Pair Rowwise Attention: CASP14 target T1082

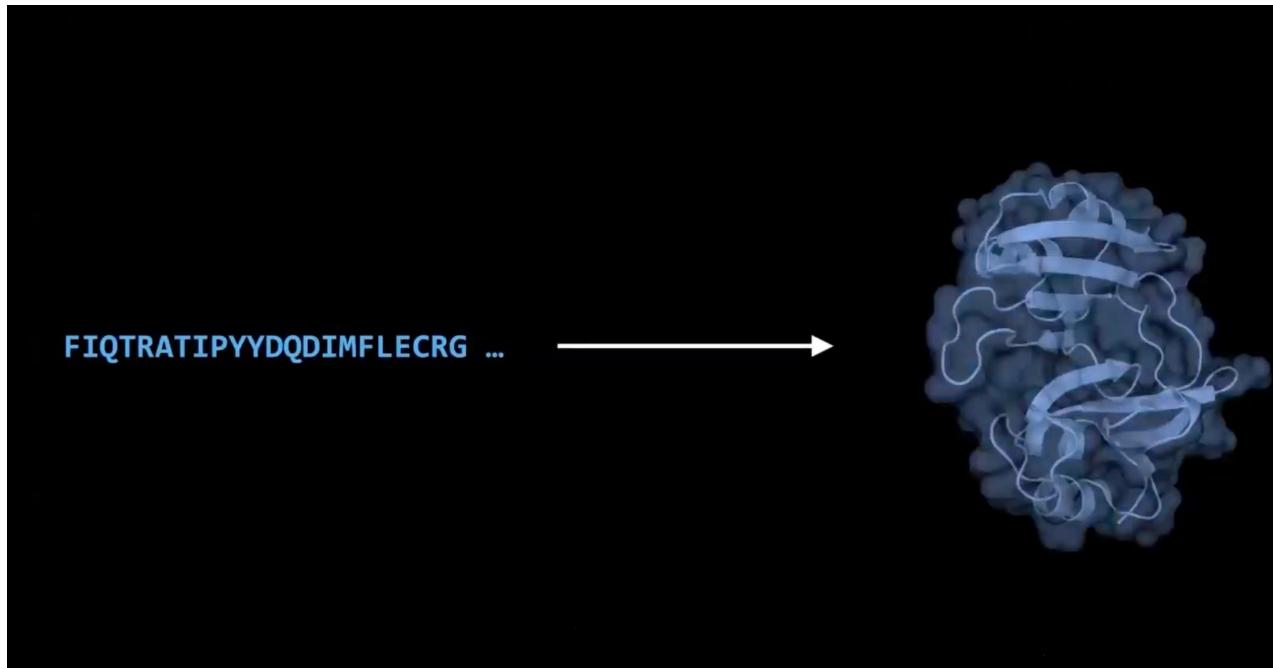


**Supplementary Figure 12 |** Visualization of row-wise pair attention. **(a)** Attention patterns in layer 0, head 0. **(b)** Attention patterns in layer 2, head 2. **(c)** Attention patterns in layer 11, head 1. **(d)** Predicted  $C\alpha - C\alpha$  distances.

# End of Part 1

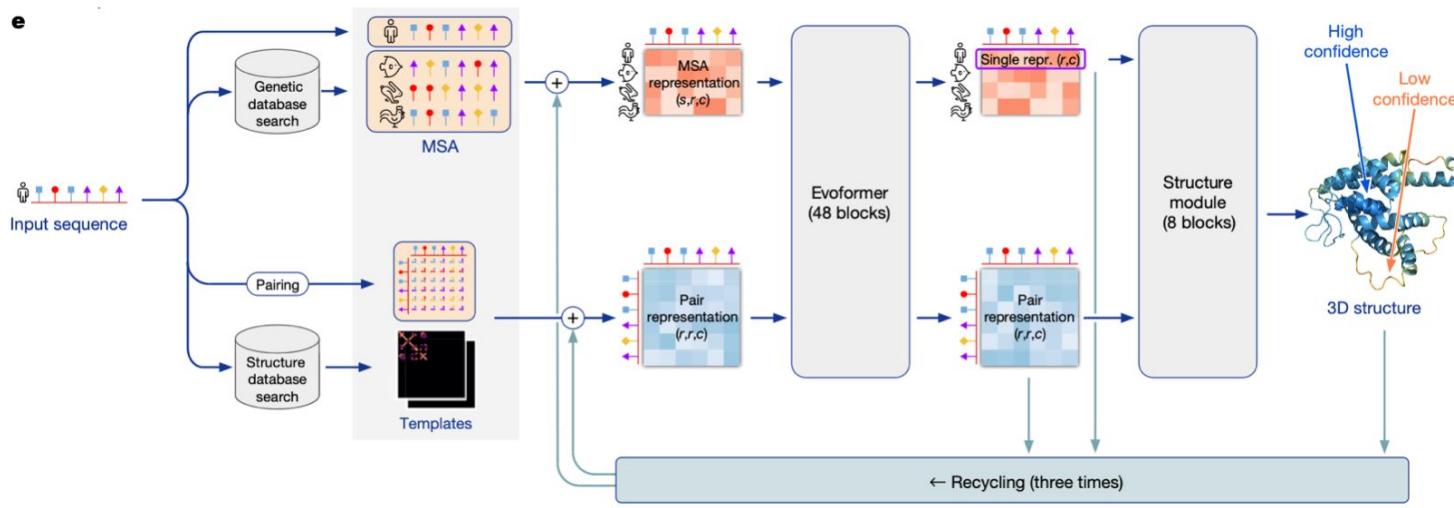
# Previously on

## Highly accurate protein structure prediction with AlphaFold

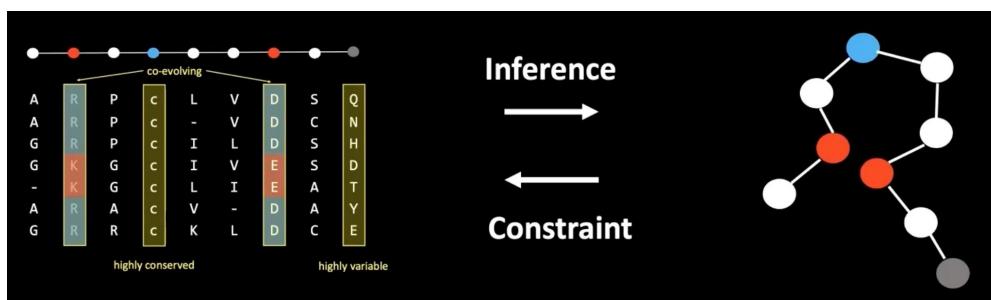


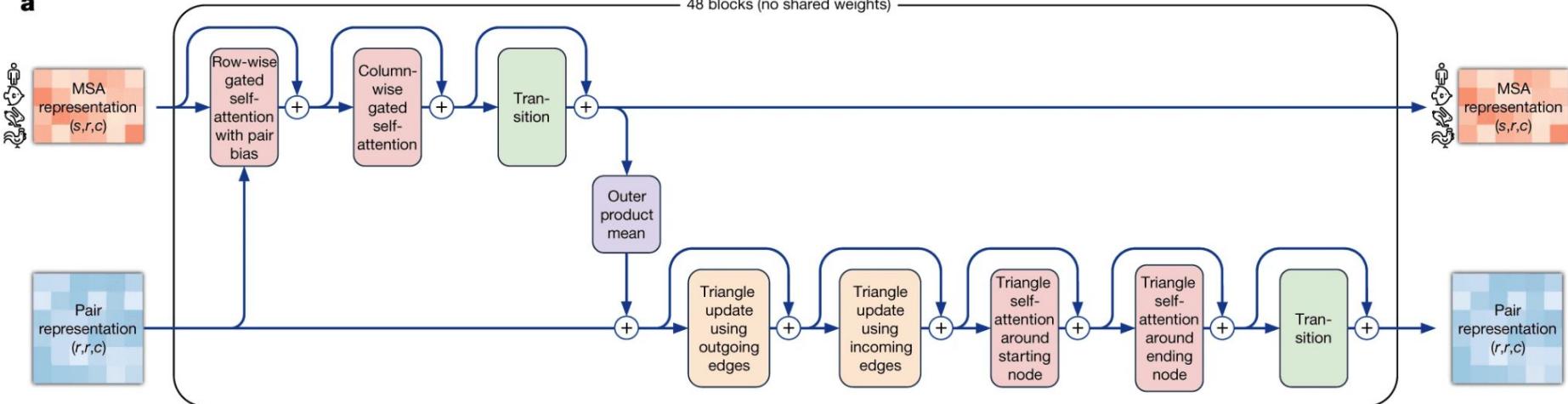
## 1. Input embedding

e

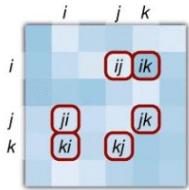


## 4. Recycling

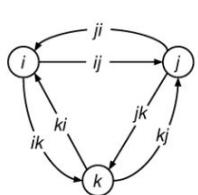


**a****b**

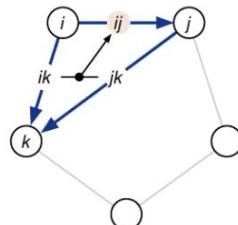
Pair representation  
( $r, r, c$ )



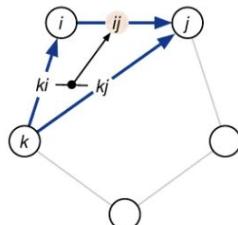
Corresponding edges  
in a graph

**c**

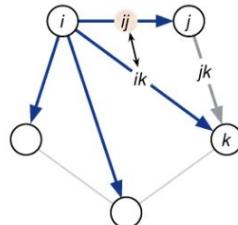
Triangle multiplicative update  
using 'outgoing' edges



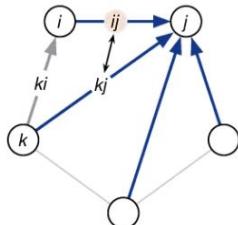
Triangle multiplicative update  
using 'incoming' edges



Triangle self-attention around  
starting node



Triangle self-attention around  
ending node



---

---

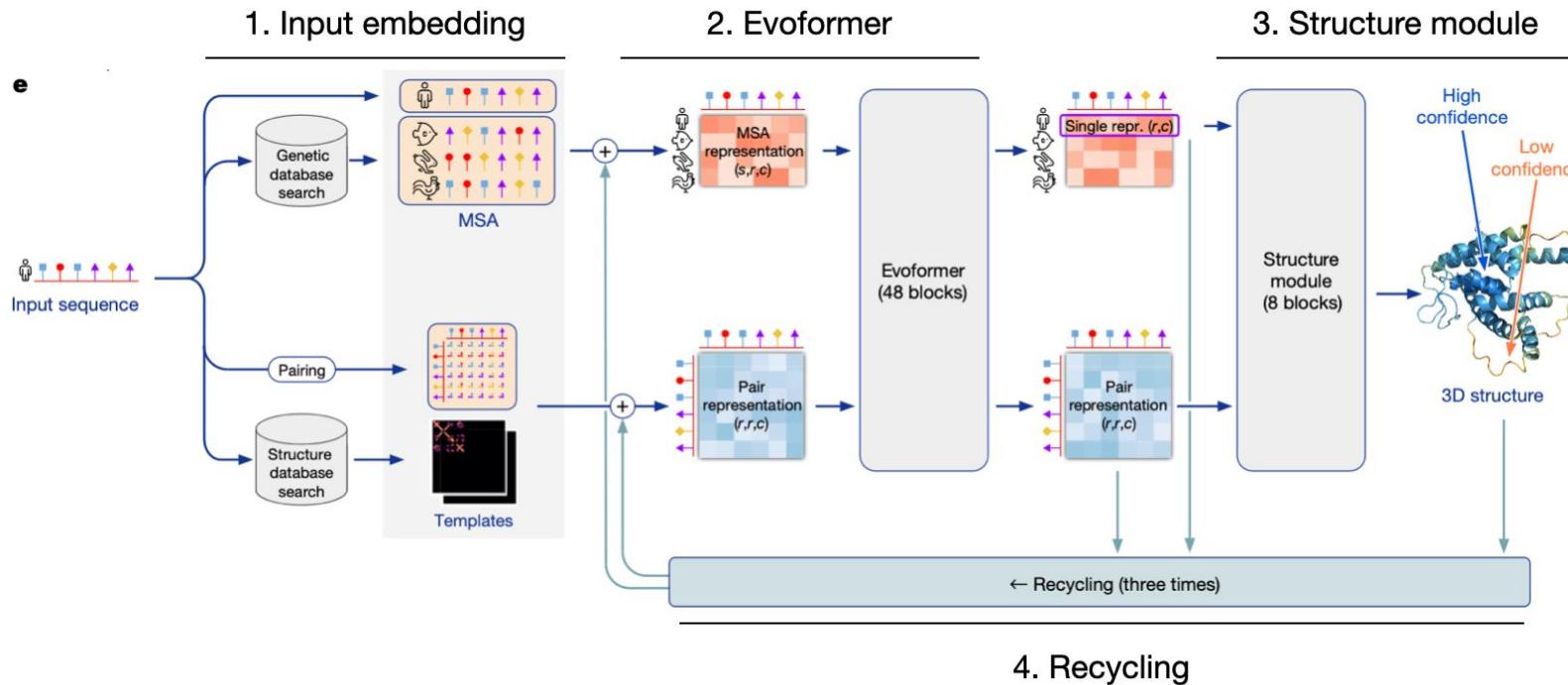
# Highly accurate protein structure prediction with AlphaFold (Part 2)

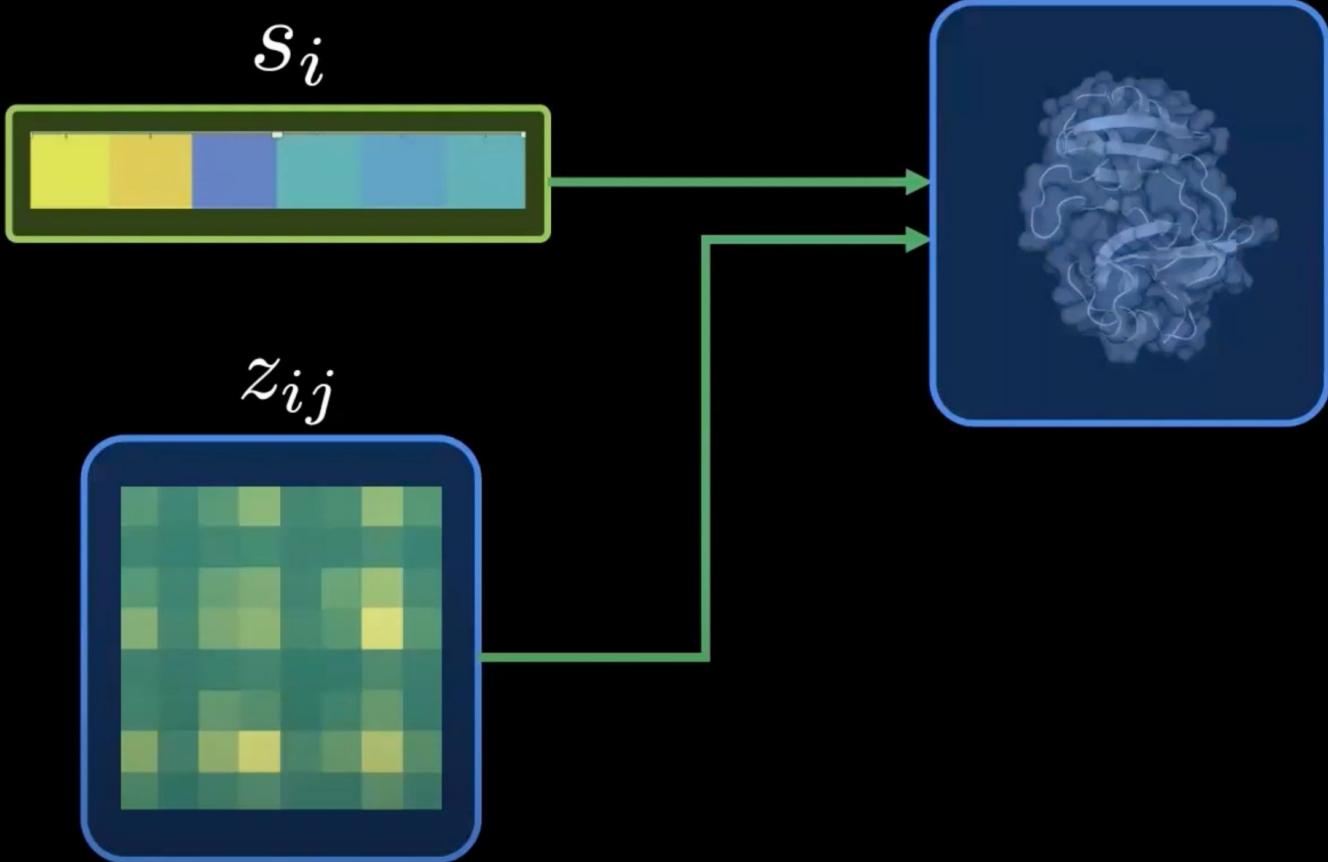
Jumper et al.

---

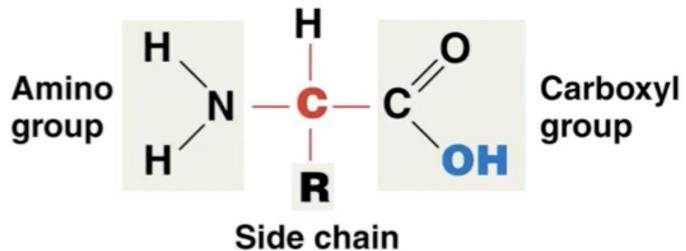
---

# STRUCTURE MODULE



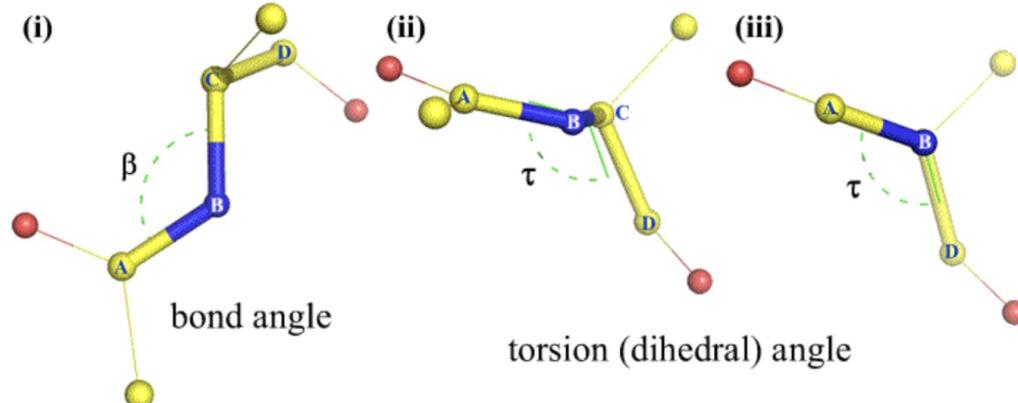


# Protein Backbone Structure

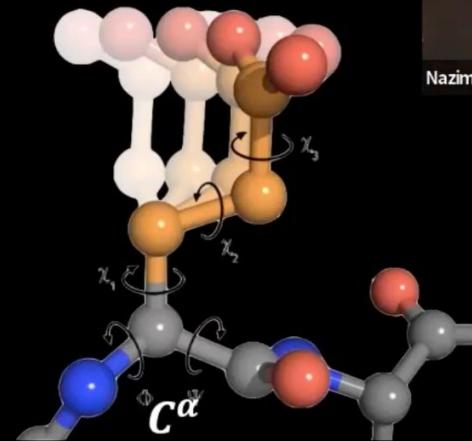
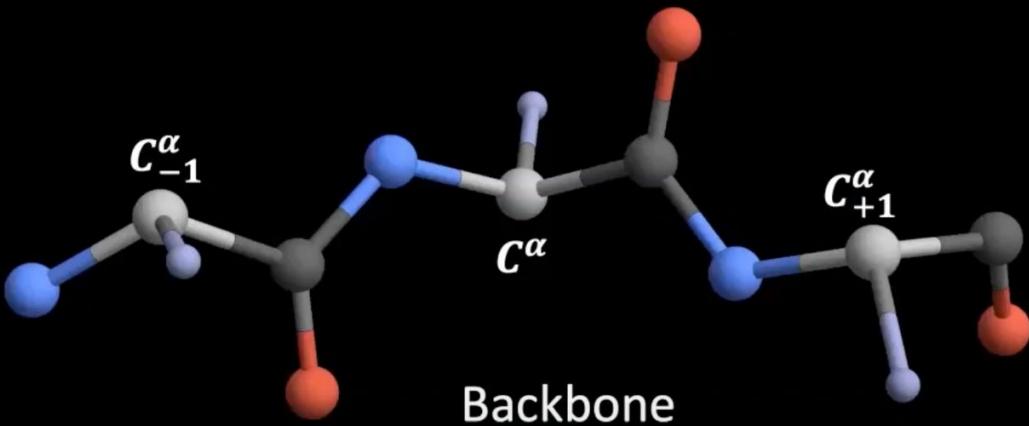


## Rigid Body Assumption

Each atom position in an amino acid residue can be determined by torsion angles ( $\omega$ ,  $\phi$ ,  $\psi$ ,  $\chi$ ) since the bond length and bond angle is almost always same.

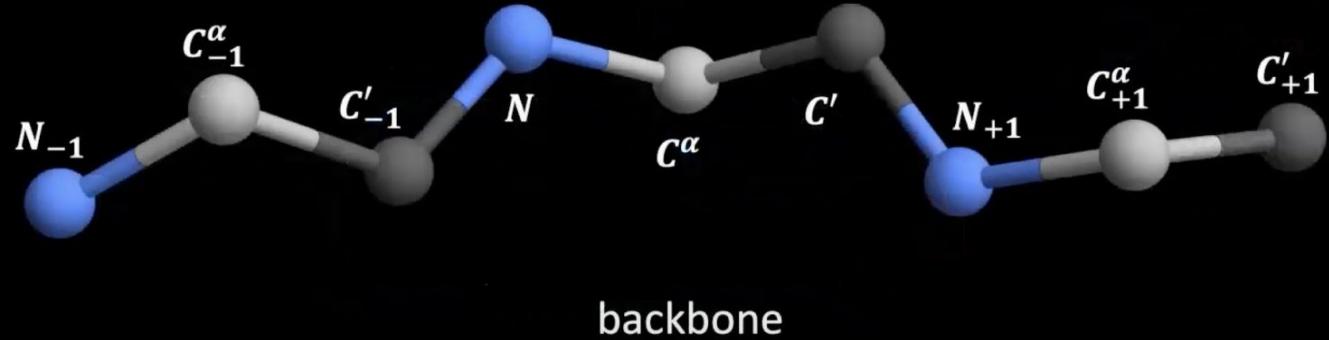


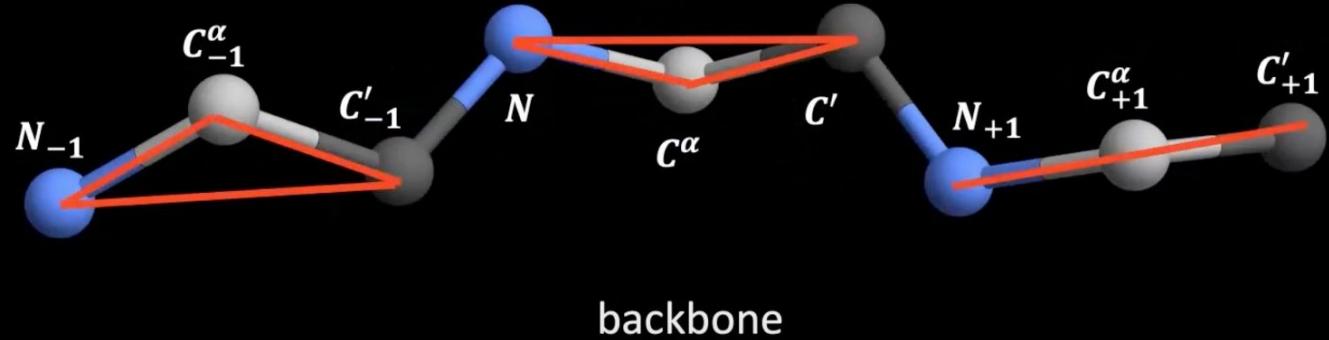
# Protein Backbone Structure

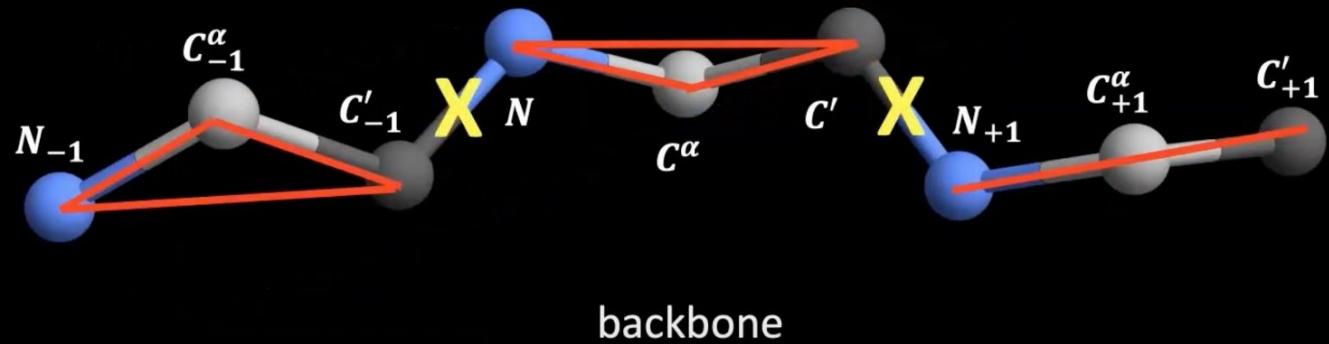


Sidechain  
conformations

Nazim Bouatta

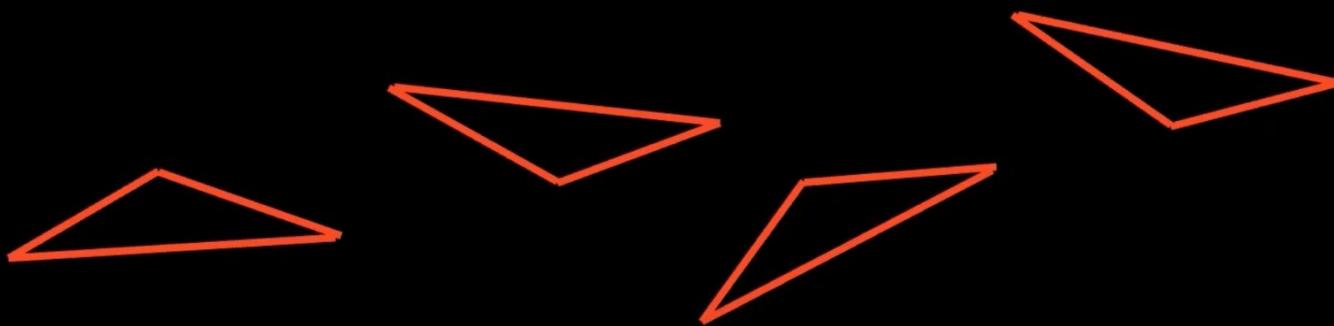


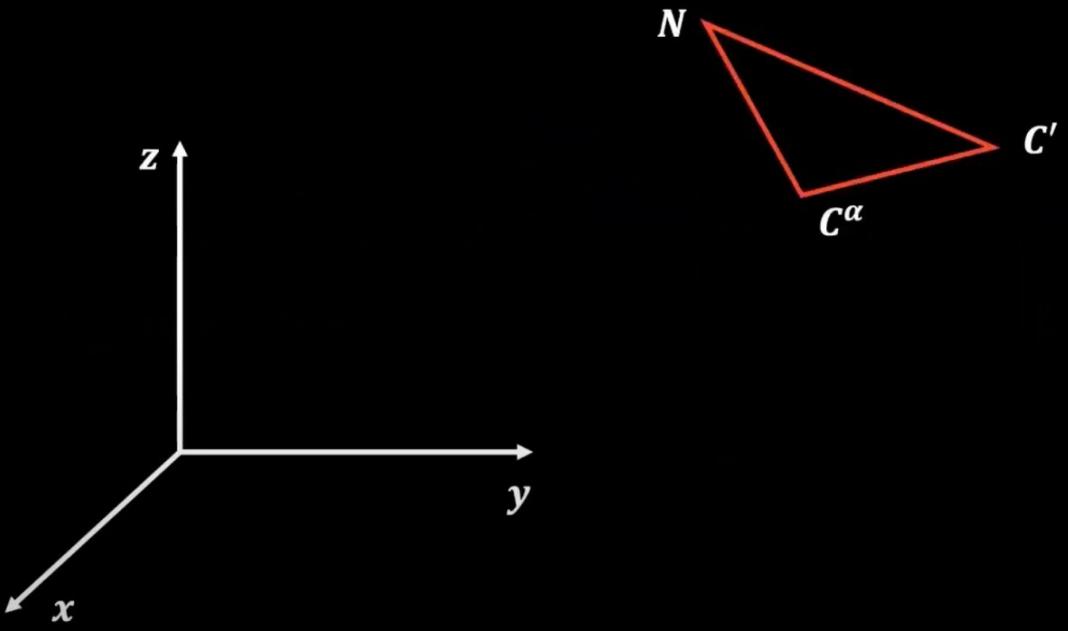


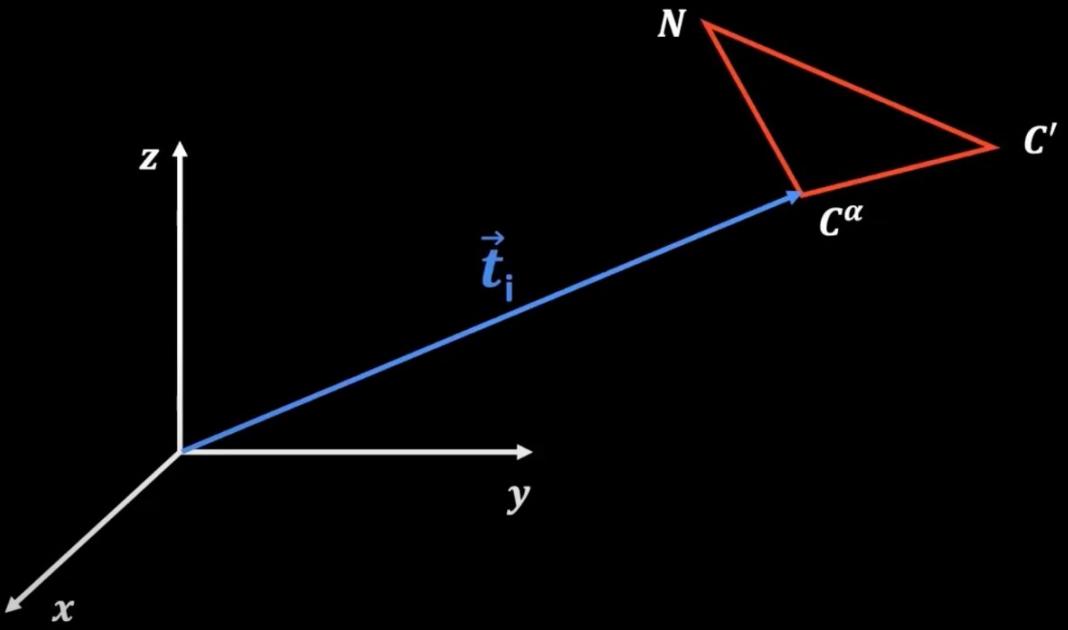


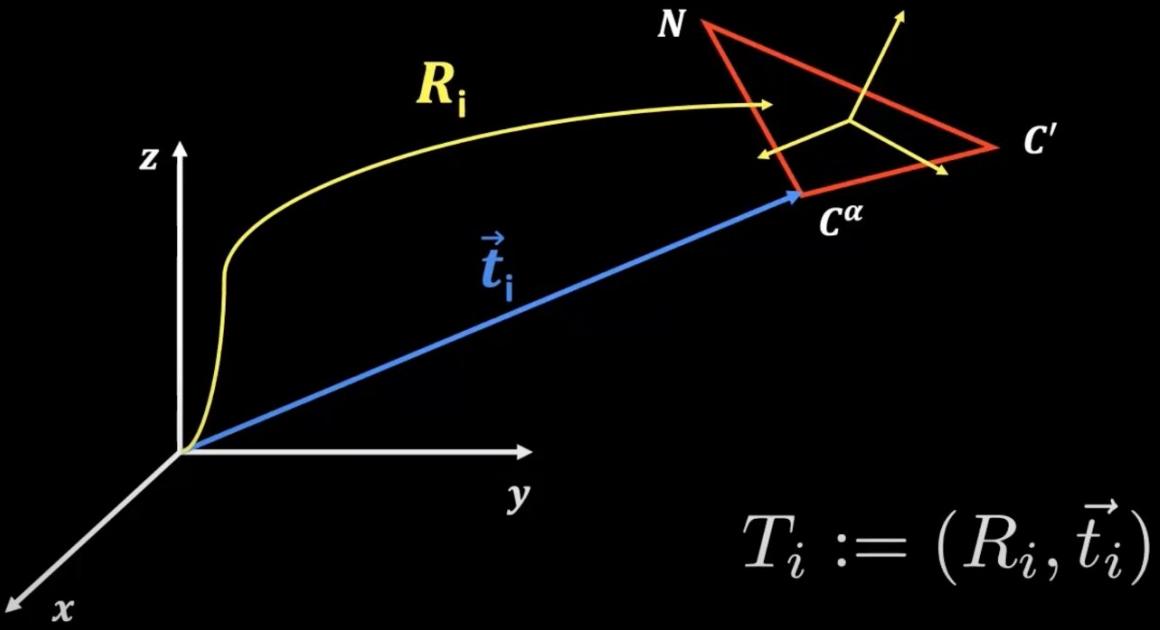


# Residue gas: a bag of triangles



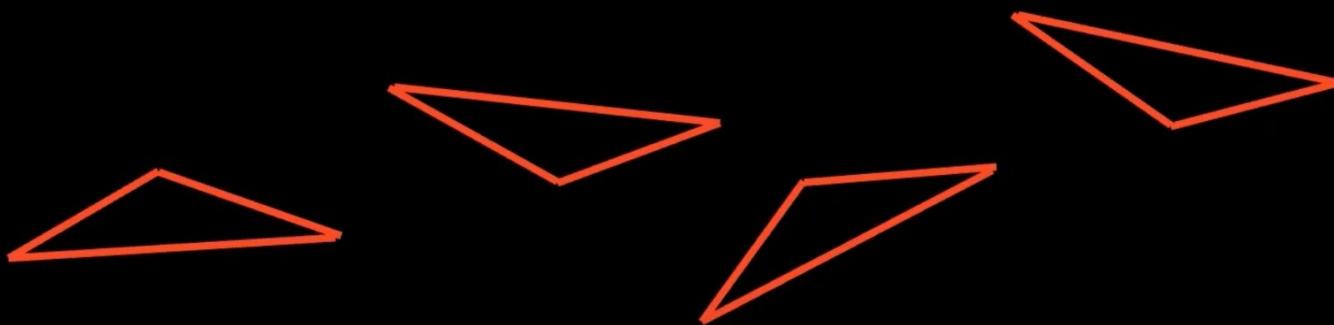




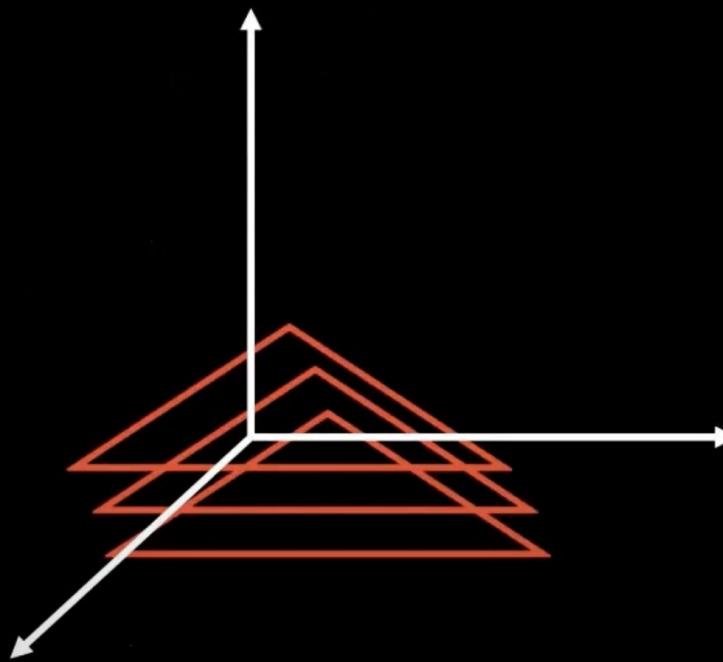




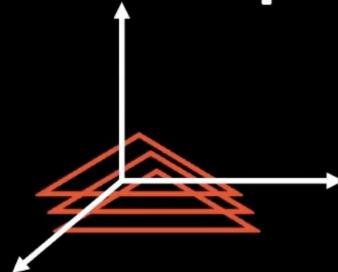
# Residue gas: a bag of triangles



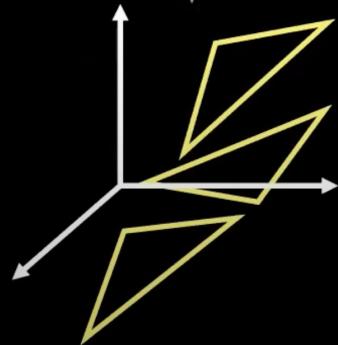
# Initial conditions: “black hole”



# Coordinate update



$\phi$



Each residue in a chain has residue frame, a rotation matrix, and a translation vector.

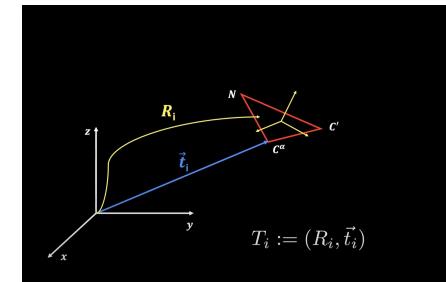
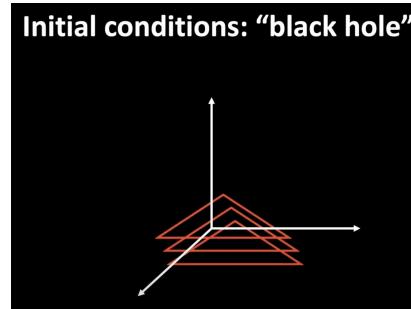
- **Residue frame:** triangle of N, CA, C of each residue (see [¶ Residues and side chains](#) for details)
- The global coordinates of the residue can be determined as:

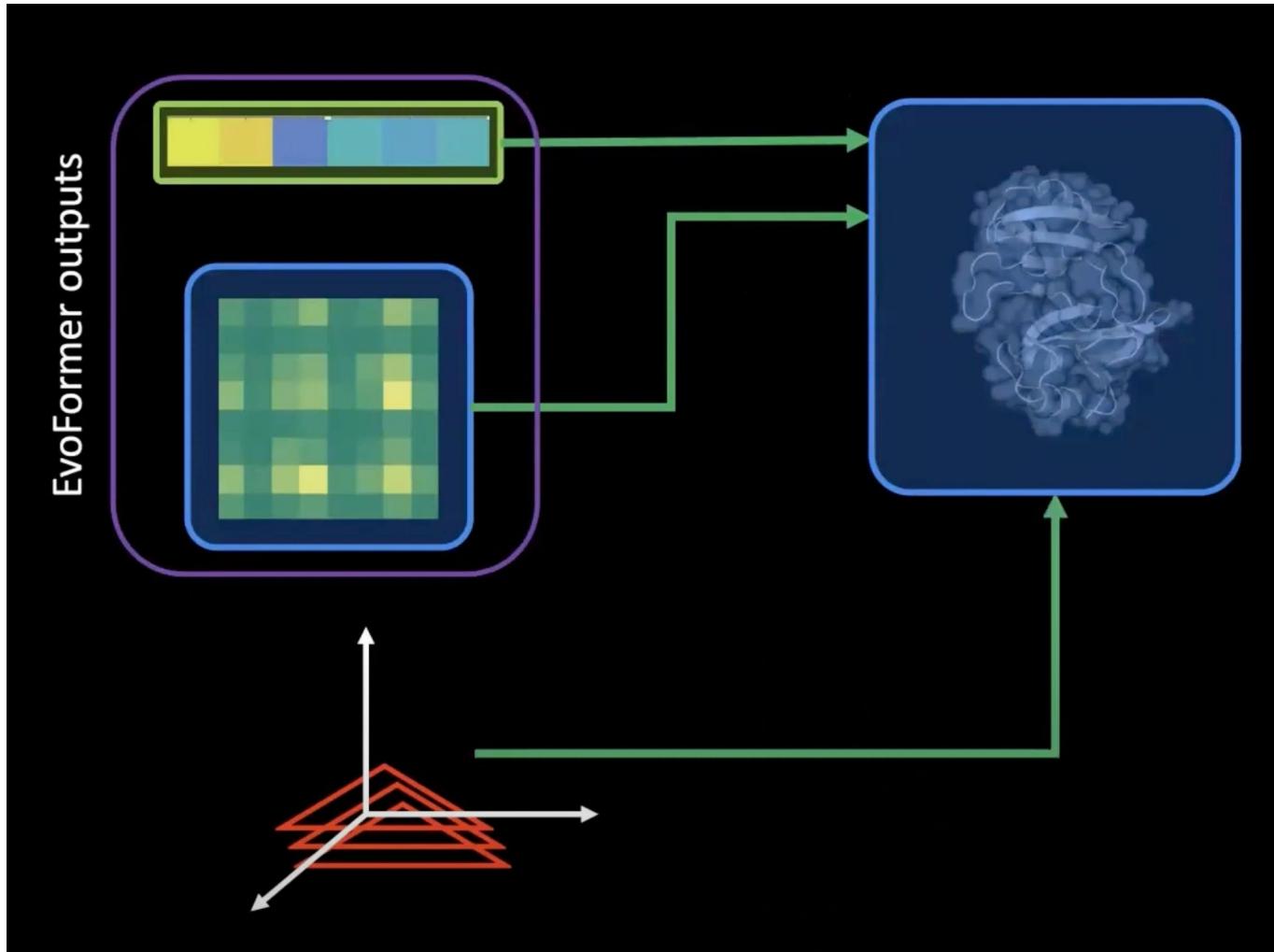
$$\vec{x}_{\text{global}} = T_i \circ \vec{x}_{\text{local}} = R_i \cdot \vec{x}_{\text{local}} + \vec{t}_i$$

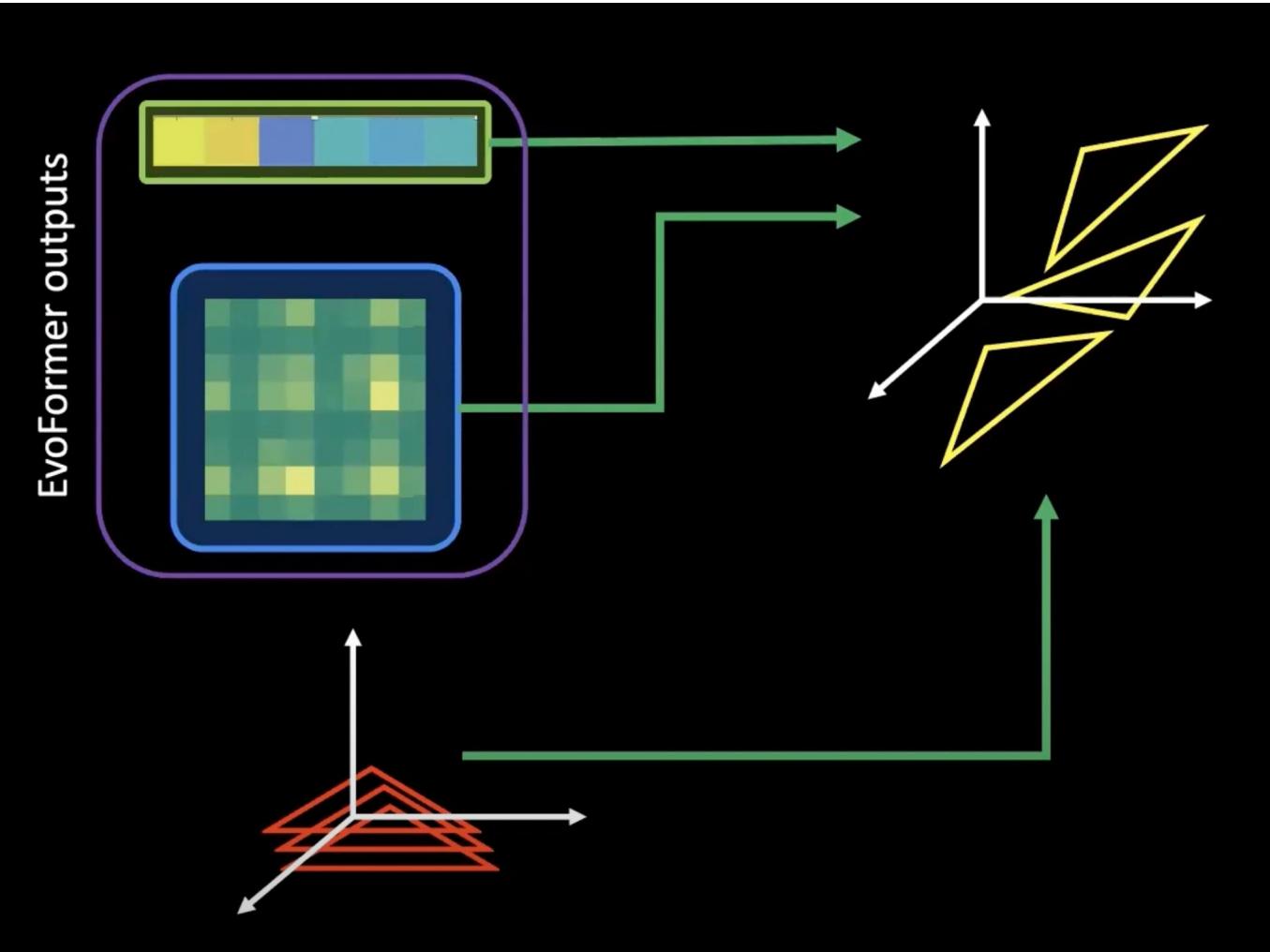
, where  $T_i = (R_i, \vec{t}_i)$

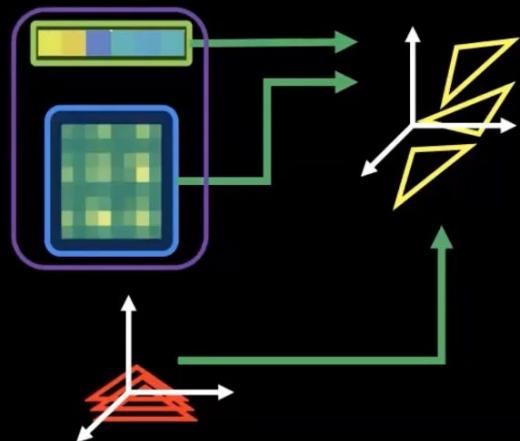
So each residue position  $\vec{x}_i, \text{local}$  is centered at origin, and can be mapped to its position by  $T_i \circ \vec{x}_i, \text{local}$

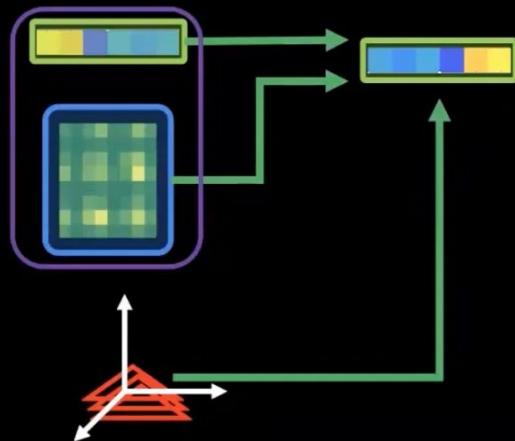
- Orientation:  $R_i \in \mathbb{R}^{3 \times 3}$  i.e. a **rotation matrix**
- Global position:  $\vec{t}_i \in \mathbb{R}^3$  i.e. a **translation vector**
- Black-hole initialization: initially, all residue is at origin  $\vec{t}_i = [0, 0, 0]$  with same identity orientation  $R_i = I_3$

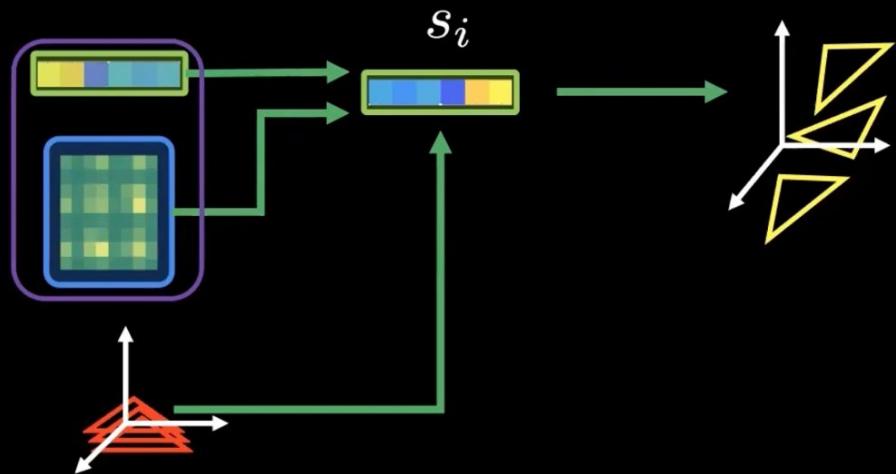


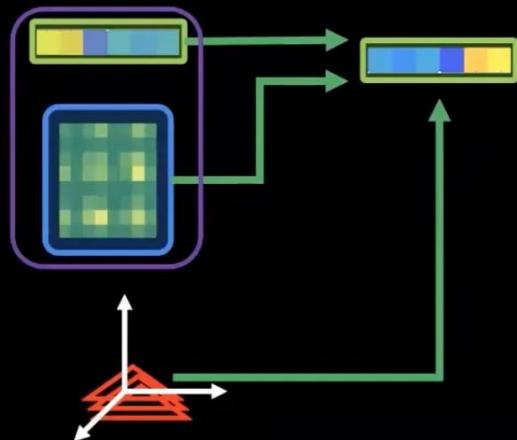






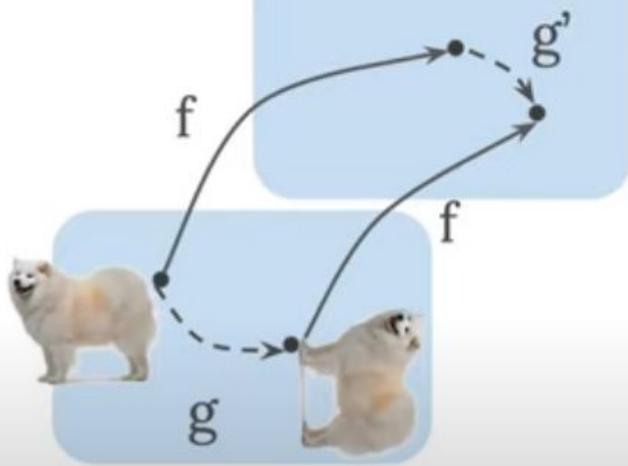






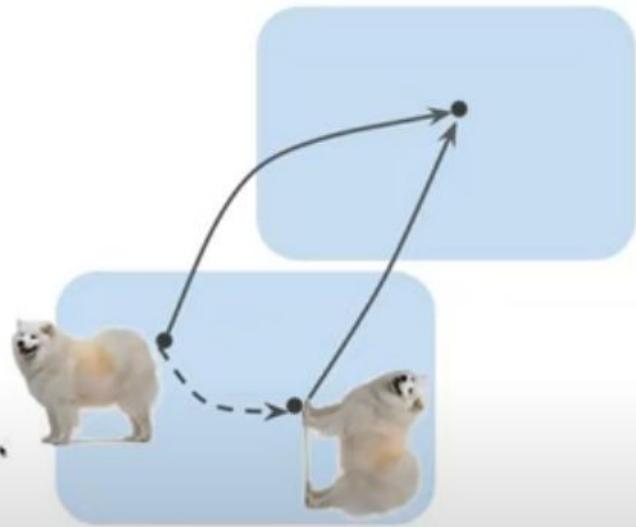
## Equivariance

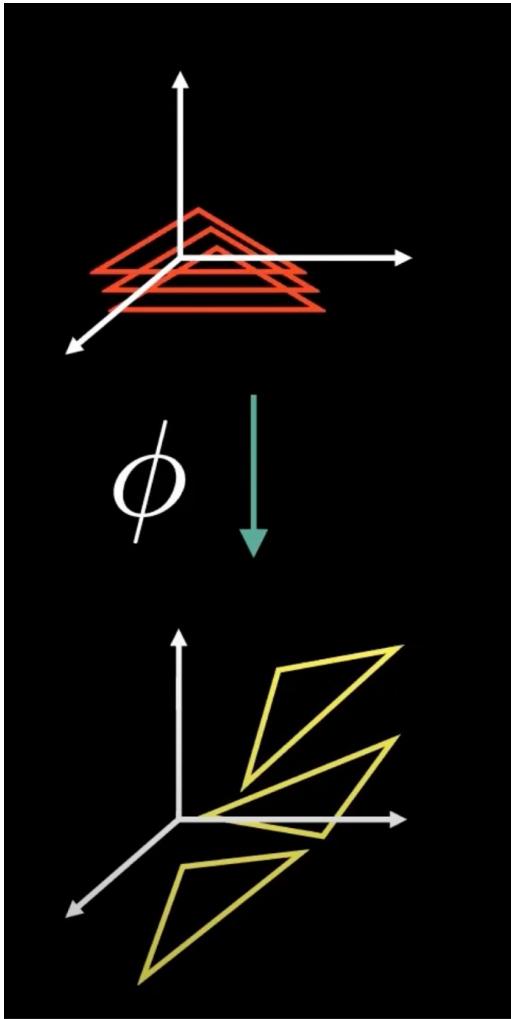
$$f(g(x)) = g' f(x)$$

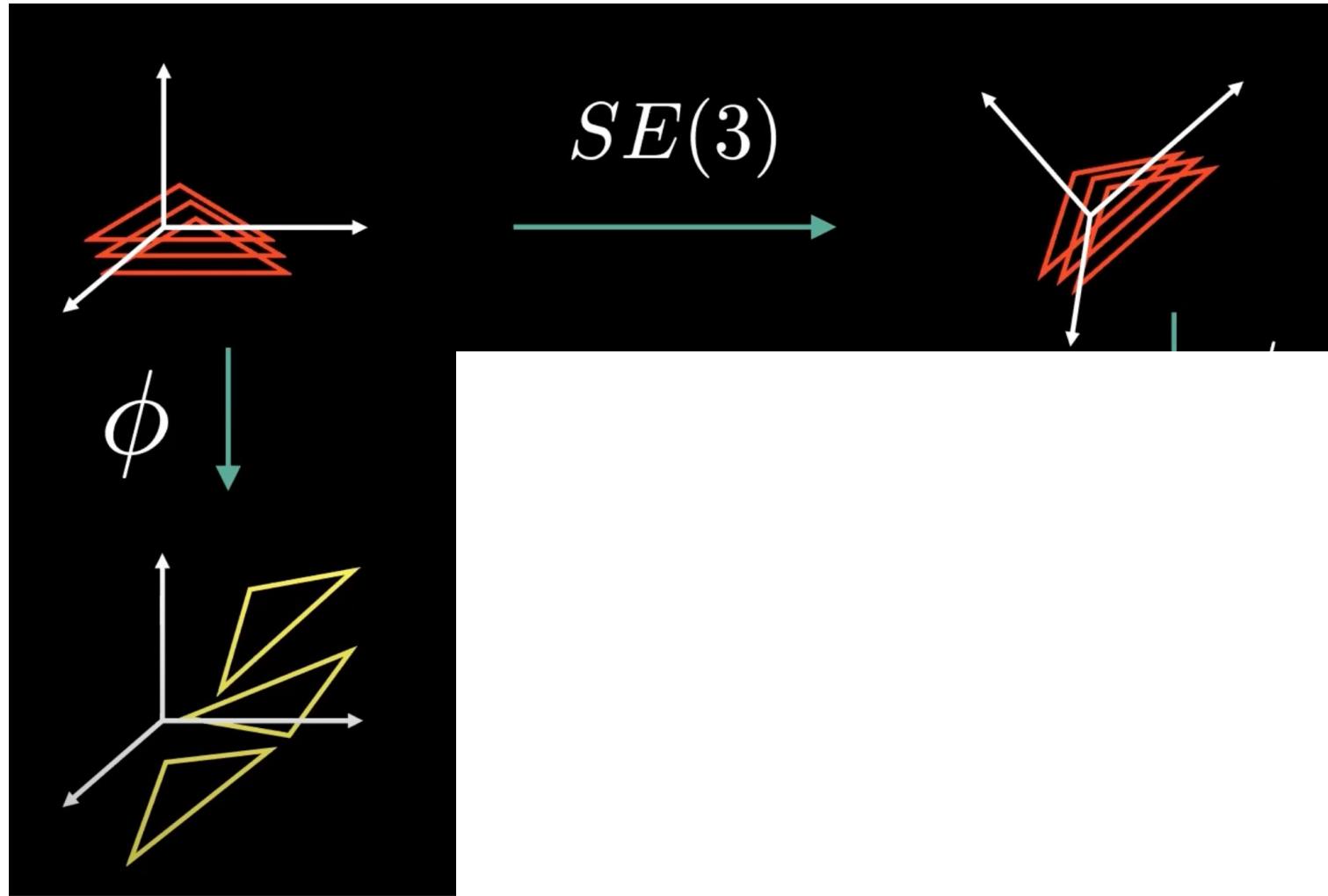


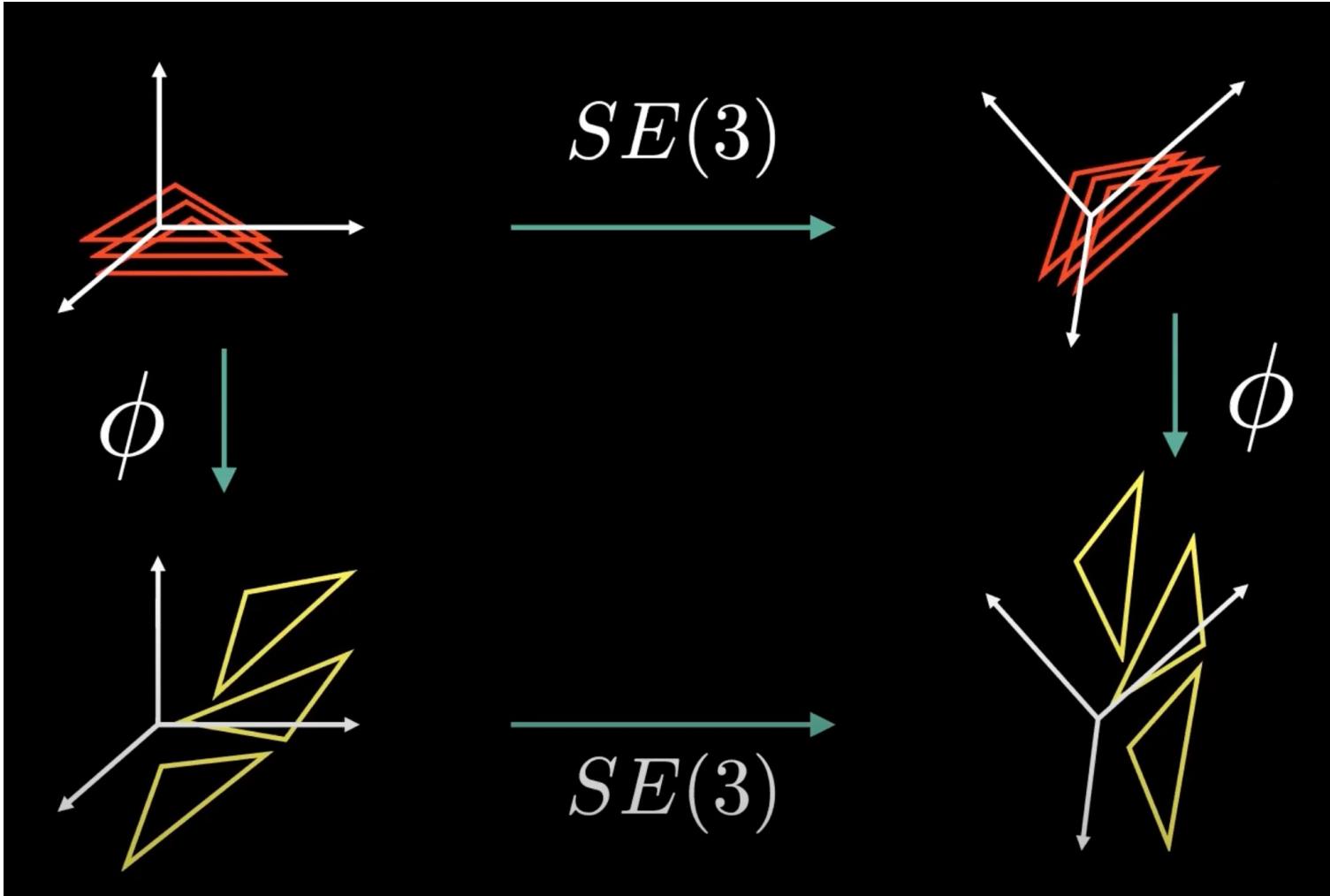
## Invariance

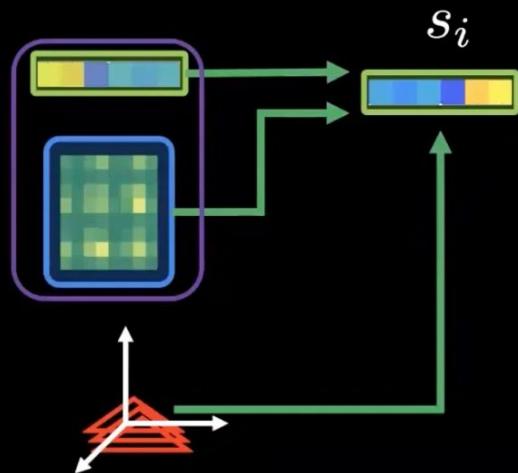
$$f(g(x)) = f(x)$$

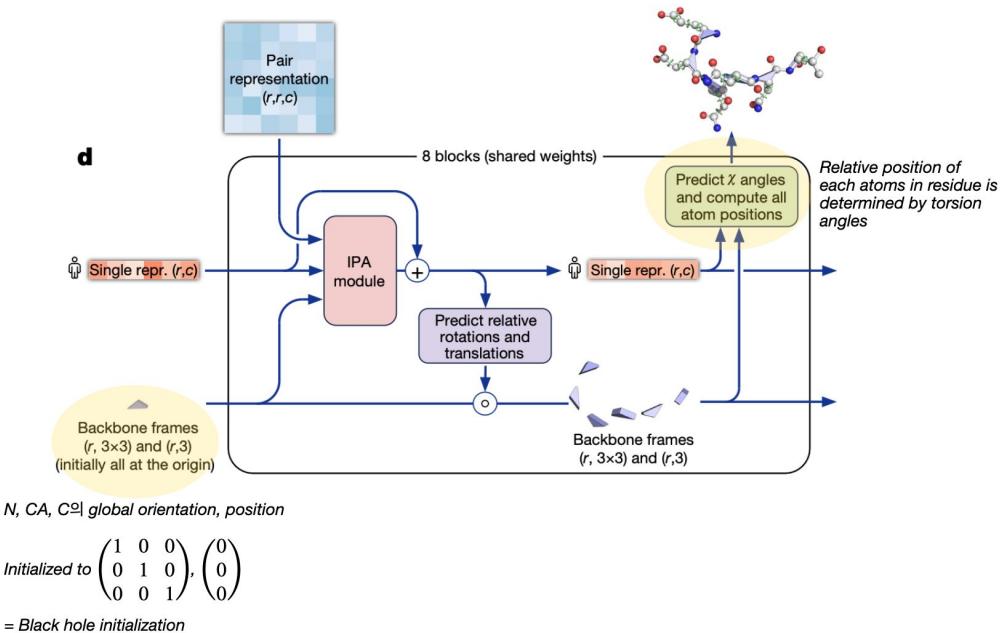
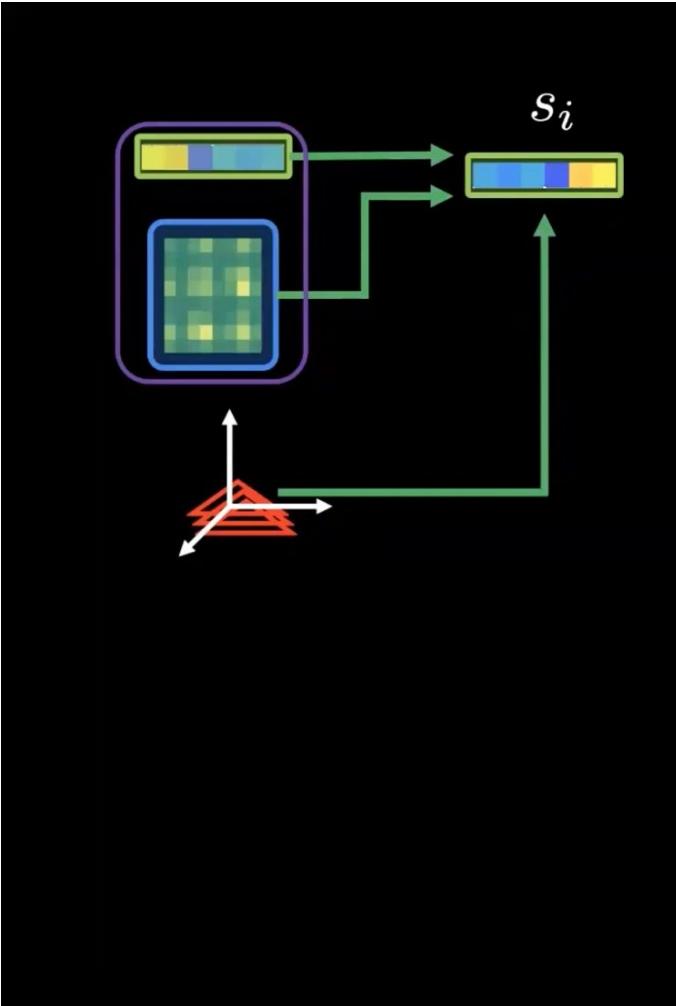


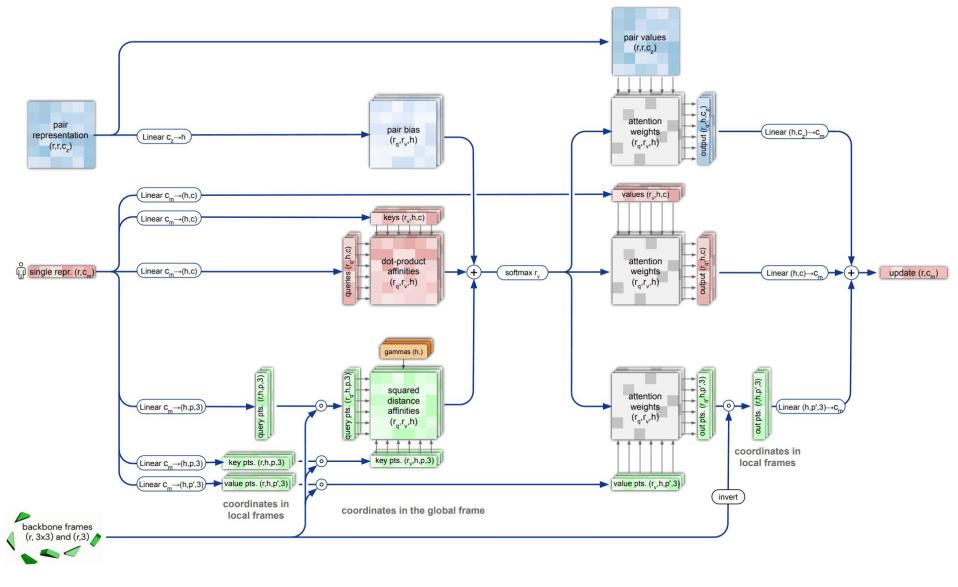
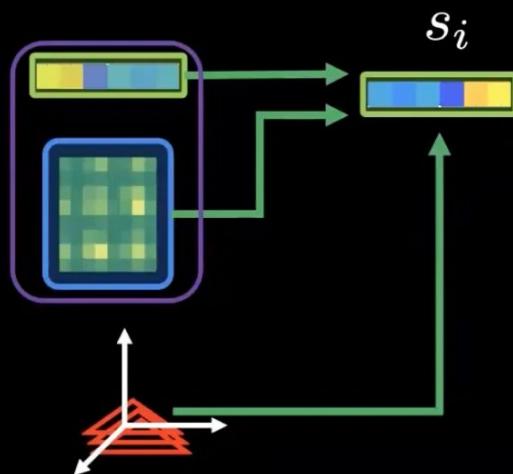




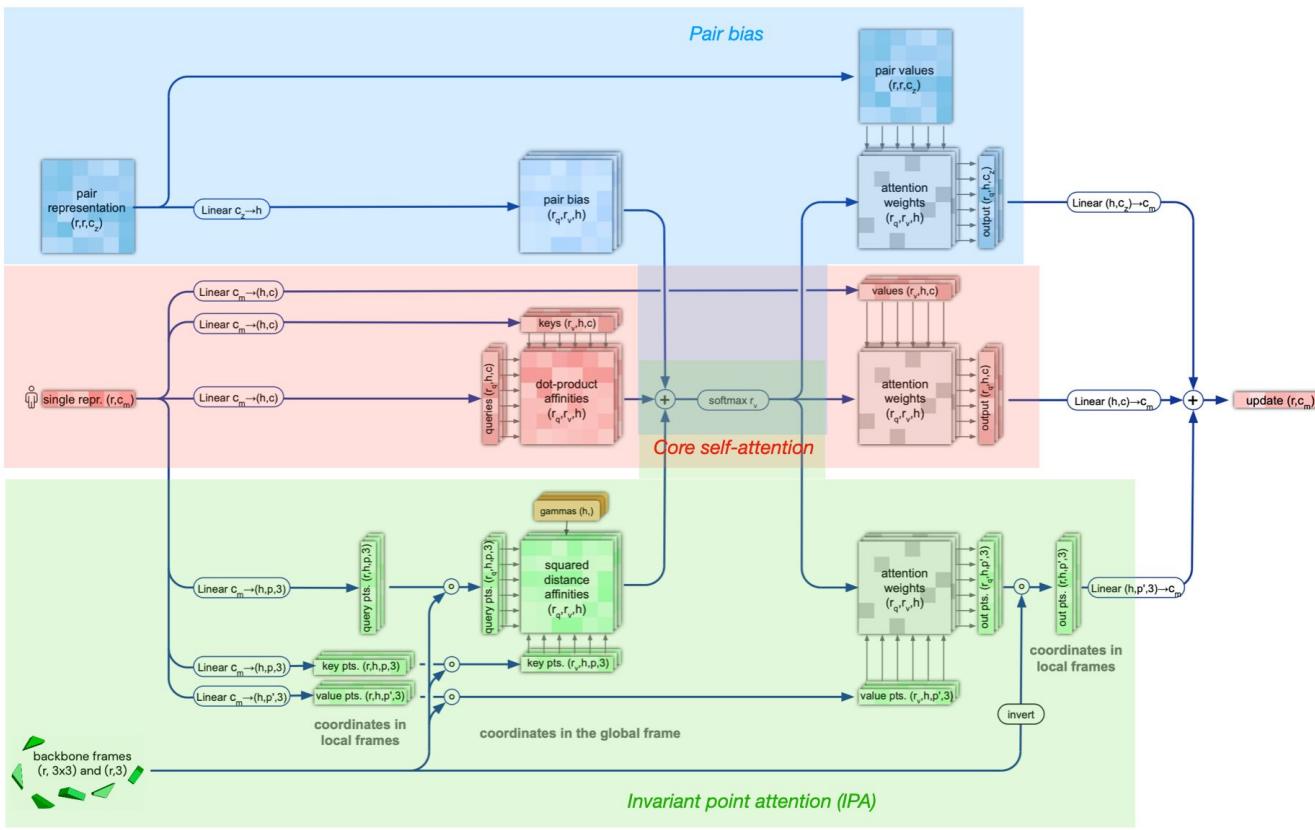








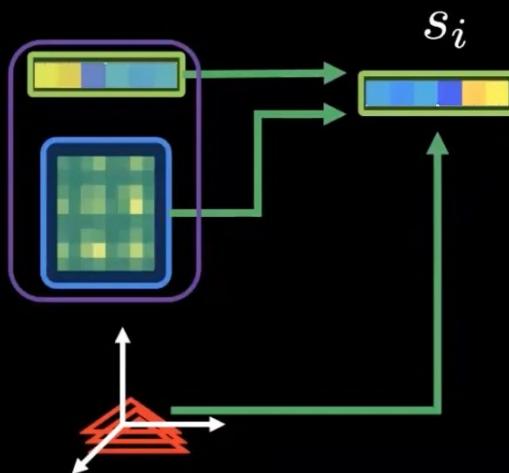
**Supplementary Figure 8 | Invariant Point Attention Module.** (top, blue arrays) modulation by the pair representation. (middle, red arrays) standard attention on abstract features. (bottom, green arrays) Invariant point attention. Dimensions:  $r$ : residues,  $c$ : channels,  $h$ : heads,  $p$ : points.



$$a_{ij}^h = \text{softmax}_k \left( w_L \left( \frac{1}{\sqrt{c}} \mathbf{q}_i^h \top \mathbf{k}_j^h + b_{ij}^h - \frac{\gamma^h w c}{2} \sum_p \| T_i \circ \mathbf{q}_i^{hp} - T_j \circ \mathbf{k}_j^{hp} \|^2 \right) \right)$$

Pair representation  
 Core self-attention of single representation  
 Invariant point attention (IPA)

The closer the distance is, the bigger the attention weights are!



**Algorithm 22** Invariant point attention (IPA)

---

```

def InvariantPointAttention( $\{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, \{T_i\}$ ,  $N_{\text{head}} = 12, c = 16, N_{\text{query points}} = 4, N_{\text{point values}} = 8$ ) :
    1:  $\mathbf{q}_i^h, \mathbf{k}_i^h, \mathbf{v}_i^h = \text{LinearNoBias}(\mathbf{s}_i)$   $\mathbf{q}_i^h, \mathbf{k}_i^h, \mathbf{v}_i^h \in \mathbb{R}^c, h \in \{1, \dots, N_{\text{head}}\}$ 
    2:  $\vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$   $\vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp} \in \mathbb{R}^3, p \in \{1, \dots, N_{\text{query points}}\}$ , units: nanometres
    3:  $\vec{\mathbf{v}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$   $\vec{\mathbf{v}}_i^{hp} \in \mathbb{R}^3, p \in \{1, \dots, N_{\text{point values}}\}$ , units: nanometres
    4:  $b_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$ 
    5:  $w_C = \sqrt{\frac{2}{9N_{\text{query points}}}}$ ,
    6:  $w_L = \sqrt{\frac{1}{3}}$ 
    7:  $a_{ij}^h = \text{softmax}_j \left( w_L \left( \frac{1}{\sqrt{c}} \mathbf{q}_i^{h\top} \mathbf{k}_j^h + b_{ij}^h - \frac{\gamma^h w_C}{2} \sum_p \left\| T_i \circ \vec{\mathbf{q}}_i^{hp} - T_j \circ \vec{\mathbf{k}}_j^{hp} \right\|^2 \right) \right)$ 
    8:  $\tilde{\mathbf{o}}_i^h = \sum_j a_{ij}^h \mathbf{z}_{ij}$ 
    9:  $\mathbf{o}_i^h = \sum_j a_{ij}^h \mathbf{v}_j^h$ 
    10:  $\vec{\mathbf{o}}_i^{hp} = T_i^{-1} \circ \sum_j a_{ij}^h (T_j \circ \vec{\mathbf{v}}_j^{hp})$ 
    11:  $\tilde{\mathbf{s}}_i = \text{Linear} \left( \text{concat}_{h,p}(\tilde{\mathbf{o}}_i^h, \mathbf{o}_i^h, \vec{\mathbf{o}}_i^{hp}, \|\vec{\mathbf{o}}_i^{hp}\|) \right)$ 
    12: return  $\{\tilde{\mathbf{s}}_i\}$ 

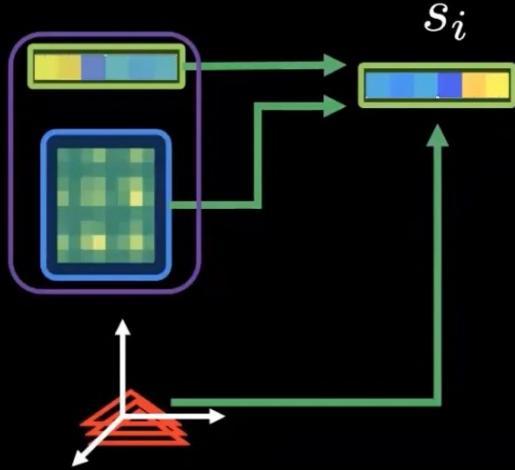
```

---

$$a_{ij} = \text{softmax} \left( q_i^T k_j + b_{ij} + \|T_i \circ \vec{q}_i - T_j \circ \vec{k}_j\|^2 \right)$$

$$T_i := (R_i, \vec{t}_i)$$

The proof for invariance is straight-forward: The global transformation cancels out in the affinity computation ([Algorithm 22 line 7](#)), because the L2-norm of a vector is invariant under rigid transformations:



$$\left\| \left( T_{\text{global}} \circ T_i \right) \circ \vec{q}_i^{hp} - \left( T_{\text{global}} \circ T_j \right) \circ \vec{k}_j^{hp} \right\|^2 = \left\| T_{\text{global}} \circ \left( T_i \circ \vec{q}_i^{hp} - T_j \circ \vec{k}_j^{hp} \right) \right\|^2 \quad (3)$$

$$= \left\| T_i \circ \vec{q}_i^{hp} - T_j \circ \vec{k}_j^{hp} \right\|^2. \quad (4)$$

In the computation of the output points ([Algorithm 22 line 10](#)) it cancels out when mapping back to the local frame:

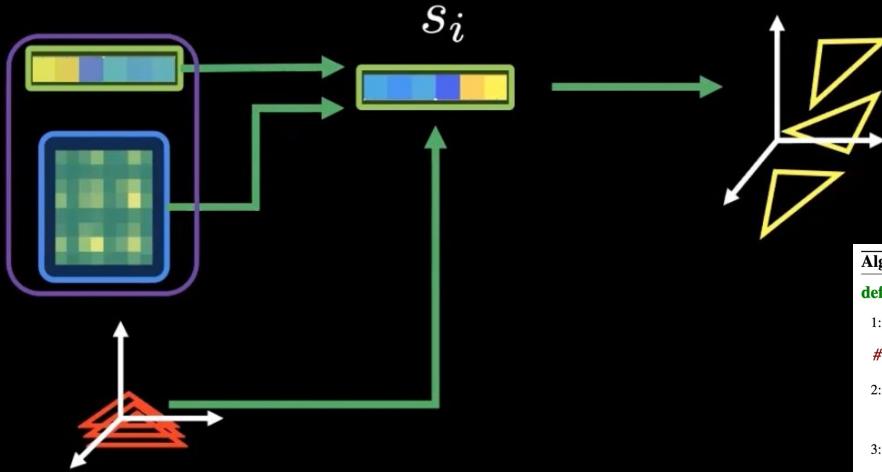
$$\left( T_{\text{global}} \circ T_i \right)^{-1} \circ \sum_j a_{ij}^h \left( \left( T_{\text{global}} \circ T_j \right) \circ \vec{v}_j^{hp} \right) = T_i^{-1} \circ T_{\text{global}}^{-1} \circ T_{\text{global}} \circ \sum_j a_{ij}^h \left( T_j \circ \vec{v}_j^{hp} \right) \quad (5)$$

$$= T_i^{-1} \circ \sum_j a_{ij}^h \left( T_j \circ \vec{v}_j^{hp} \right) \quad (6)$$

The invariance with respect to the global reference frame in turn implies that applying a shared rigid motion to all residues, while keeping the embeddings fixed, will lead to the same update in the local frames. Therefore, the updated structure will be transformed by the same shared rigid motion showing that this update rule is equivariant under rigid motions. Here and elsewhere, “rigid motion” includes proper rotation and translation but not reflection.

$$a_{ij} = \text{softmax} \left( q_i^T k_j + b_{ij} + \| T_i \circ \vec{q}_i - T_j \circ \vec{k}_j \|^2 \right)$$

$$T_i := (R_i, \vec{t}_i)$$




---

**Algorithm 23** Backbone update

```

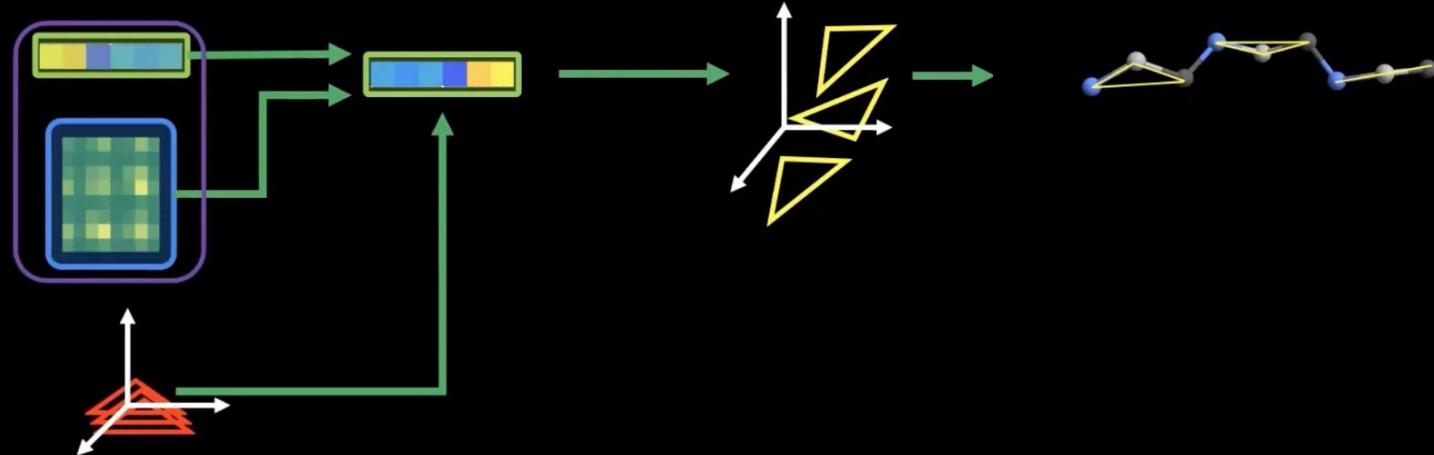
def BackboneUpdate( $s_i$ ) :
    1:  $b_i, c_i, d_i, \vec{t}_i = \text{Linear}(s_i)$ 
    # Convert (non-unit) quaternion to rotation matrix.
    2:  $(a_i, b_i, c_i, d_i) \leftarrow (1, b_i, c_i, d_i) / \sqrt{1 + b_i^2 + c_i^2 + d_i^2}$ 
    3:  $R_i = \begin{pmatrix} a_i^2 + b_i^2 - c_i^2 - d_i^2 & 2b_i c_i - 2a_i d_i & 2b_i d_i + 2a_i c_i \\ 2b_i c_i + 2a_i d_i & a_i^2 - b_i^2 + c_i^2 - d_i^2 & 2c_i d_i - 2a_i b_i \\ 2b_i d_i - 2a_i c_i & 2c_i d_i + 2a_i b_i & a_i^2 - b_i^2 - c_i^2 + d_i^2 \end{pmatrix}$ 
    4:  $T_i = (R_i, \vec{t}_i)$ 
    5: return  $T_i$ 

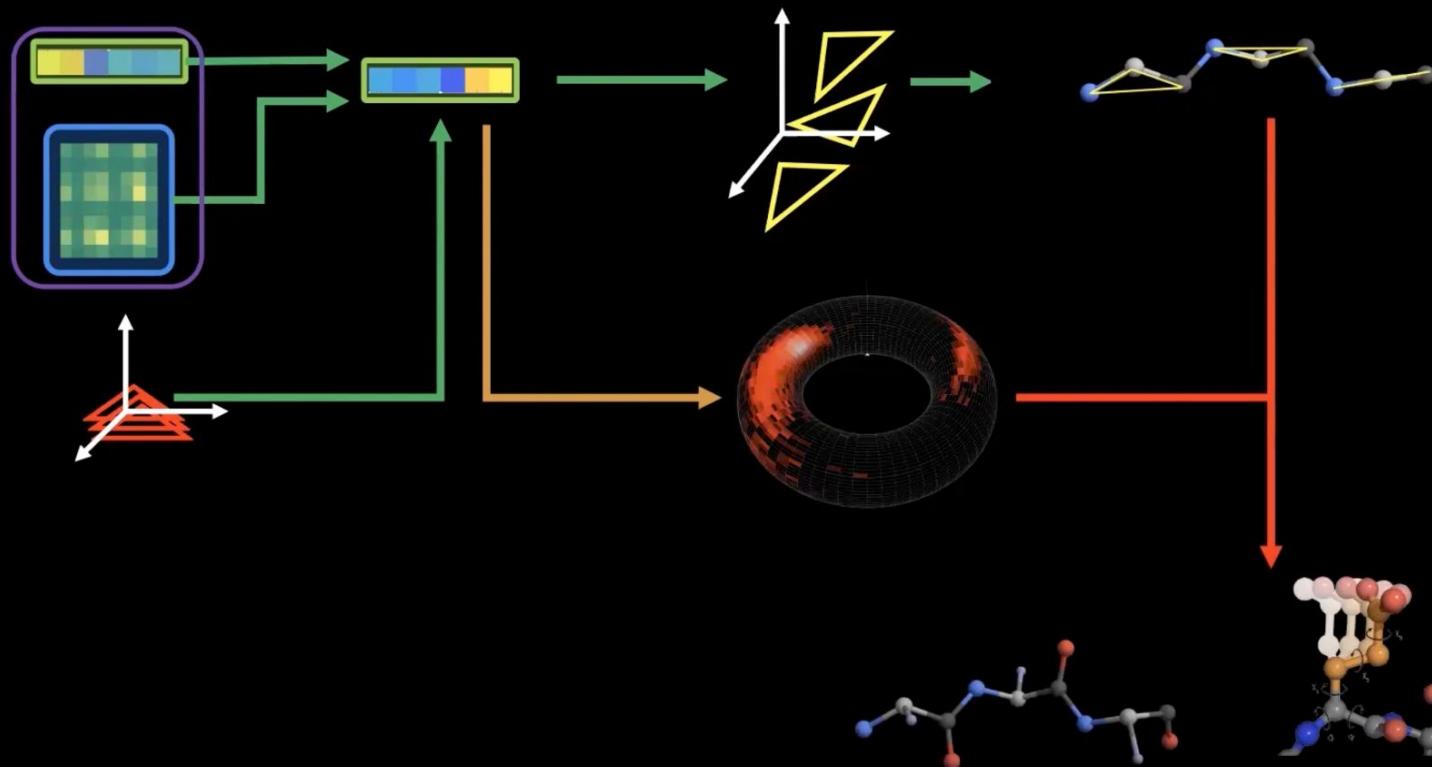
```

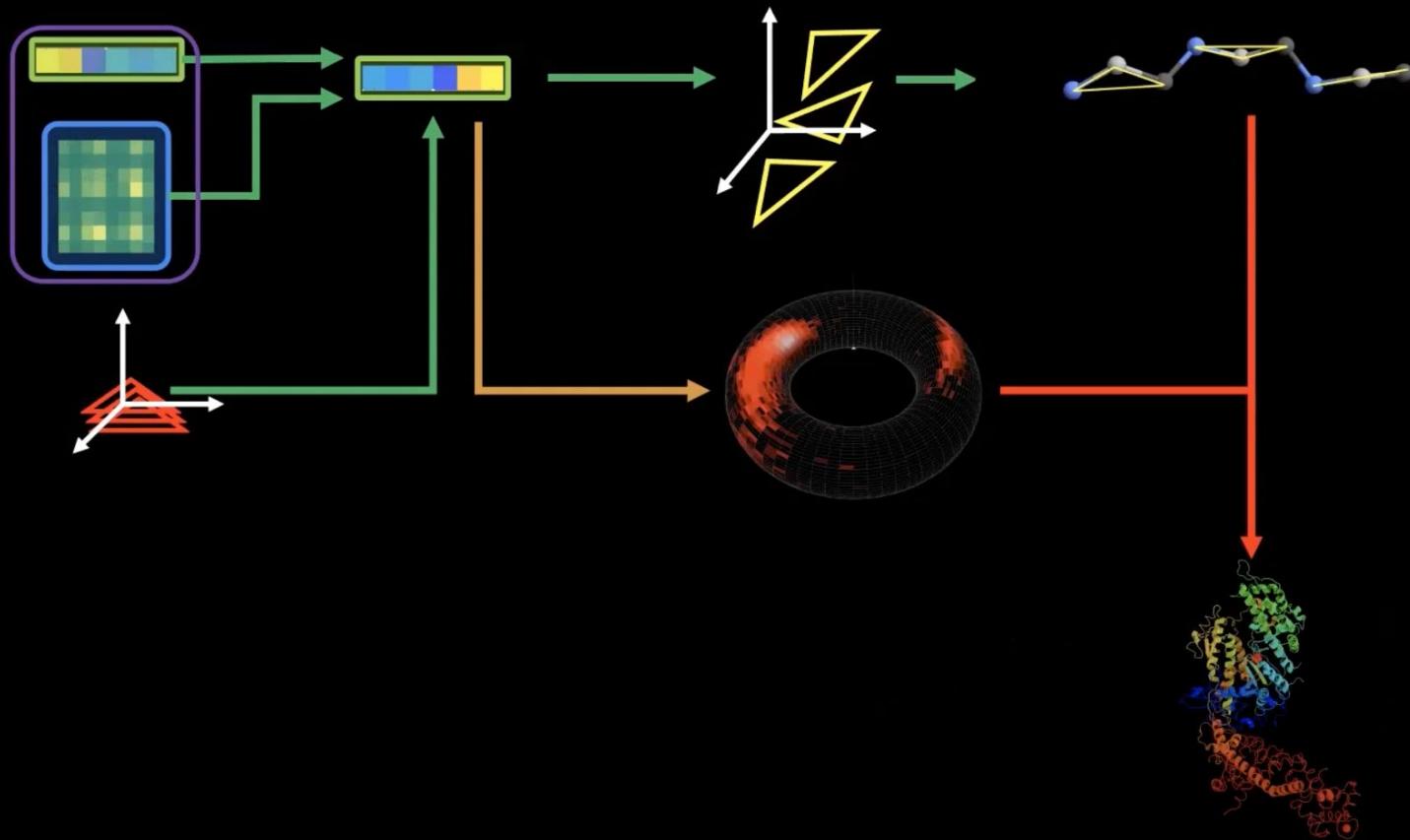
---

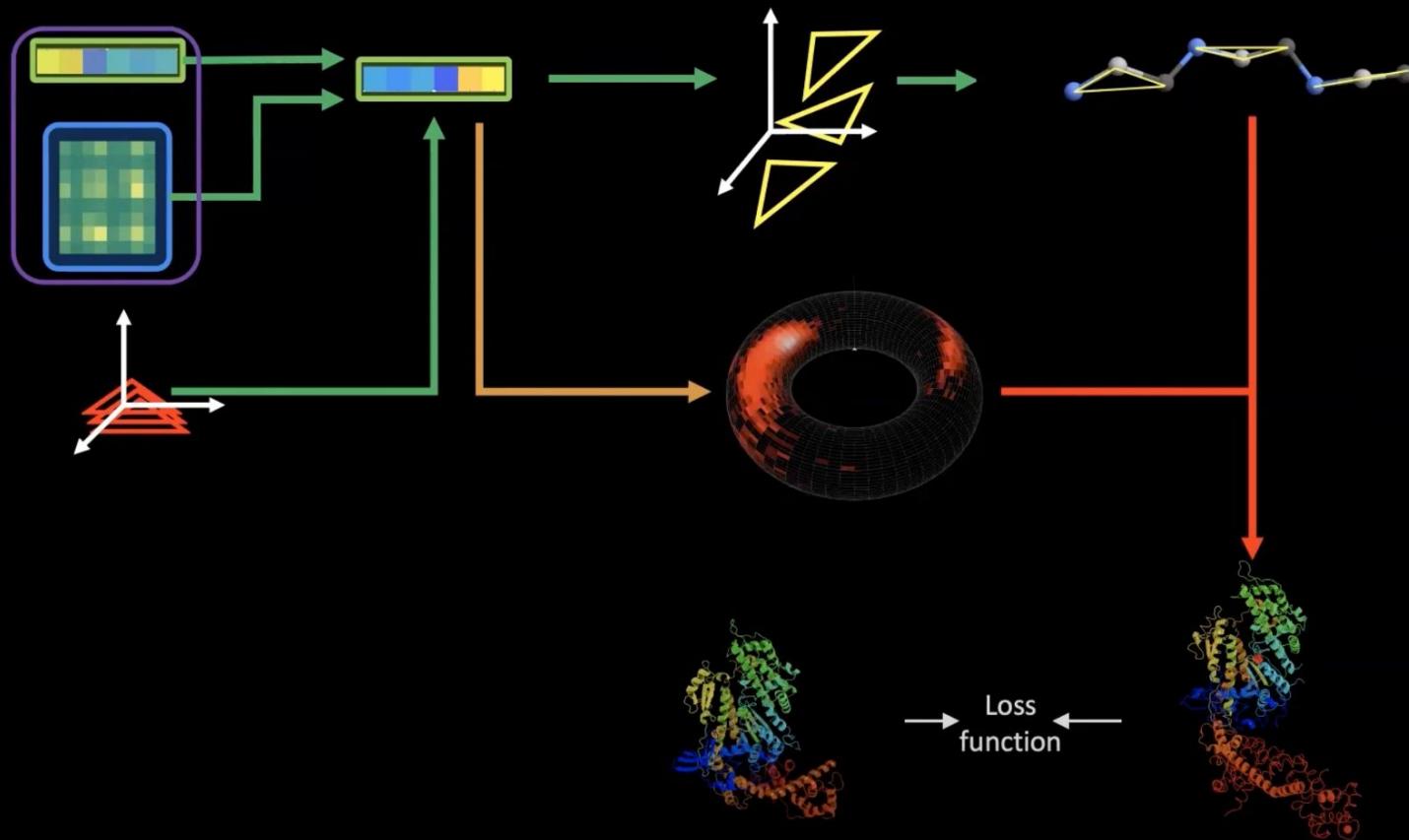
$$\text{BackboneUpdate}(s_i) := (R_i, \vec{t}_i)$$

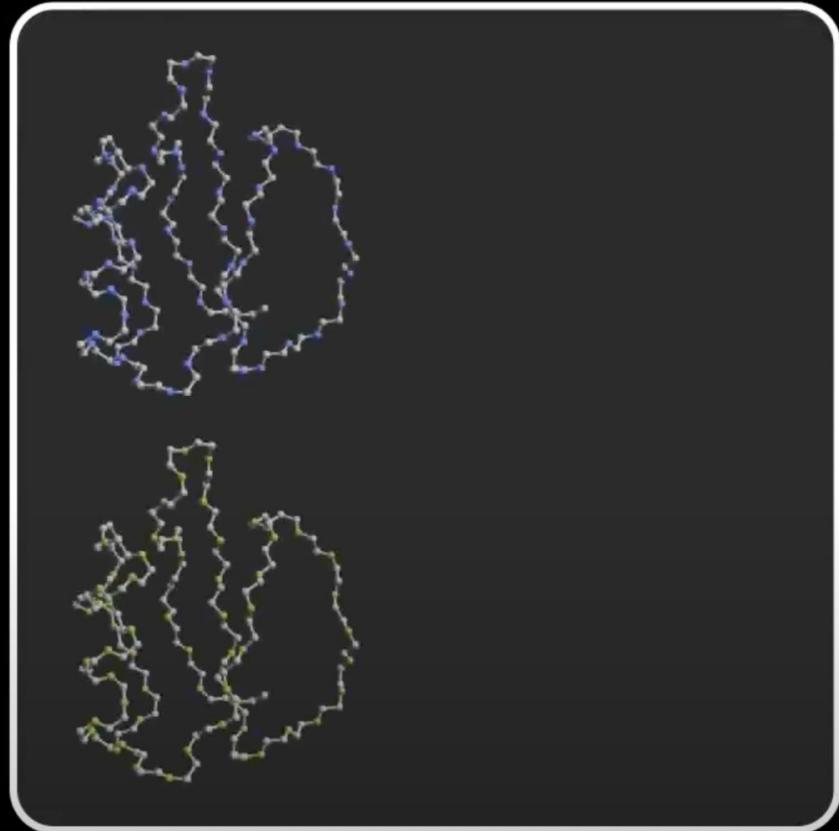
$$T_i \leftarrow T_i \circ \text{BackboneUpdate}(s_i)$$

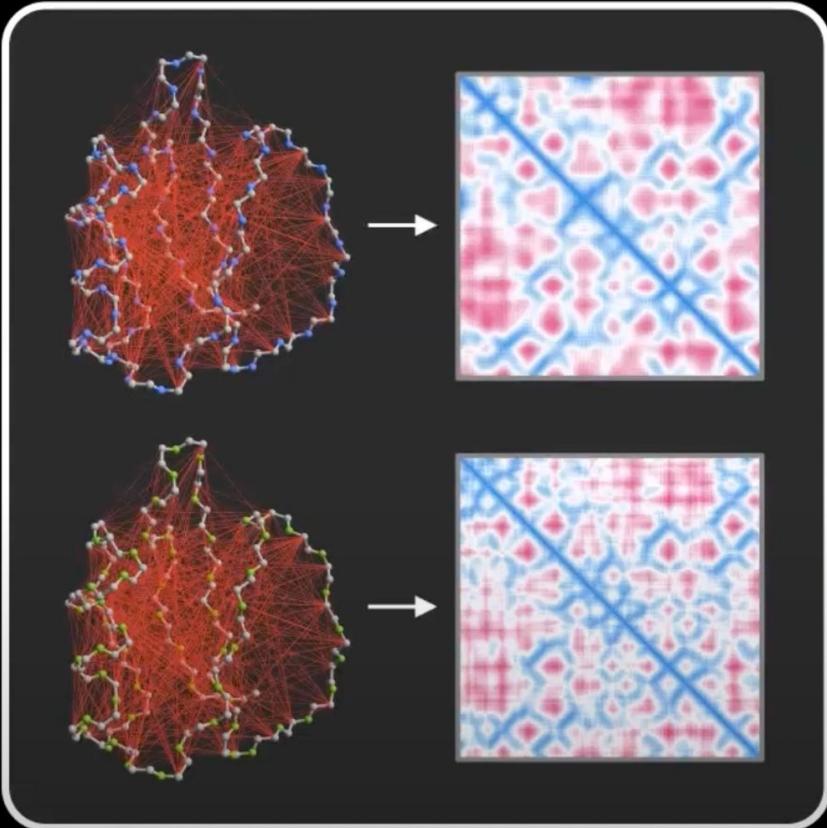




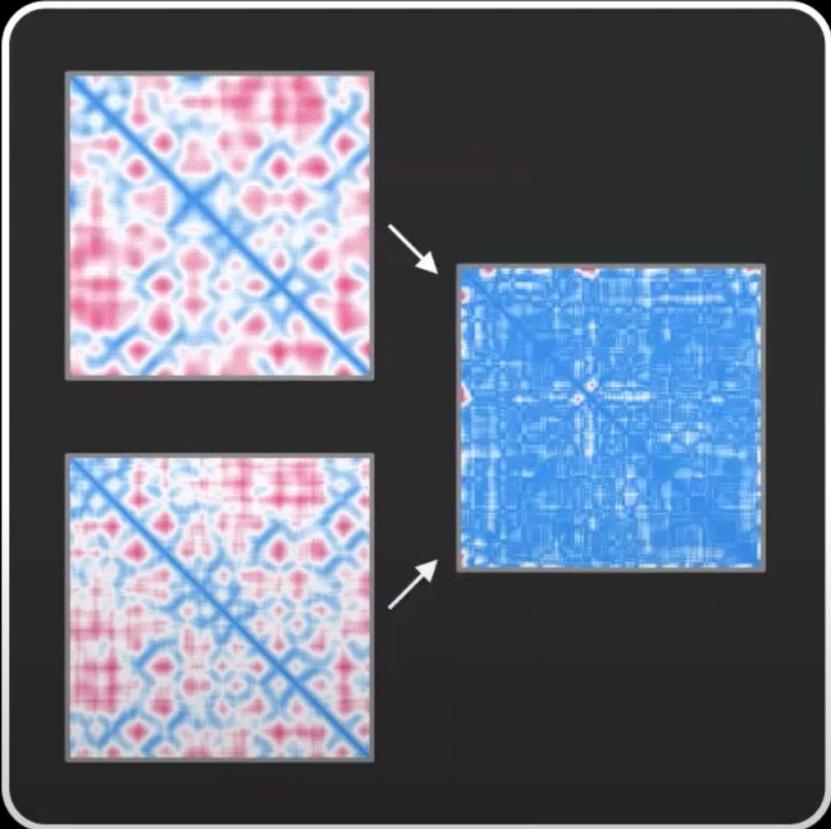






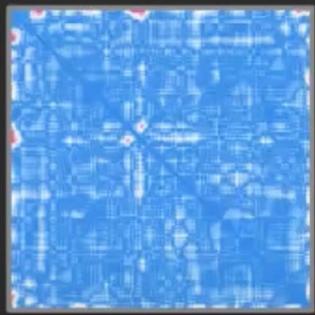


$$\tilde{d}_{j,k} = \|c_j - c_k\|_2$$



$$\tilde{d}_{j,k} = \|c_j - c_k\|_2$$

$$d_{j,k} = d_{j,k}^{(exp)} - d_{j,k}^{(pred)}$$

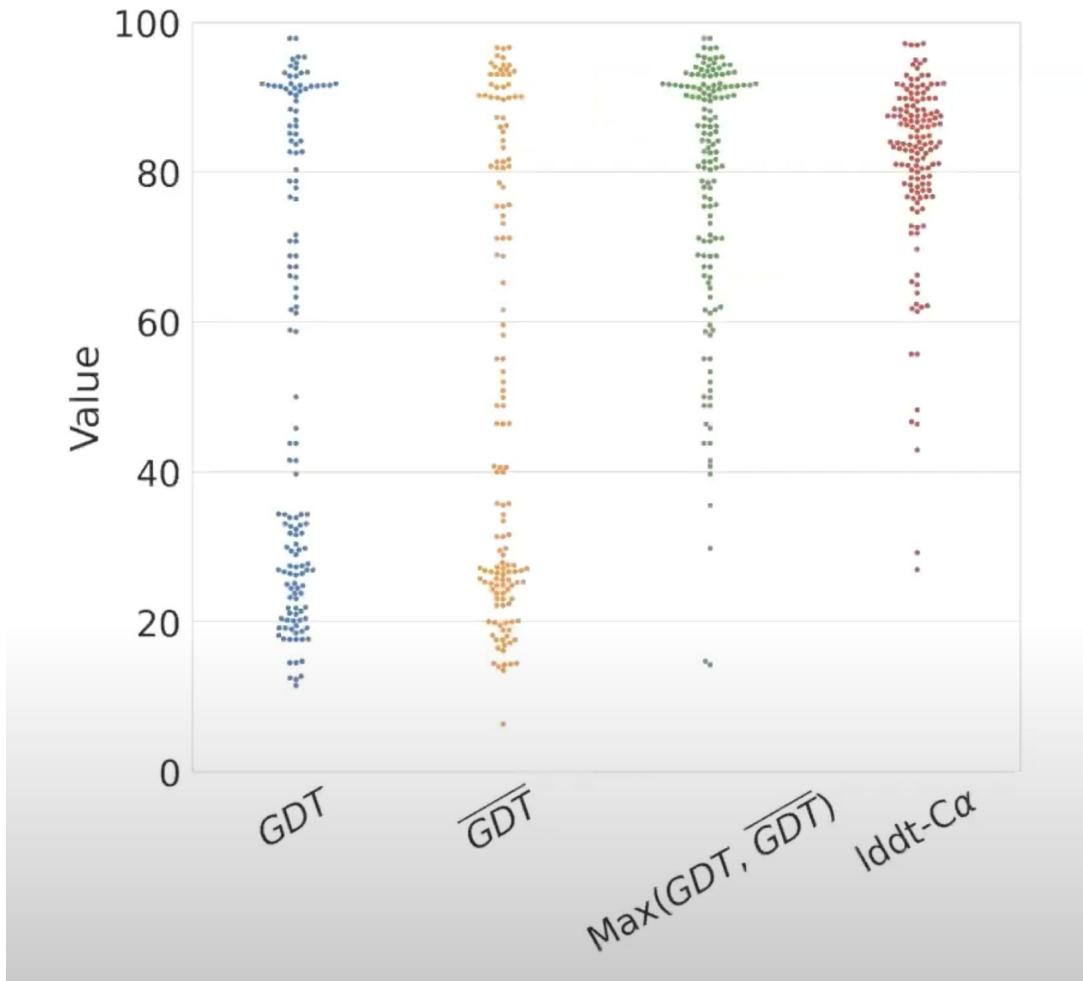


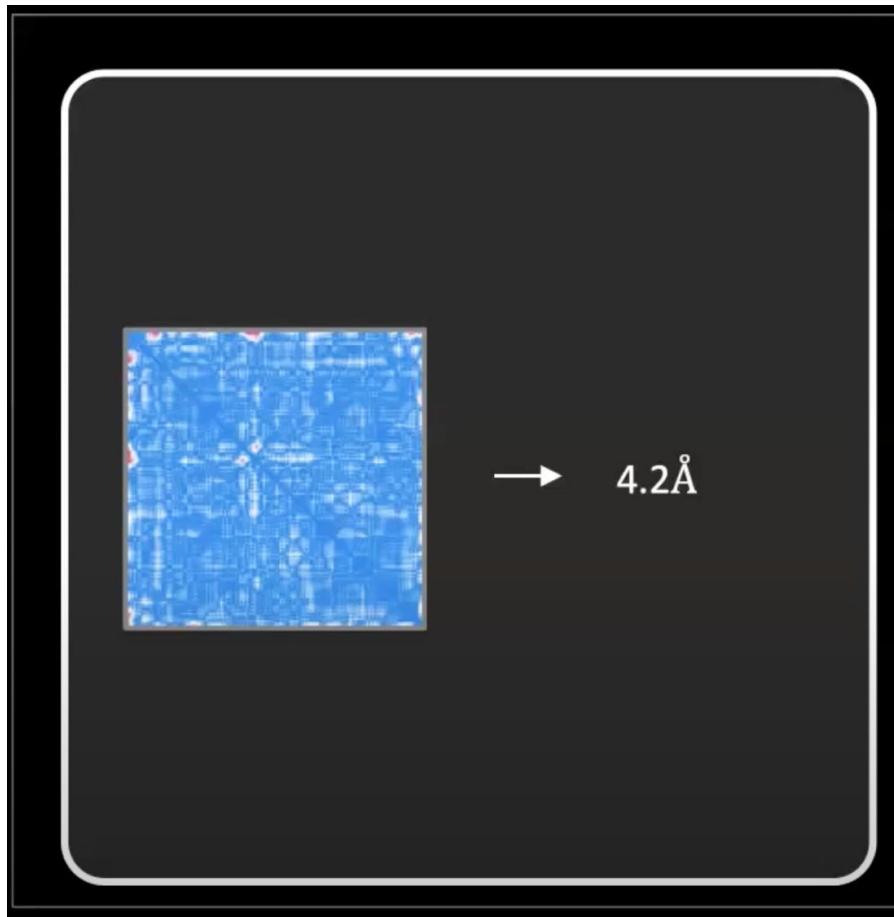
→ 4.2Å

$$\tilde{d}_{j,k} = \|c_j - c_k\|_2$$

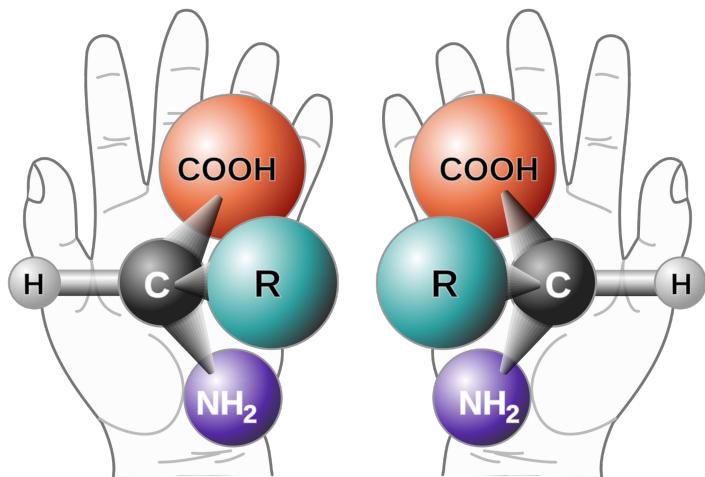
$$d_{j,k} = d_{j,k}^{(exp)} - d_{j,k}^{(pred)}$$

$$dRMSD = \frac{\|D\|_2}{L(L-1)}$$

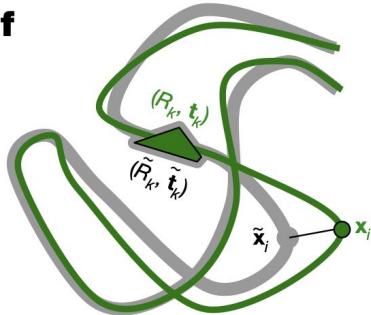




## Chirality



# FAPE Loss



FAPE loss is the most crucial loss of AF2. This loss aligns two different residue frames and compute the structural loss between other residues.

---

**Algorithm 28** Compute the Frame aligned point error
 

---

**def** computeFAPE( $\{T_i\}, \{\vec{x}_j\}, \{T_i^{\text{true}}\}, \{\vec{x}_j^{\text{true}}\}, Z = 10\text{\AA}, d_{\text{clamp}} = 10\text{\AA}, \epsilon = 10^{-4}\text{\AA}^2$ ) :

$$T_i, T_i^{\text{true}} \in (\mathbb{R}^{3 \times 3}, \mathbb{R}^3)$$

$$\vec{x}_j, \vec{x}_j^{\text{true}} \in \mathbb{R}^3,$$

$$i \in \{1, \dots, N_{\text{frames}}\}, j \in \{1, \dots, N_{\text{atoms}}\}$$

- 1:  $\vec{x}_{ij} = T_i^{-1} \circ \vec{x}_j$   $\vec{x}_{ij} \in \mathbb{R}^3$
  - 2:  $\vec{x}_{ij}^{\text{true}} = T_i^{\text{true}-1} \circ \vec{x}_j^{\text{true}}$   $\vec{x}_{ij}^{\text{true}} \in \mathbb{R}^3$
  - 3:  $d_{ij} = \sqrt{\|\vec{x}_{ij} - \vec{x}_{ij}^{\text{true}}\|^2 + \epsilon}$   $d_{ij} \in \mathbb{R}$
  - 4:  $\mathcal{L}_{\text{FAPE}} = \frac{1}{Z} \text{mean}_{i,j}(\min(d_{\text{clamp}}, d_{ij}))$
  - 5: **return**  $\mathcal{L}_{\text{FAPE}}$
- 

FAPE computing algorithm

👉 Let's break this down step-by-step!

We want to compute the structural loss of our predicted atom position  $\vec{x}_j$  relative to frame  $T_i$ , compared to the ground truth atom position  $\vec{x}_j^{\text{true}}$  and frame  $T_i^{\text{true}}$ .

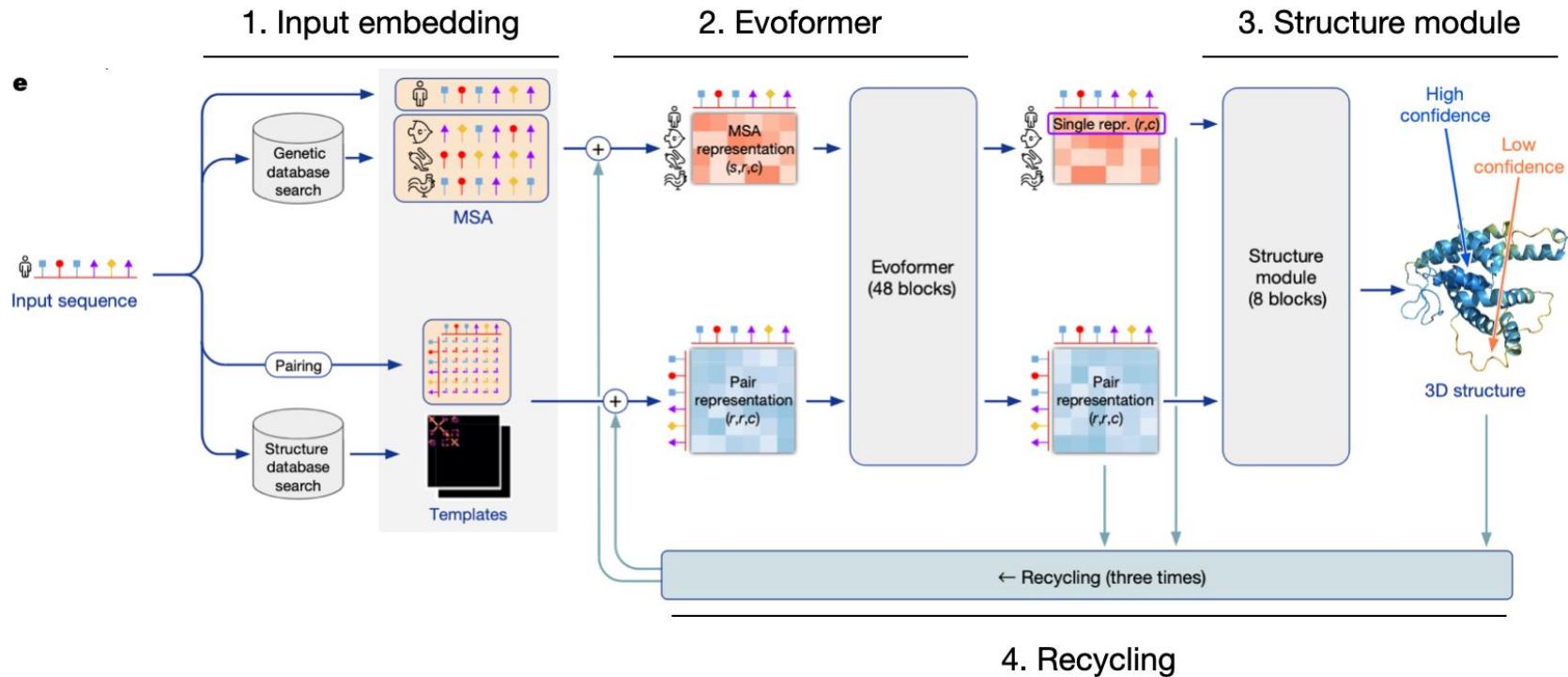
1. Align predicted  $j$ -th residue frame to predicted  $i$ -th residue frame.
2. Align true  $j$ -th residue frame to true  $i$ -th residue frame.
3. Compute distance between the two aligned  $j$ -th residue frames.
4. Calculate loss while clamping

- $\mathcal{L}_{FAPE}$  is FAPE loss
- $\mathcal{L}_{aux}$  is a loss from auxiliary metrics in structure module
- $\mathcal{L}_{dist}$  is loss from distogram prediction. AF2 tries to predict not only the distances between residues, but also distributions of distances, called *distograms*. Again, errors in those distributions are used as auxilliary losses.
- $\mathcal{L}_{msa}$  is a loss from BERT-like masked MSA position prediction. AlphaFold2 masks-out or mutates a position of alignment and tries to predict it in a self-supervised way.
- $\mathcal{L}_{conf}$  is a loss from model confidence in pLDDT prediction.
- $\mathcal{L}_{exp\ resolved}$  is a loss from a head, predicting, if structure comes from a highly accurate experimental prediction, such as CryoEM or high-resolution X-ray crystallography.
- $\mathcal{L}_{viol}$  is a loss for structural violations; penalizes unlikely lengths of bonds, torsion angles and sterically clashing atoms, even if ground truth structure is unavailable.

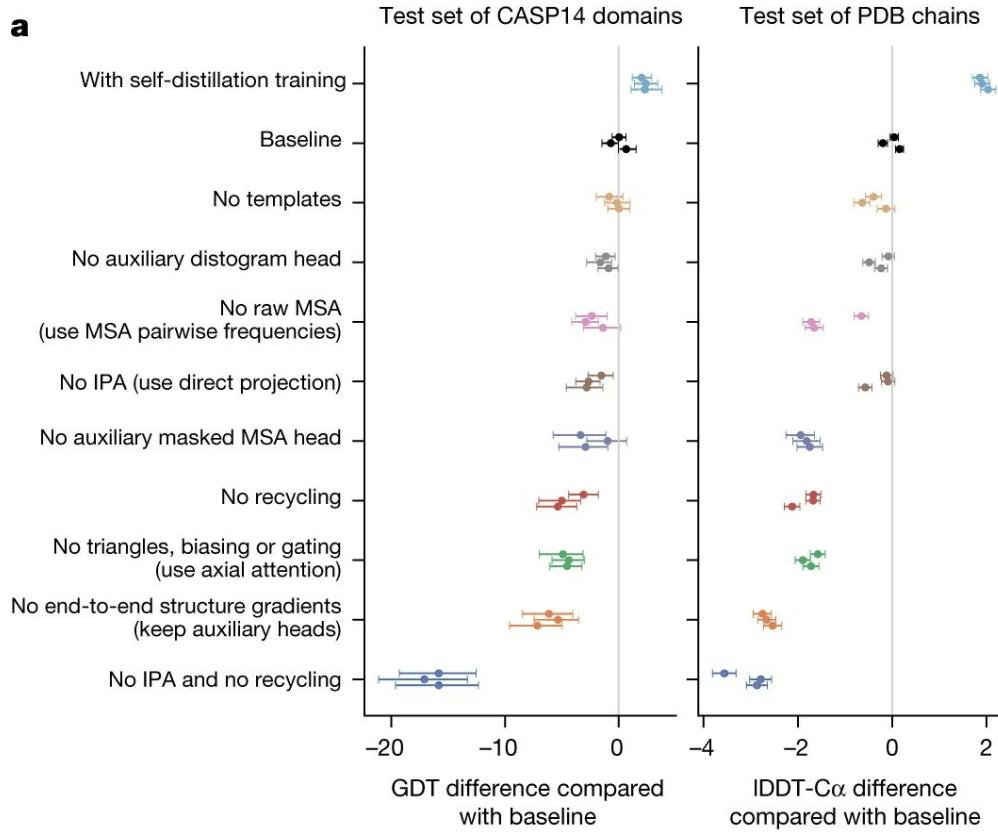
$$\mathcal{L} = \begin{cases} 0.5\mathcal{L}_{FAPE} + 0.5\mathcal{L}_{aux} + 0.3\mathcal{L}_{dist} + 2.0\mathcal{L}_{msa} + 0.01\mathcal{L}_{conf} & \text{training} \\ 0.5\mathcal{L}_{FAPE} + 0.5\mathcal{L}_{aux} + 0.3\mathcal{L}_{dist} + 2.0\mathcal{L}_{msa} + 0.01\mathcal{L}_{conf} + 0.01\mathcal{L}_{exp\ resolved} + 1.0\mathcal{L}_{viol} & \text{fine-tuning} \end{cases}$$

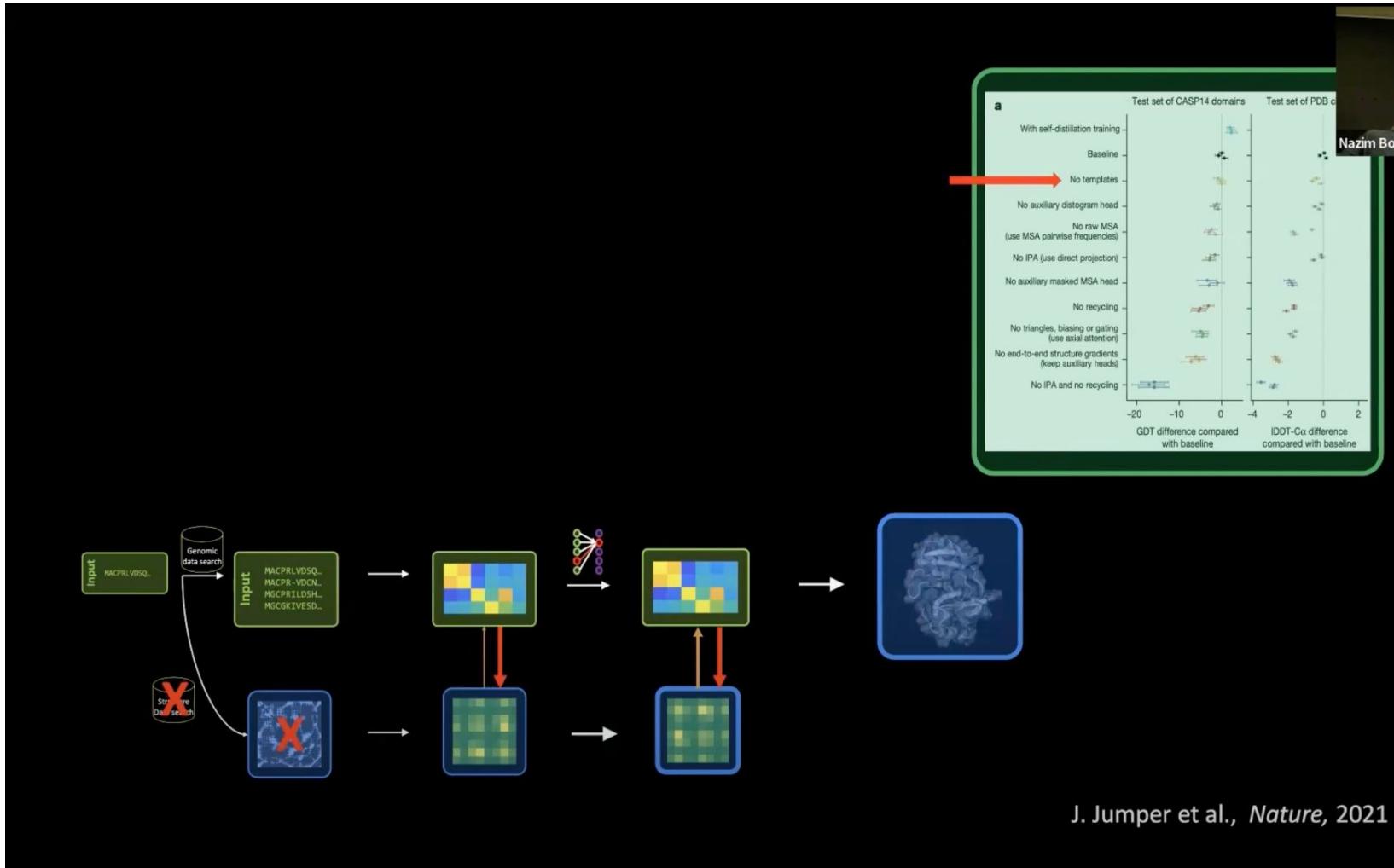
AlphaFold2 losses

# Recycling

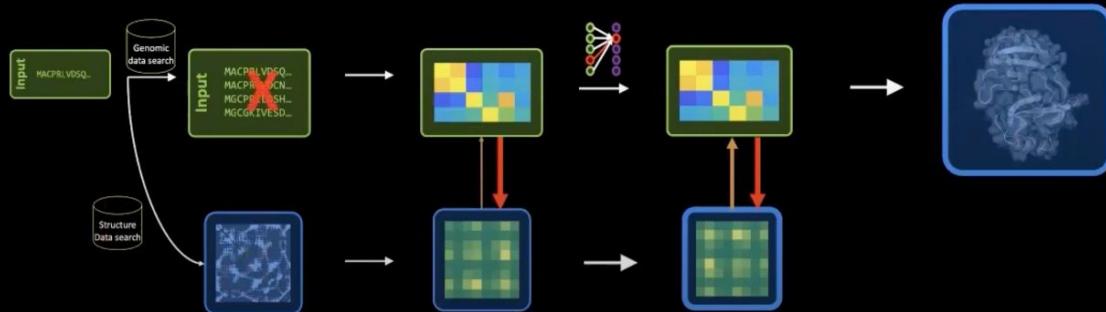
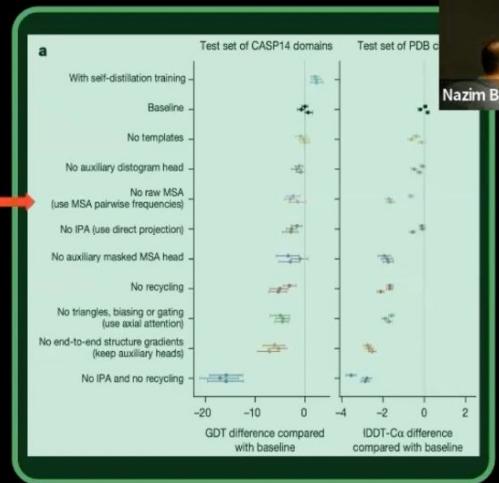


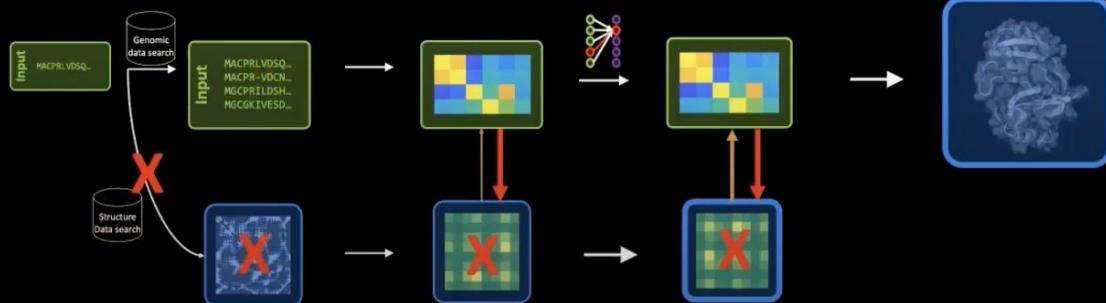
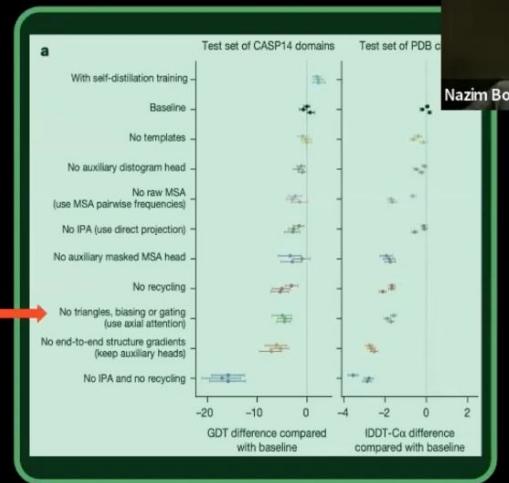
# Ablation Studies



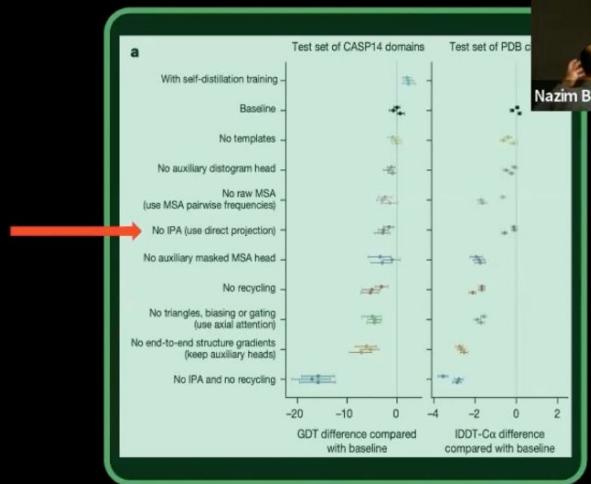
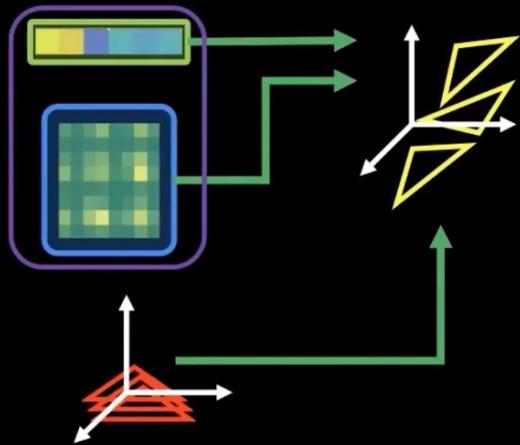


J. Jumper et al., *Nature*, 2021

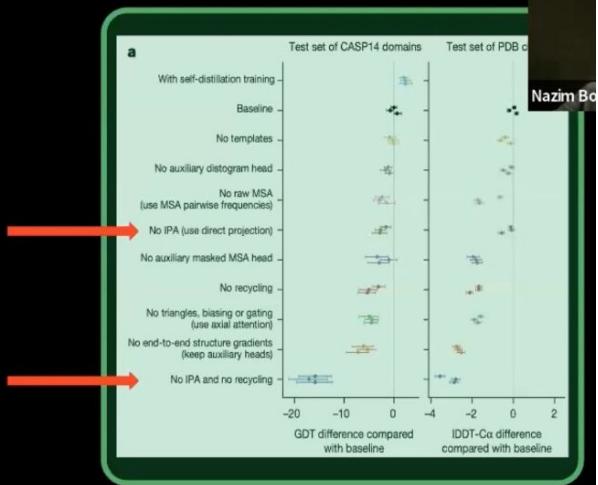
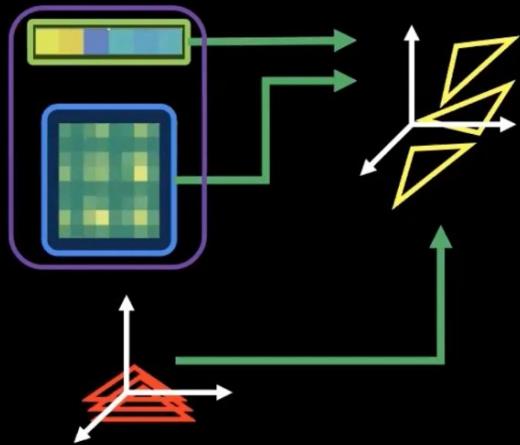




J. Jumper et al., *Nature*, 2021

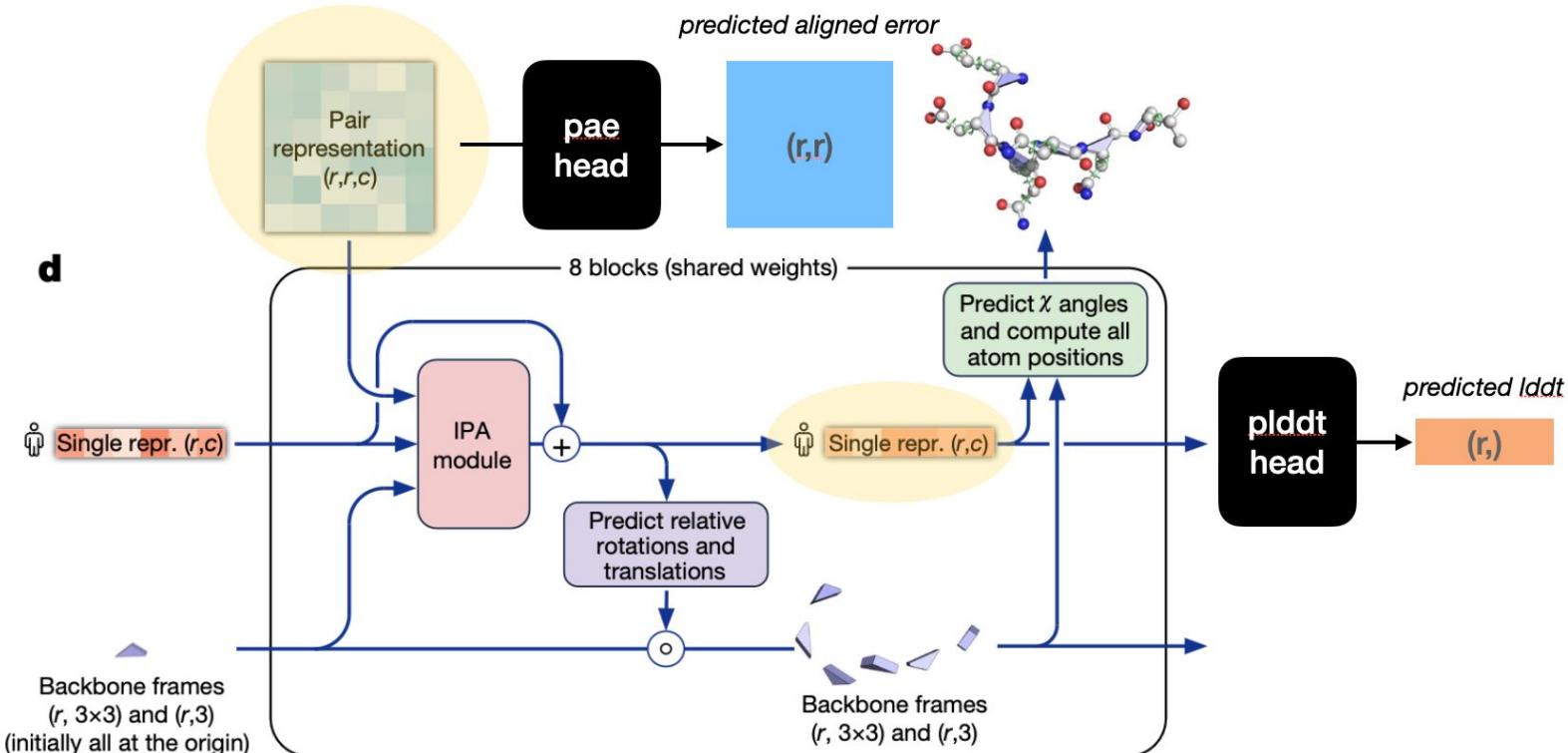


$$a_{ij} = \text{softmax}\left(q_i^T k_j + b_{ij} \textcolor{red}{X} \|T_i \circ \vec{q}_i - T_j \circ \vec{k}_j\|^2\right)$$



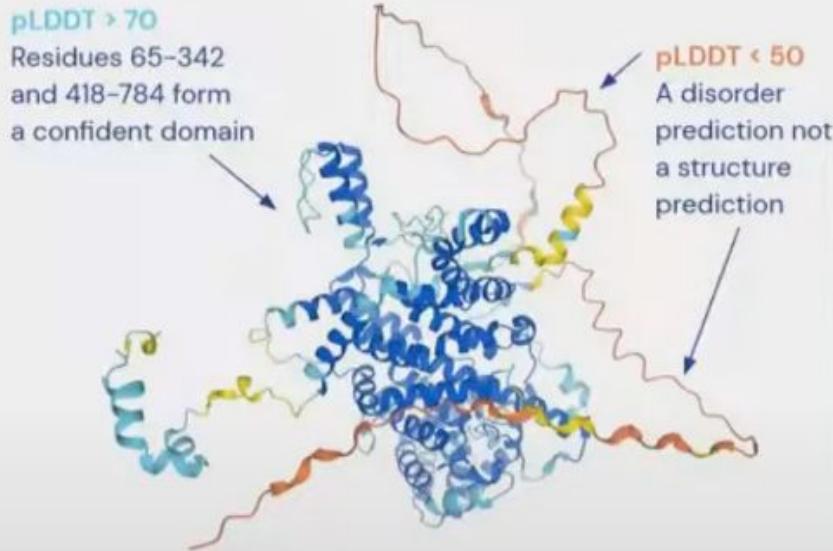
$$a_{ij} = \text{softmax}\left(q_i^T k_j + b_{ij} \textcolor{red}{X} \|T_i \circ \vec{q}_i - T_j \circ \vec{k}_j\|^2\right)$$

# Confidence Modules

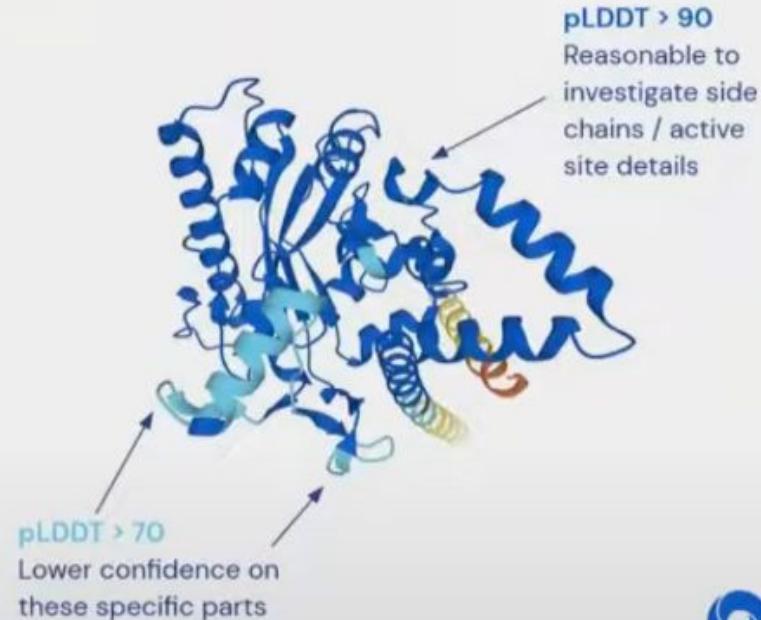


# Predicted LDDT (Predicted Local Distance Difference Test)

Identifying domains & possible disordered regions

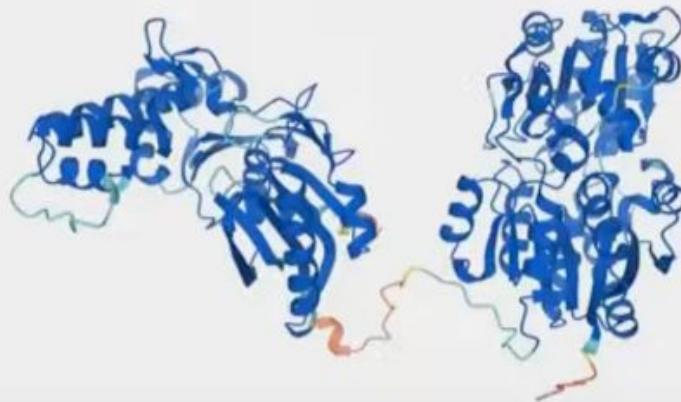


Assessing confidence within a domain



## Predicted LDDT: pitfalls

High pLDDT on all domains does **not** imply AlphaFold is confident of their relative positions



Assessing inter-domain confidence requires a different metric...



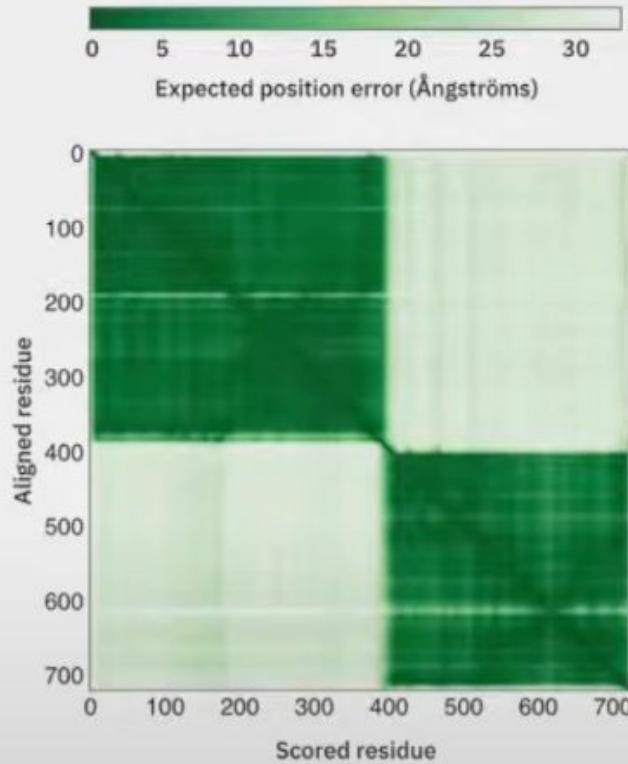
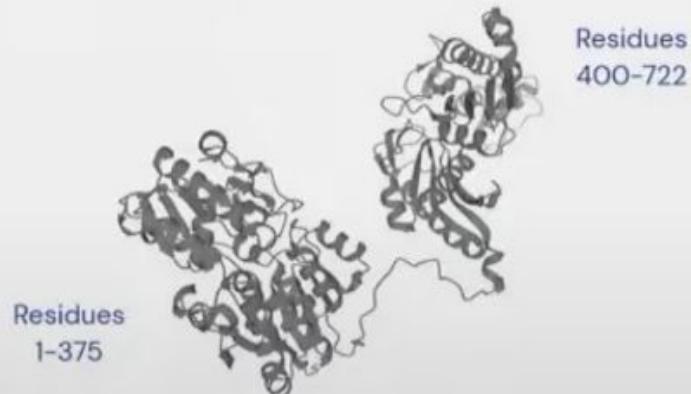
## Predicted Aligned Error

PAE is displayed as a 2D plot.

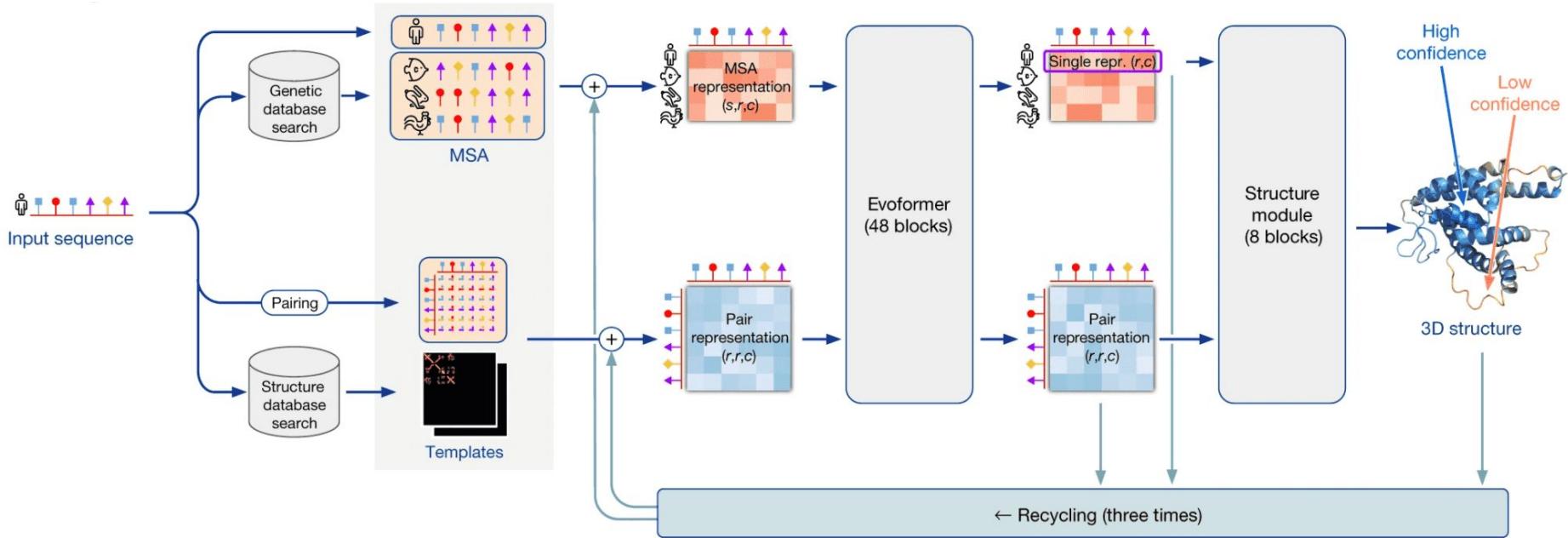
Suppose residue  $y$  were aligned to the true structure and we measured the position error at residue  $x$

The color at  $(x, y)$  is AlphaFold's prediction of that error

In this case the squares correspond to two domains



# Thank you for listening



# References

## Paper:

 [Highly accurate protein structure prediction with AlphaFold](#)

## Articles:

 [AlphaFold 2 is here: What's behind the structure prediction miracle?](#)

I'd say this is the first article to read after the main paper. It provides sufficient information about the model and offers insightful commentary.

 [From AlphaGo to AlphaFold, from games to science](#)

Notion style, detailed, easy to understand explanations.

 [Boris Burkov's Blog](#)

This is the most detailed and explanatory article I've found on the model architecture. It covers supplementary material extensively and was the most helpful for understanding the model's details.

 [The AlphaFold2 method paper: A fount of good ideas](#)

This article highlights specific aspects of the model and provides valuable insights.

## Videos:

 [Nazim Bouatta | Machine learning for protein structure prediction, Part 2: AlphaFold2 architecture](#)

A great explanation of AF2. It does an excellent job of justifying why certain design choices were made, though it doesn't go into the deepest architectural details. The best AF2 presentation I've seen.

 [Highly Accurate Protein Structure Prediction with AlphaFold | Simon Kohl](#)

A well-structured presentation that covers every part of the paper without diving into too much detail.