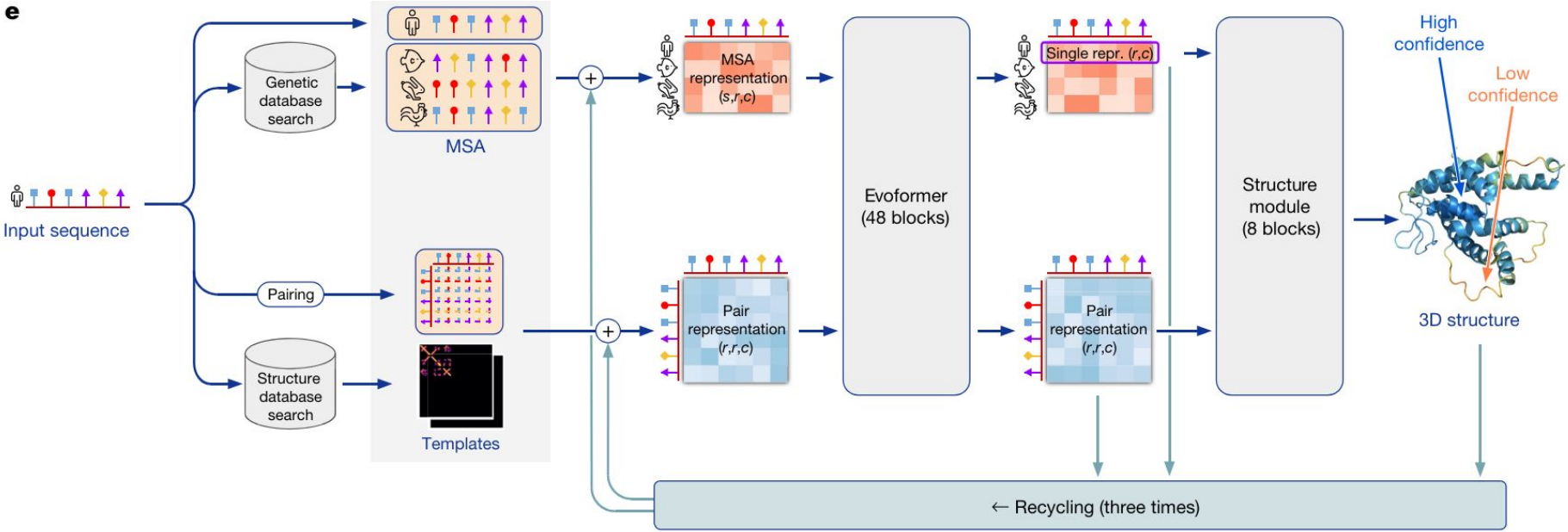
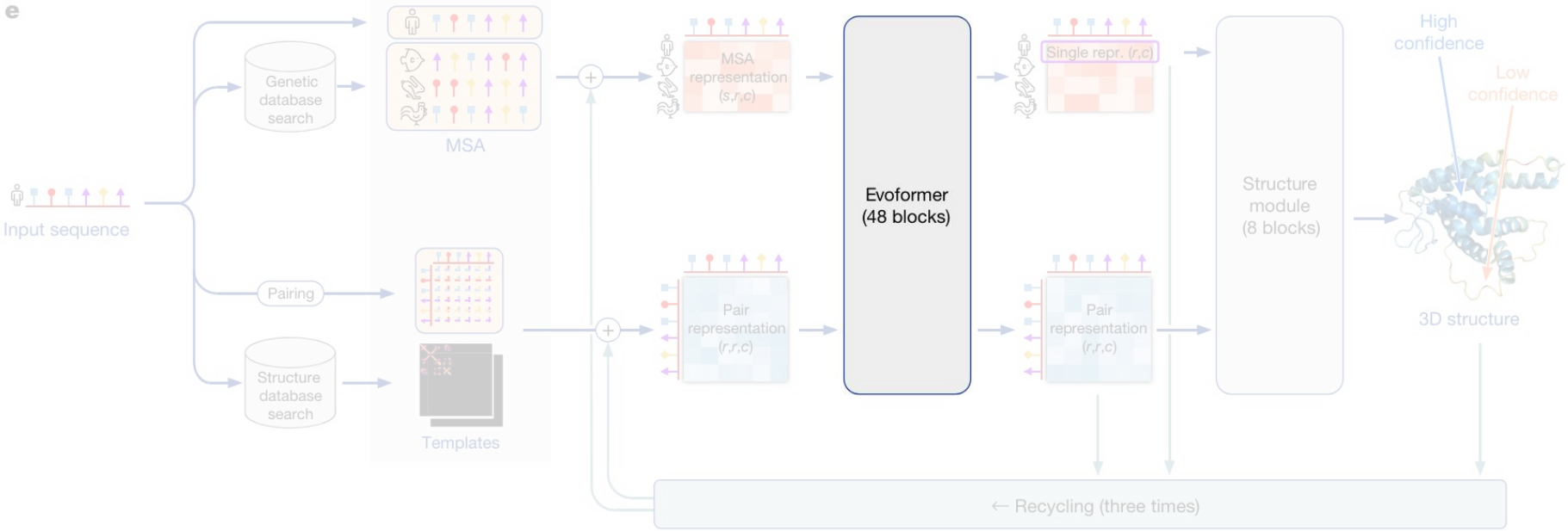
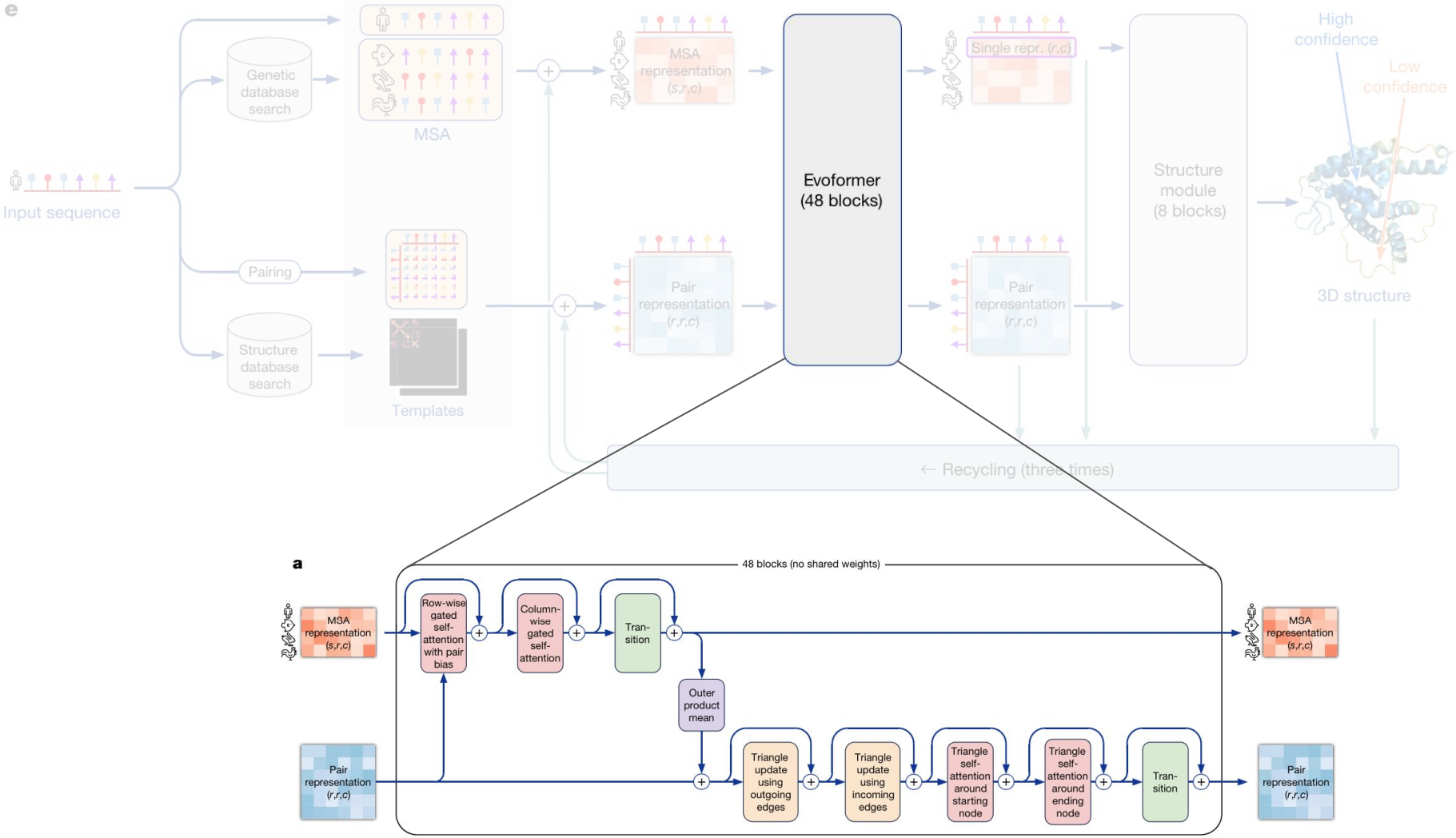
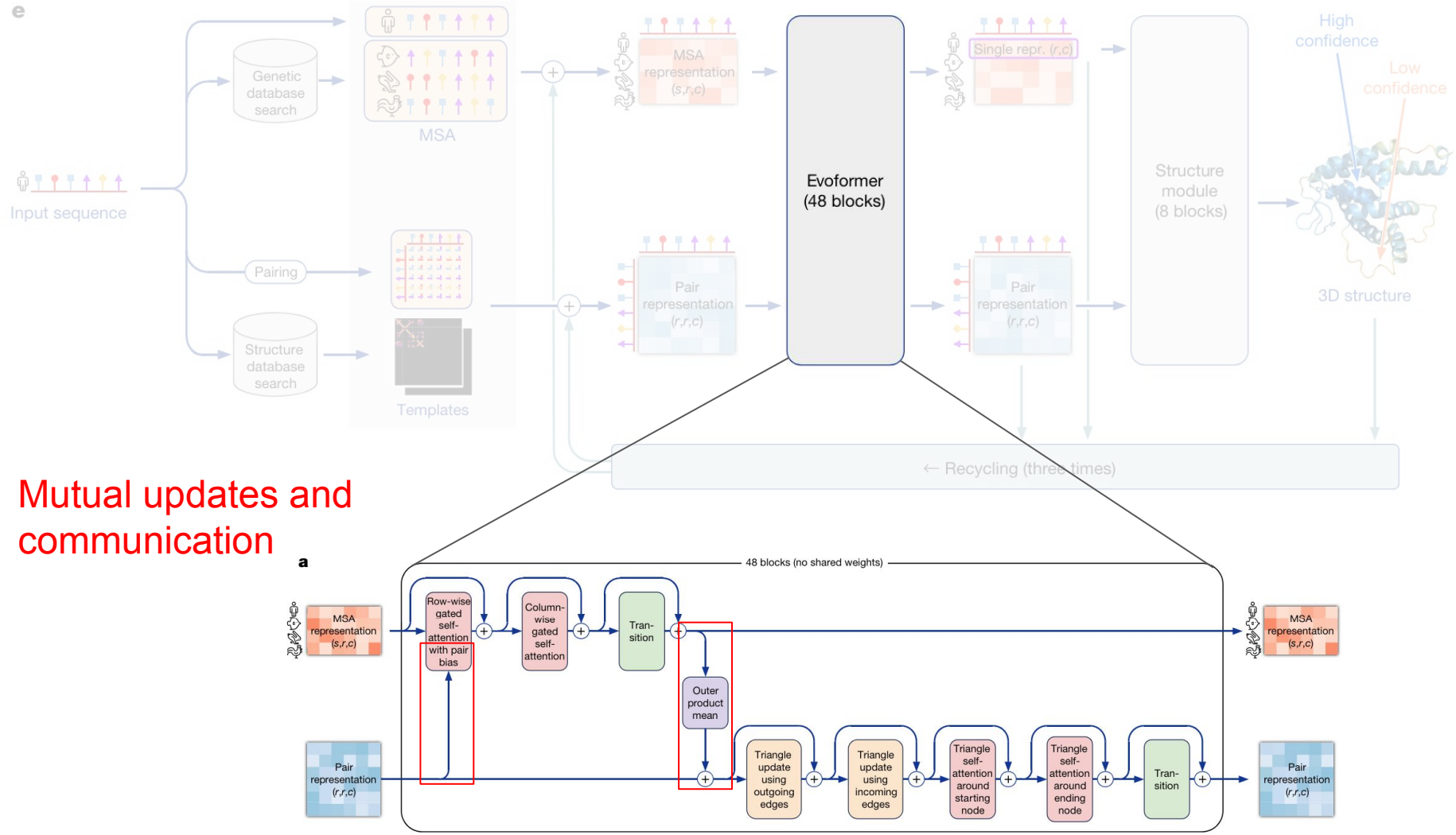
EvoFormer

Alpha Fold 2 in details


e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)


e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)


e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)
a
48 blocks (no shared weights)
MSA representation (s,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
Pair representation (r,r,c)
Triangle update using outgoing edges
Triangle update using incoming edges
Triangle self-attention around starting node
Triangle self-attention around ending node
Tran-sition
MSA representation (s,r,c)
Pair representation (r,r,c)



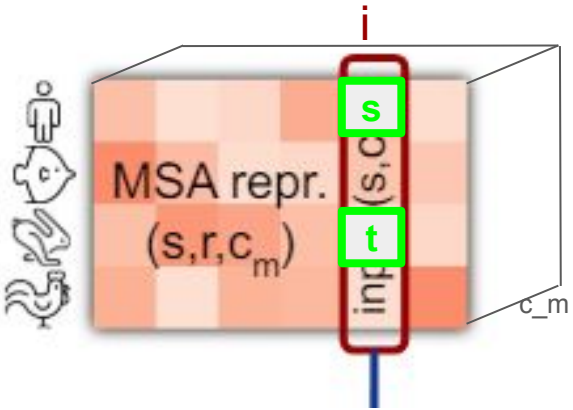
Mutual updates and communication

Notation

We use $\odot$ for the element-wise multiplication, $\otimes$ for the outer product, $\oplus$ for the outer sum, and $\mathbf{a}^\top \mathbf{b}$ for the dot product of two vectors. Indices i, j, k always operate on the residue dimension, indices s, t on the sequence dimension, and index h on the attention heads dimension. The channel dimension is implicit and we type the channel-wise vectors in bold, e.g. $\mathbf{z}_{ij}$. Algorithms operate on sets of such vectors, e.g. we use $\{\mathbf{z}_{ij}\}$ to denote all pair representations.

Note: **Gated Self-Attention**

A gating vector g is used to further modulate attention scores, effectively controlling how much influence each element should have

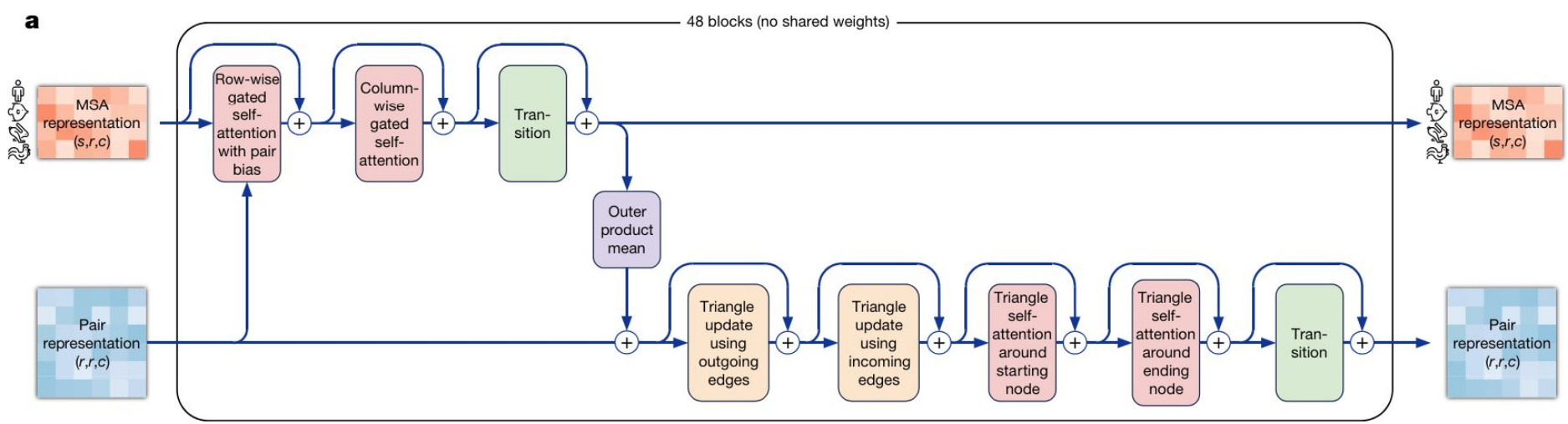


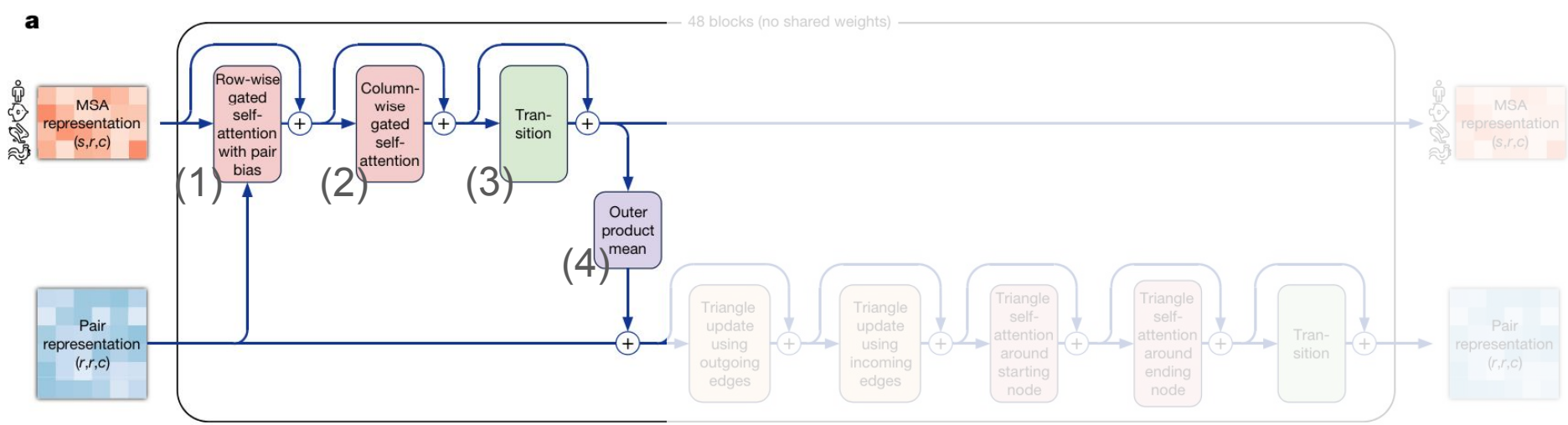
$$a^h_{sti} = \mathrm{softmax}_t\left(\frac{1}{\sqrt{c}}\,\mathbf{q}^{h\top}_{si}\mathbf{k}^h_{ti}\right)$$

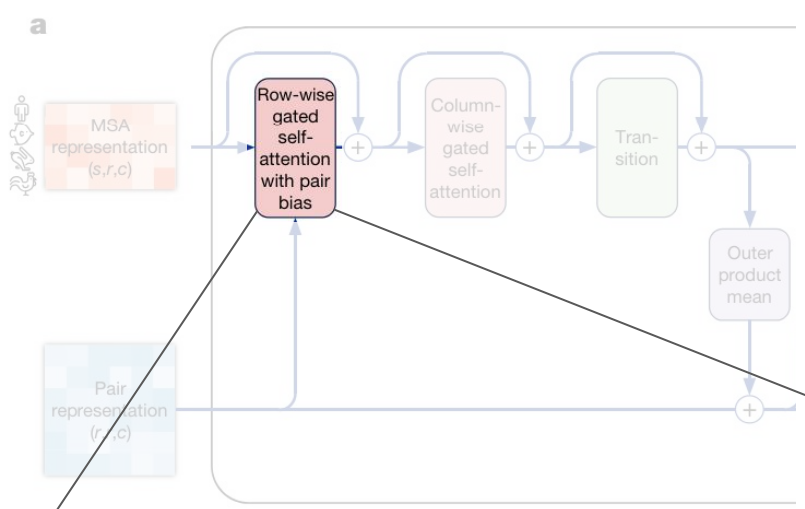
$$\mathbf{o}^h_{si} = \mathbf{g}^h_{si} \odot \sum_t a^h_{sti}\mathbf{v}^h_{st}$$

element-wise product

vanilla scaled dot-product attn


a
48 blocks (no shared weights)
MSA representation (s,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
Pair representation (r,r,c)
Triangle update using outgoing edges
Triangle update using incoming edges
Triangle self-attention around starting node
Triangle self-attention around ending node
Tran-sition
MSA representation (s,r,c)
Pair representation (r,r,c)


a
48 blocks (no shared weights)
MSA representation (s,r,c)
Pair representation (r,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
(1)
(2)
(3)
(4)
Triangle update using outgoing edges
Triangle update using incoming edges
Triangle self-attention around starting node
Triangle self-attention around ending node
Tran-sition
MSA representation (s,r,c)
Pair representation (r,r,c)



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

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

$\mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} = \text{LinearNoBias}(\mathbf{m}_{si})$ $\quad \mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} \in \mathbb{R}^c,\ h \in \{1, \dots, N_{\text{head}}\}$

$b^h_{ij} = \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$

$\mathbf{g}^h_{si} = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\quad \mathbf{g}^h_{si} \in \mathbb{R}^c$

 # Attention

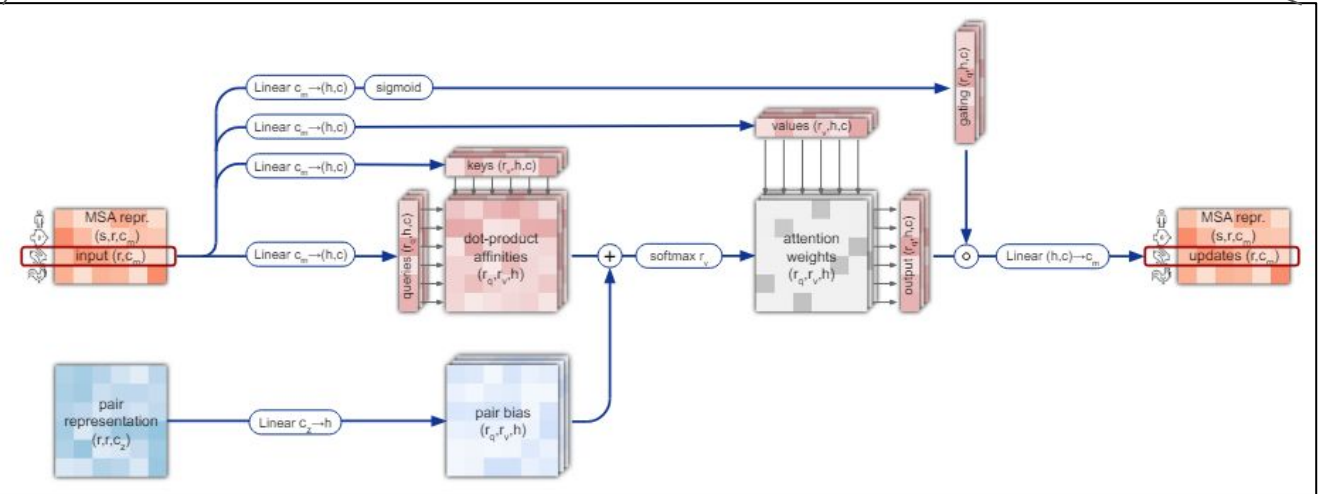
$a^h_{sij} = \text{softmax}_j\left(\frac{1}{\sqrt{c}}\, \mathbf{q}^{h\top}_{si} \mathbf{k}^h_{sj} + b^h_{ij}\right)$

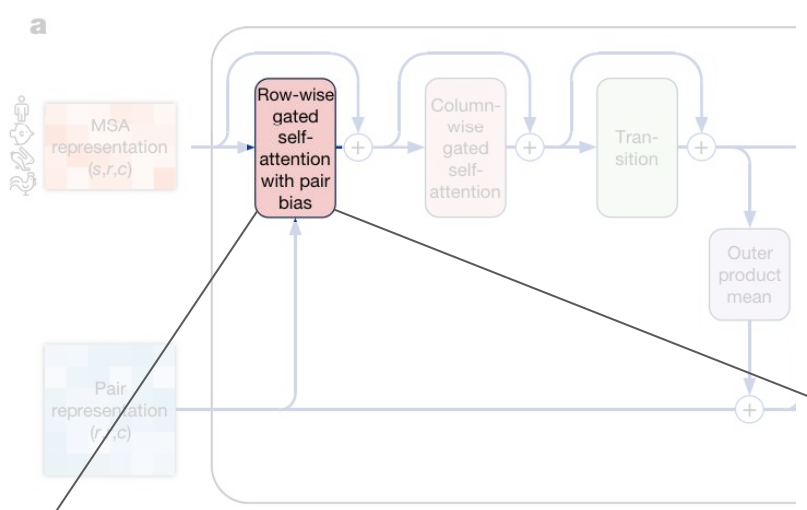
$\mathbf{o}^h_{si} = \mathbf{g}^h_{si} \odot \sum_j a^h_{sij} \mathbf{v}^h_{sj}$

 # Output projection

$\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}^h_{si})\right)$ $\quad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

return $\{\tilde{\mathbf{m}}_{si}\}$





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

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

$\mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} = \text{LinearNoBias}(\mathbf{m}_{si})$ $\quad \mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

$b^h_{ij} = \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$

$\mathbf{g}^h_{si} = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\quad \mathbf{g}^h_{si} \in \mathbb{R}^c$

 # Attention

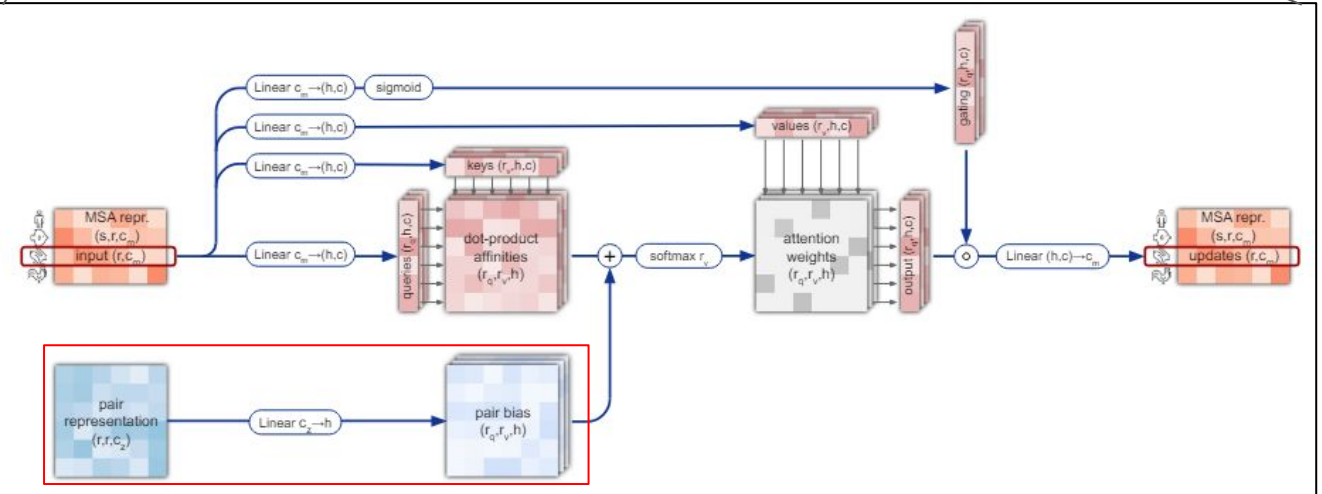
$a^h_{sij} = \text{softmax}_j\left(\frac{1}{\sqrt{c}}\, \mathbf{q}^{h\top}_{si} \mathbf{k}^h_{sj} + b^h_{ij}\right)$

$\mathbf{o}^h_{si} = \mathbf{g}^h_{si} \odot \sum_j a^h_{sij} \mathbf{v}^h_{sj}$

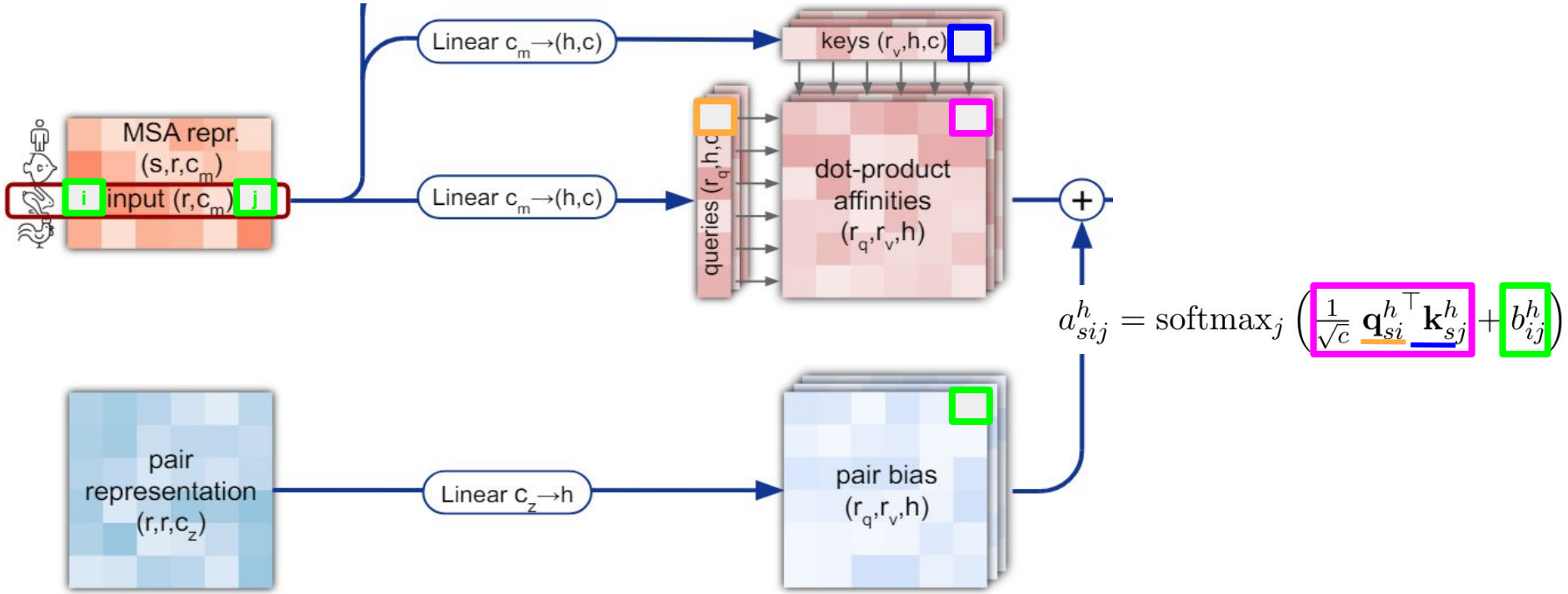
 # Output projection

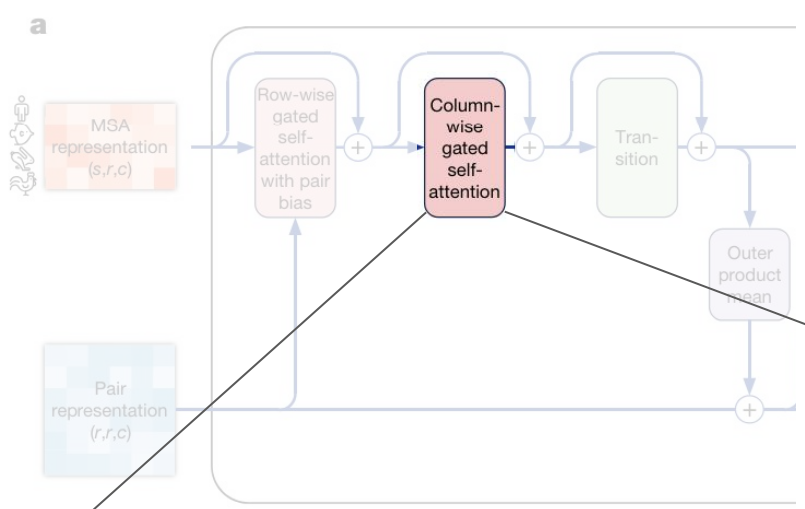
$\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}^h_{si})\right)$ $\quad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

return $\{\tilde{\mathbf{m}}_{si}\}$



Pair Representation influences MSA via a bias term in the row-wise Gated self-attn


MSA repr.
(s,r,c_m)
i
input (r,c_m)
j
Linear c_m→(h,c)
Linear c_m→(h,c)
keys (r_v,h,c)
queries (r_q,h,c)
dot-product
affinities
(r_q,r_v,h)
pair
representation
(r,r,c_z)
Linear c_z→h
pair bias
(r_q,r_v,h)
$a^h_{sij} = \mathrm{softmax}_j\left(\frac{1}{\sqrt{c}}\,\mathbf{q}^{h\top}_{si}\mathbf{k}^h_{sj} + b^h_{ij}\right)$



Algorithm 8 MSA column-wise gated self-attention

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

 # Input projections

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

$\mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} = \text{LinearNoBias}(\mathbf{m}_{si})$ $\quad \mathbf{q}^h_{si}, \mathbf{k}^h_{si}, \mathbf{v}^h_{si} \in \mathbb{R}^c,\ h \in \{1, \ldots, N_{\text{head}}\}$

$\mathbf{g}^h_{si} = \text{sigmoid}\left(\text{Linear}(\mathbf{m}_{si})\right)$ $\quad \mathbf{g}^h_{si} \in \mathbb{R}^c$

 # Attention

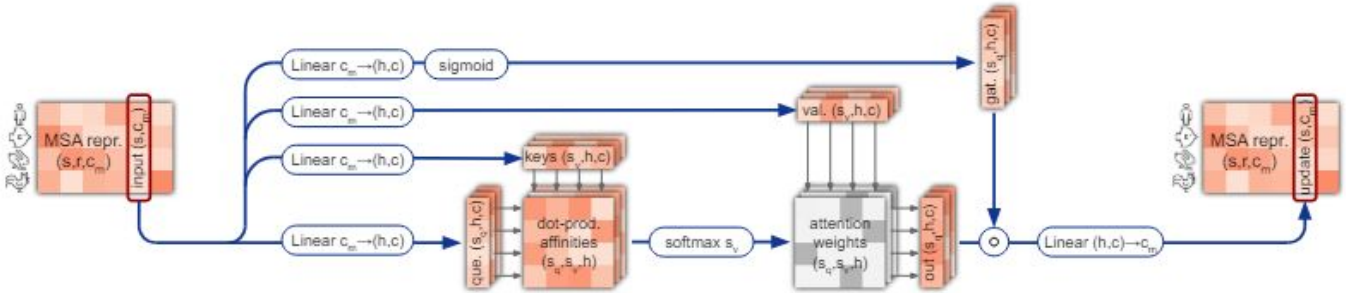
$a^h_{sti} = \text{softmax}_t\left(\frac{1}{\sqrt{c}}\, \mathbf{q}^{h\top}_{si} \mathbf{k}^h_{ti}\right)$

$\mathbf{o}^h_{si} = \mathbf{g}^h_{si} \odot \sum_t a^h_{sti} \mathbf{v}^h_{st}$

 # Output projection

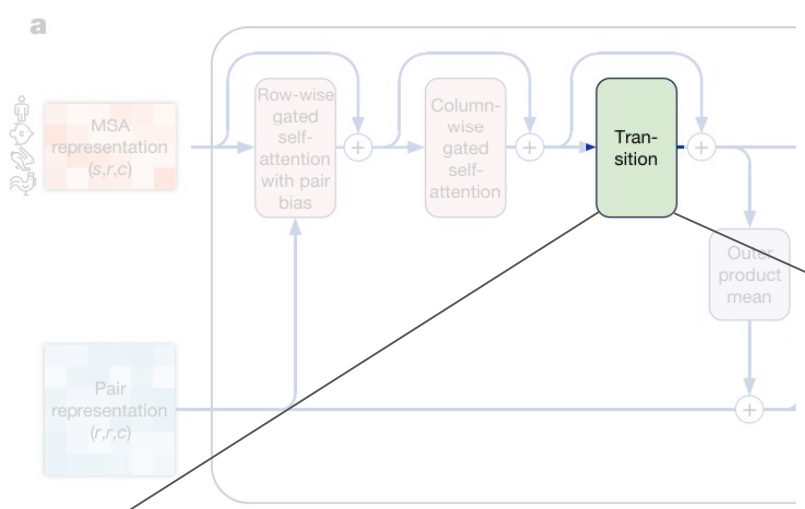
$\tilde{\mathbf{m}}_{si} = \text{Linear}\left(\text{concat}_h(\mathbf{o}^h_{si})\right)$ $\quad \tilde{\mathbf{m}}_{si} \in \mathbb{R}^{c_m}$

return $\{\tilde{\mathbf{m}}_{si}\}$



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

Vanilla
Column-wise
Gated self-attn



Algorithm 9 Transition layer in the MSA stack

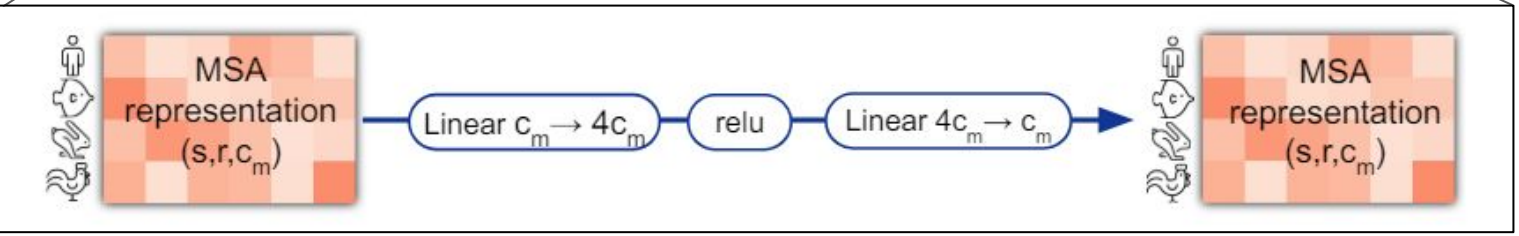
def MSATransition($\{\mathbf{m}_{si}\}, n = 4$) :

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

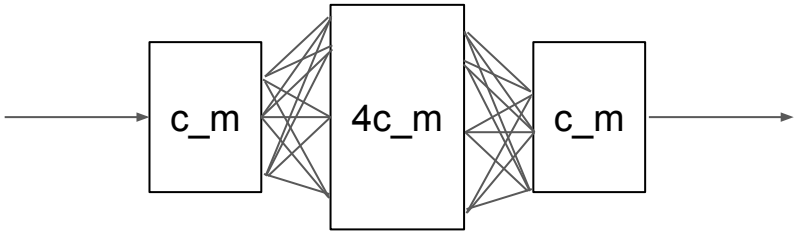
2: $\mathbf{a}_{si} = \text{Linear}(\mathbf{m}_{si})$ $\quad \mathbf{a}_{si} \in \mathbb{R}^{n \cdot c_m}$

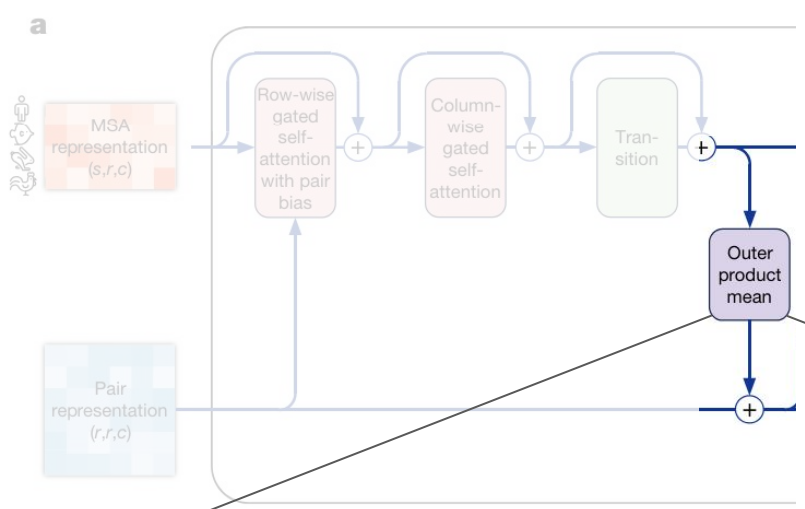
3: $\mathbf{m}_{si} \leftarrow \text{Linear}(\text{relu}(\mathbf{a}_{si}))$

4: **return** $\{\mathbf{m}_{si}\}$



1-layer MLP with relu activation





Algorithm 10 Outer product mean

def OuterProductMean($\{\mathbf{m}_{si}\}, c = 32$) :

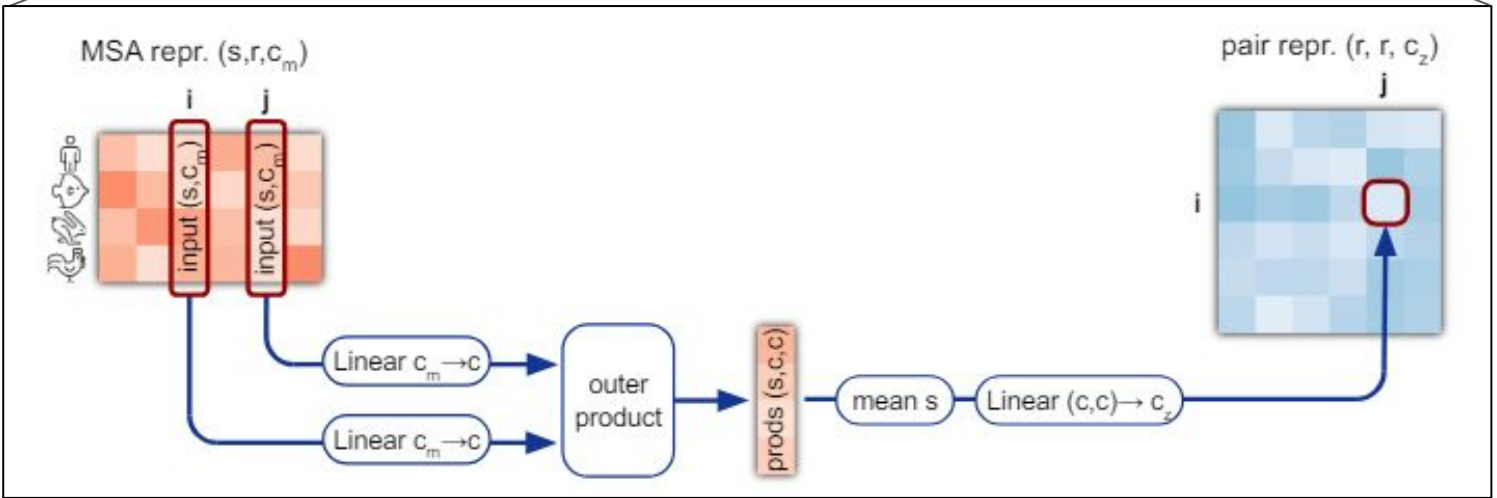
1: $\mathbf{m}_{si} \leftarrow \text{LayerNorm}(\mathbf{m}_{si})$

2: $\mathbf{a}_{si}, \mathbf{b}_{si} = \text{Linear}(\mathbf{m}_{si})$ $\quad \mathbf{a}_{si}, \mathbf{b}_{si} \in \mathbb{R}^{c}$

3: $\mathbf{o}_{ij} = \text{flatten}\left(\text{mean}_s(\mathbf{a}_{si} \otimes \mathbf{b}_{sj})\right)$ $\quad \mathbf{o}_{ij} \in \mathbb{R}^{c \cdot c}$

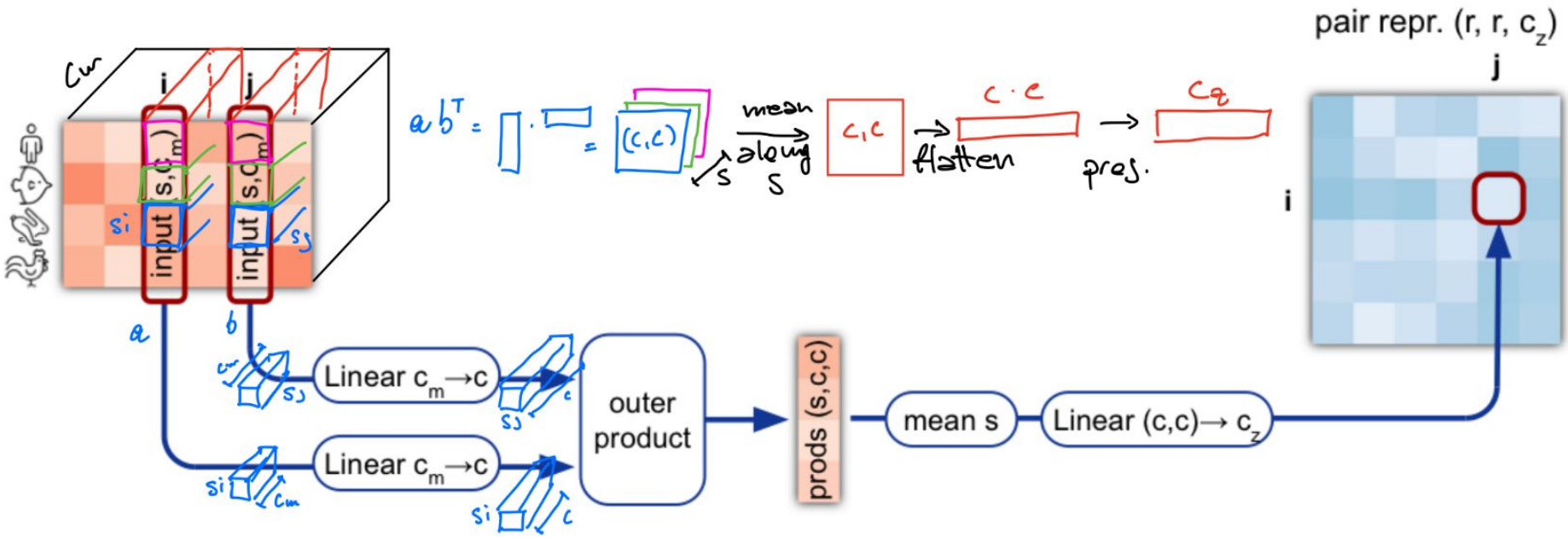
4: $\mathbf{z}_{ij} = \text{Linear}(\mathbf{o}_{ij})$ $\quad \mathbf{z}_{ij} \in \mathbb{R}^{c_z}$

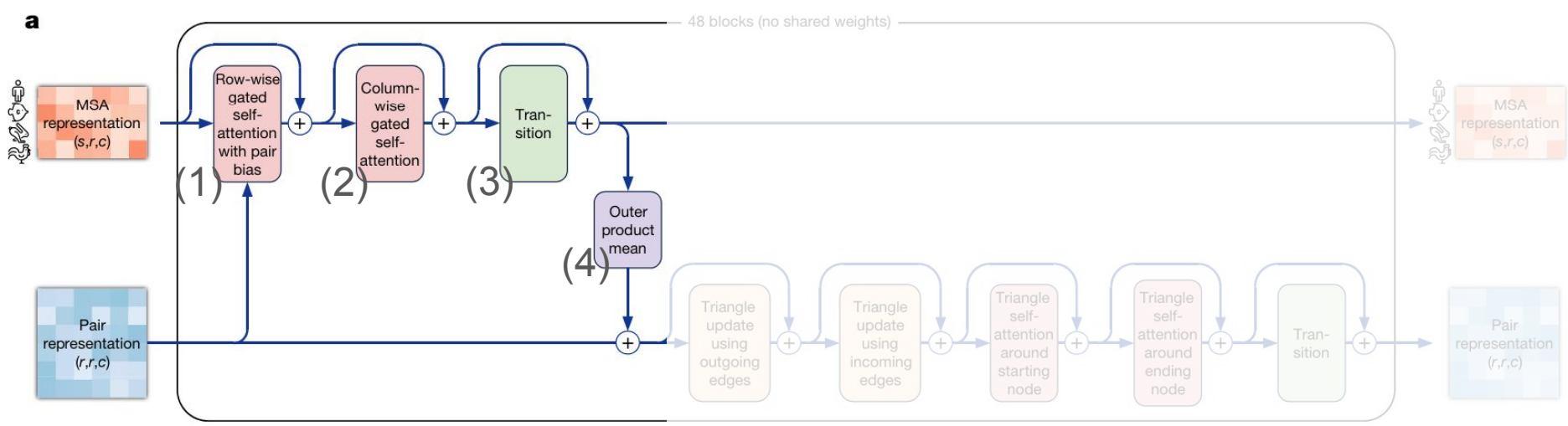
5: **return** $\{\mathbf{z}_{ij}\}$



MSA influences Pair Representation via outer-product update of each entry

Note: **Outer Product Mean**

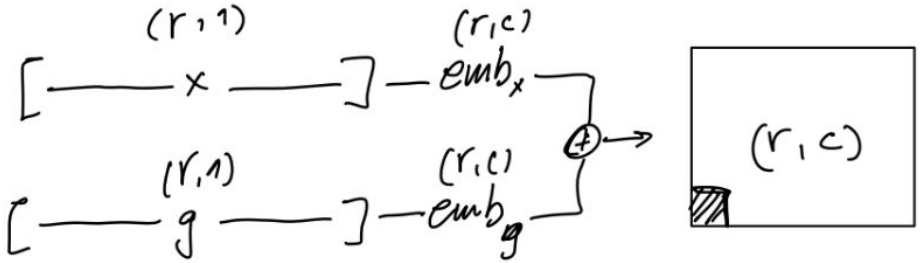


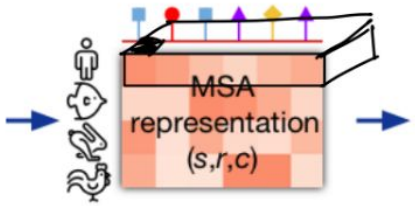

a
48 blocks (no shared weights)
MSA representation (s,r,c)
Pair representation (r,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
(1)
(2)
(3)
(4)
Triangle update using outgoing edges
Triangle update using incoming edges
Triangle self-attention around starting node
Triangle self-attention around ending node
Tran-sition
MSA representation (s,r,c)
Pair representation (r,r,c)

idea

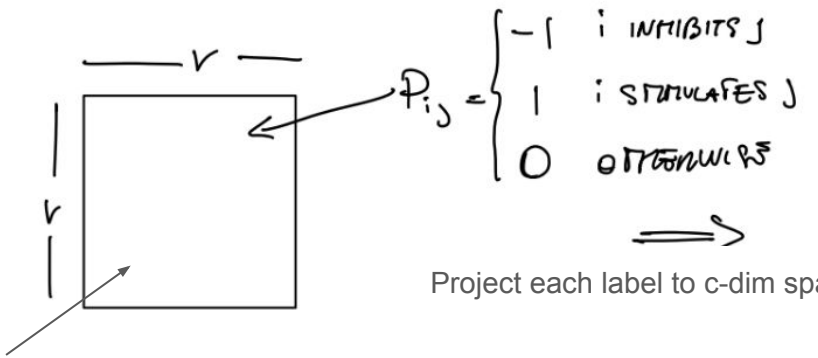
Idea

scRNA-seq:

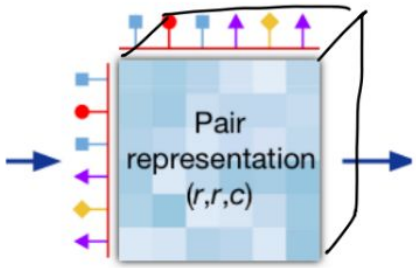




TFs Interactions:



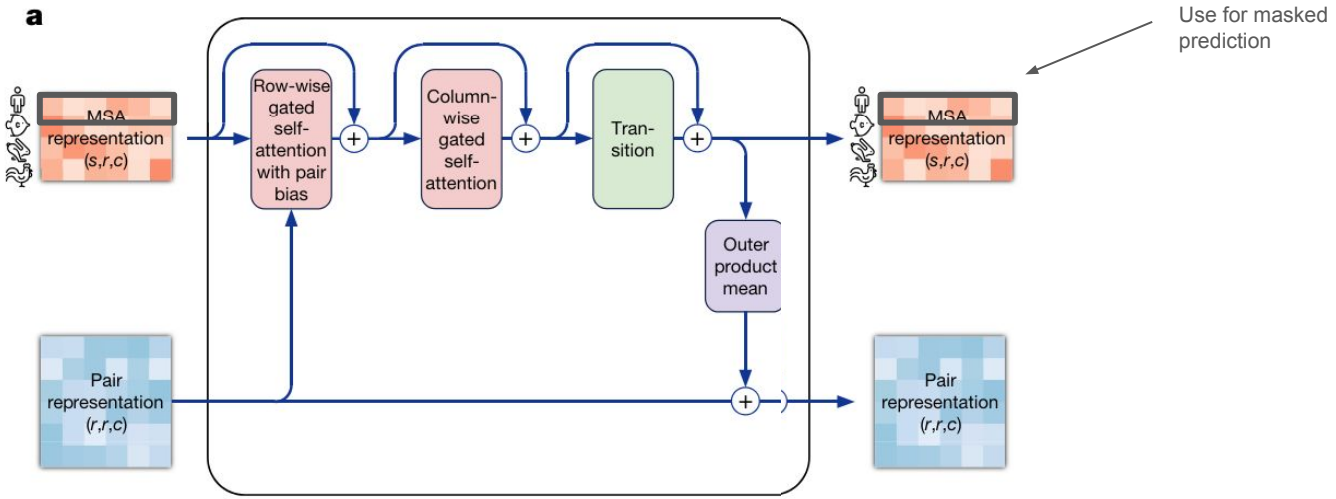
Project each label to c-dim space



Could add multiple layers (e.g. Tommaso's enrichment score), then project and sum as above

Idea

scRNA-seq:



TFs Interactions:

Thank You :)