AlphaFold: Algorithmic Building Blocks for Protein Structure Prediction

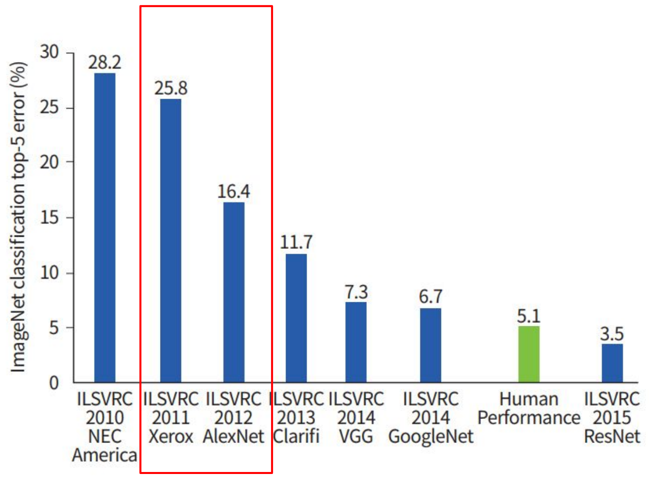
BIOE 145/245: Introduction to Machine Learning for Computational Biology
Spring 2024

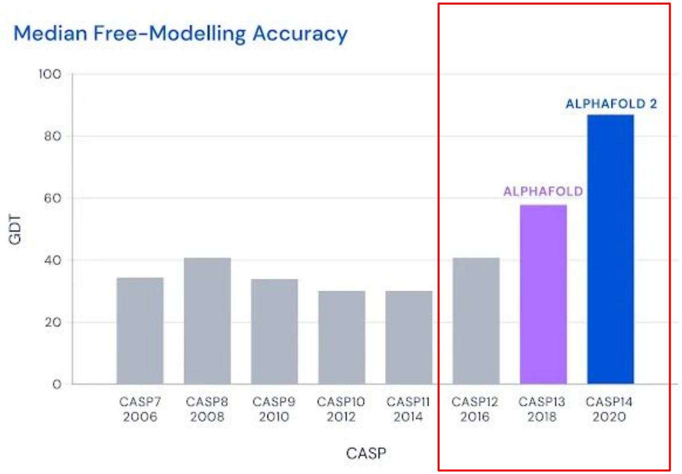
Guest Lecturer: Amy X. Lu
PhD Student
Department of Electrical Engineering and Computer Science
Berkeley Artificial Intelligence Lab

Agenda

1. The protein folding problem
2. AlphaFold2 and its internals
    a. Training data
    b. Evoformer
    c. Structure module
    d. Loss functions
3. Using AlphaFold
4. Limitations and open problems

AlphaFold: Biology's AlexNet moment?



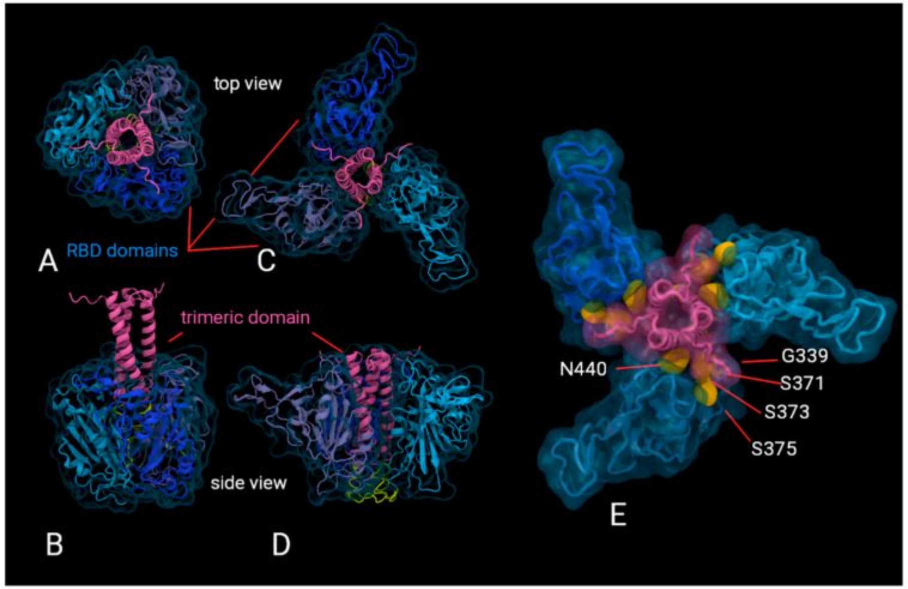


How can the key ingredients for AlphaFold's success be used in other AI for biology tasks?

Berkeley
UNIVERSITY OF CALIFORNIA

Why study AlphaFold?

- As a **tool** for forming biological hypotheses
  - Democratization of estimated structure unlocks new opportunities in understanding biological mechanisms
- As a **repertoire of techniques**
  - Transferable lessons for capturing inductive biases in scientific data
  - Bridges the adoption of machine learning methods for science



Example: AlphaFold-predicted structures help us hypothesize how sequence-level mutations in the SARS-CoV2 Omicron variant impacts its mechanism.
(Source: van Vuren et al., 2022)

Berkeley
UNIVERSITY OF CALIFORNIA

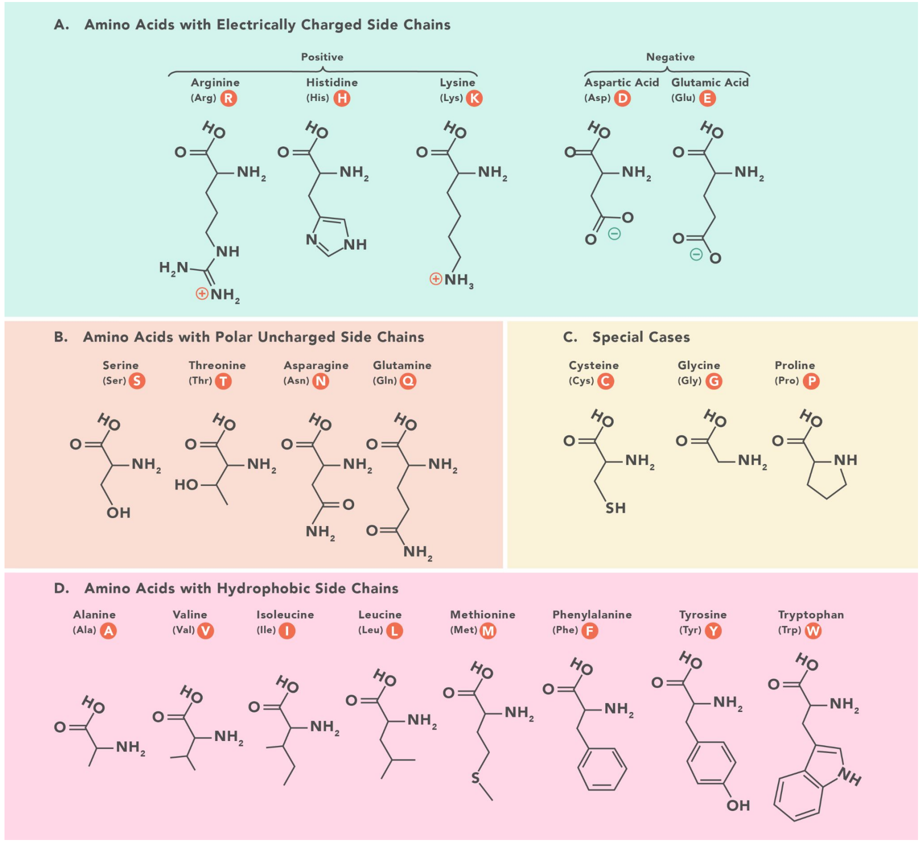
Background:

The Protein Folding Problem

tl;dr: Predict 3D coordinates from 1D string

What is a protein?

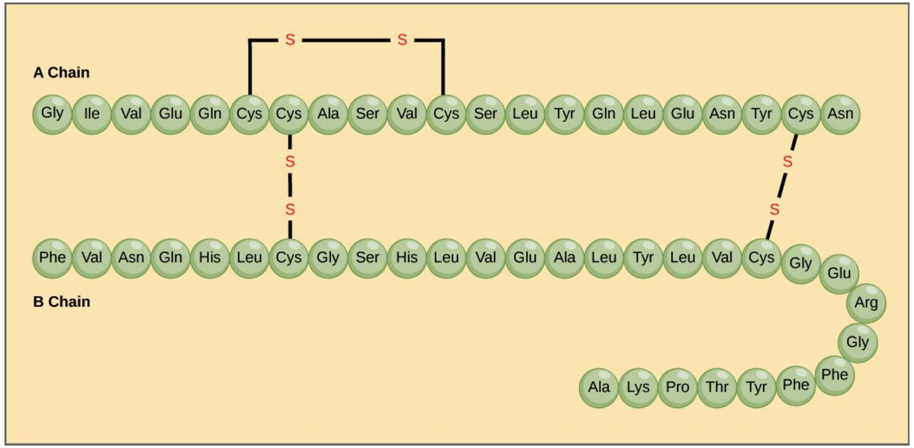
- Workhorse of biological functions
- Built as an assortment of 20 types of amino acids
    - Each amino acid consists of a distinct assortment of atoms



Berkeley UNIVERSITY OF CALIFORNIA

What is a protein *sequence*?

- Since there are only 20 amino acids, we can give each amino acid an letter as an abstraction
  - Allows for representation as a string, which is more amenable to computation


A Chain
Gly Ile Val Glu Gln Cys Cys Ala Ser Val Cys Ser Leu Tyr Gln Leu Glu Asn Tyr Cys Asn
S S S S S S
Phe Val Asn Gln His Leu Cys Gly Ser His Leu Val Glu Ala Leu Tyr Leu Val Cys Gly Glu Arg Gly Phe Phe Tyr Thr Pro Lys Ala
B Chain

```
>UniRef90_P02057 Hemoglobin subunit beta-1/2 n=6
Tax=Euarchontoglires TaxID=314146 RepID=HBB_RABIT
MVHLSSEEKSAVTALWGKVNVEEVGGEALGRLLVVYPWTQRFFESFGDLSSANAVMNNPK
VKAHGKKVLAAFSEGLSHLDNLKGTFAKLSELHCDKLHVDPENFRLLGNVLVIVLSHHFG
KEFTPQVQAAYQKVVAGVANALAHKYH
```

Berkeley UNIVERSITY OF CALIFORNIA

What is a protein *structure*?

- **Secondary structure:** Interactions between atoms cause the series of amino acids to form regular substructures
- **Tertiary structure**: secondary structures are assembled into folds in 3D space

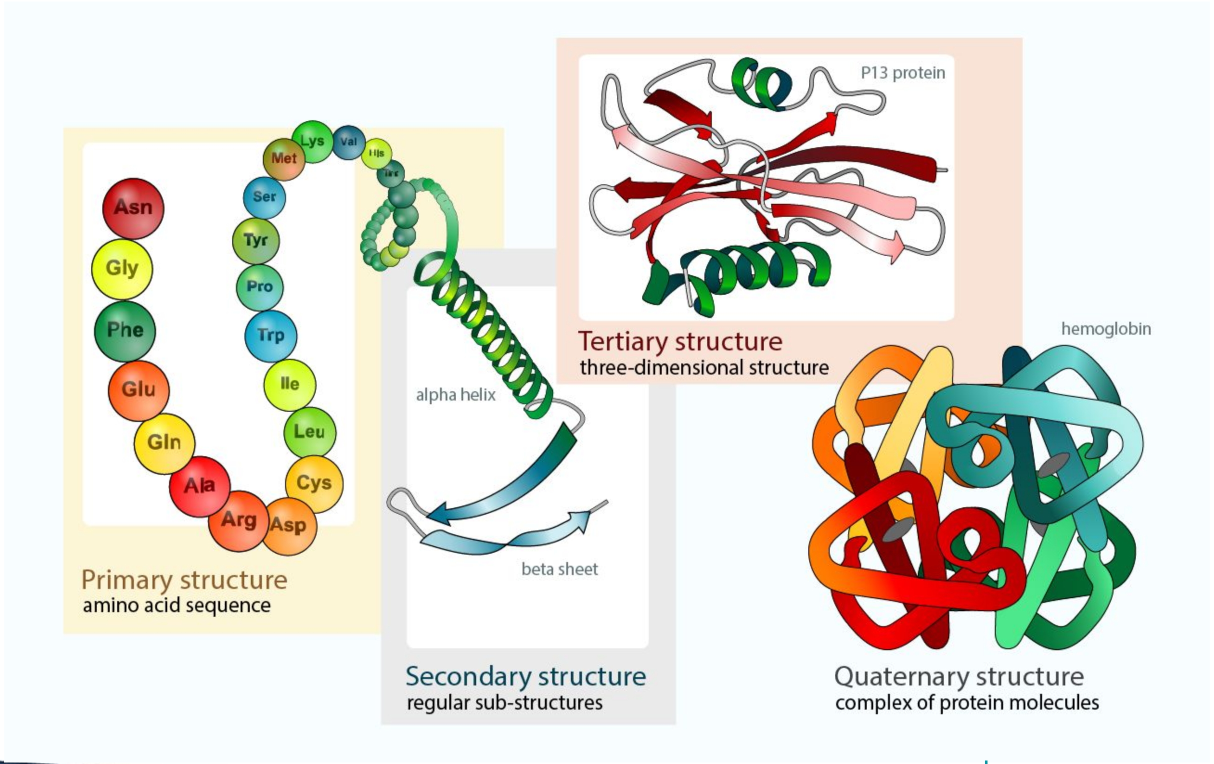
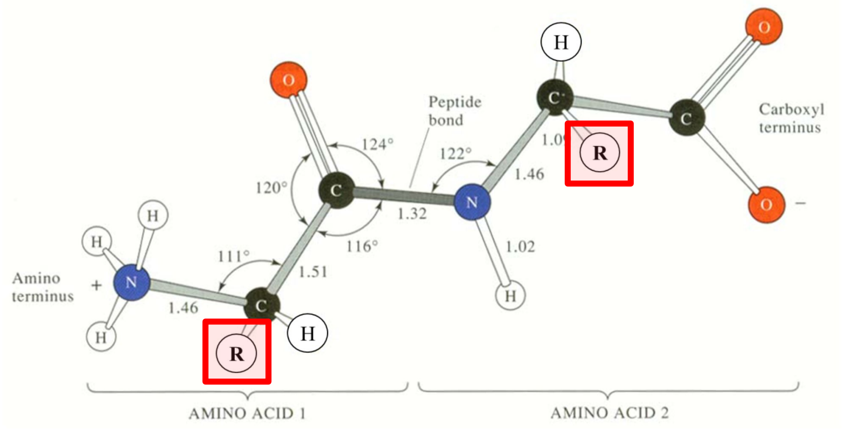


Image source

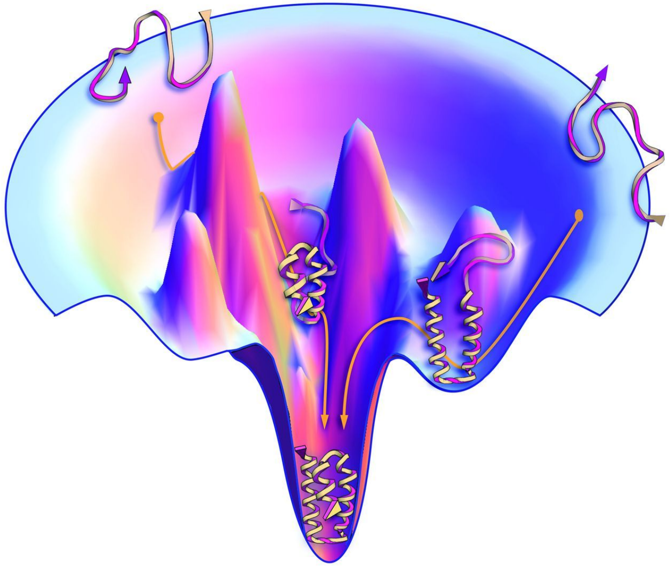
Backbone structure vs. *all-atom* structure



- Protein backbone atoms: (N, Cα, C)
- Side chain groups (denoted generically as R) determine the amino acid identity
  - Analogy: “protrudes” from the backbone like t-shirts on this clothesline, where color of the t-shirts defines the sequence.
- The **all-atom structure** include positions of the backbone and sidechain atoms.

The protein folding problem

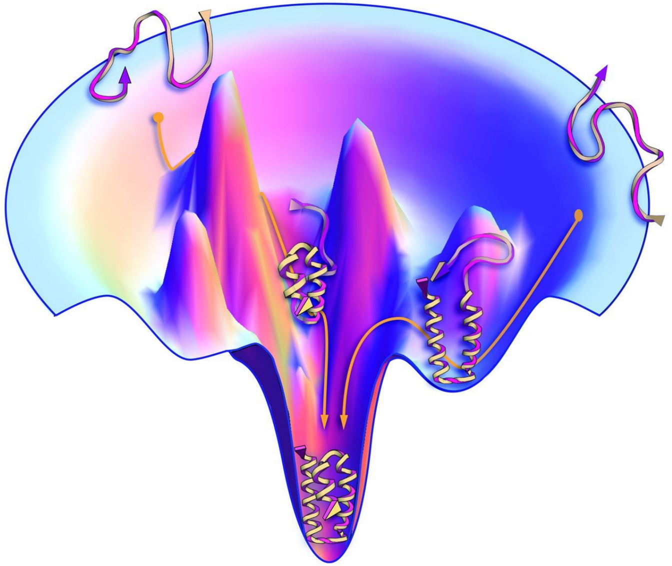
- Proteins seek to "fold" into a structure that minimizes free energy
- **Anfinsen's dogma:** 3D structure of (most) natural proteins is determined only by its amino acid sequence.
- Protein folding problem: predicting 3D structure from 1D sequence.



The protein folding problem

- **Levinthal's paradox:** searching through possible structural configurations for a protein should take longer than the age of the universe, yet proteins can fold in seconds.

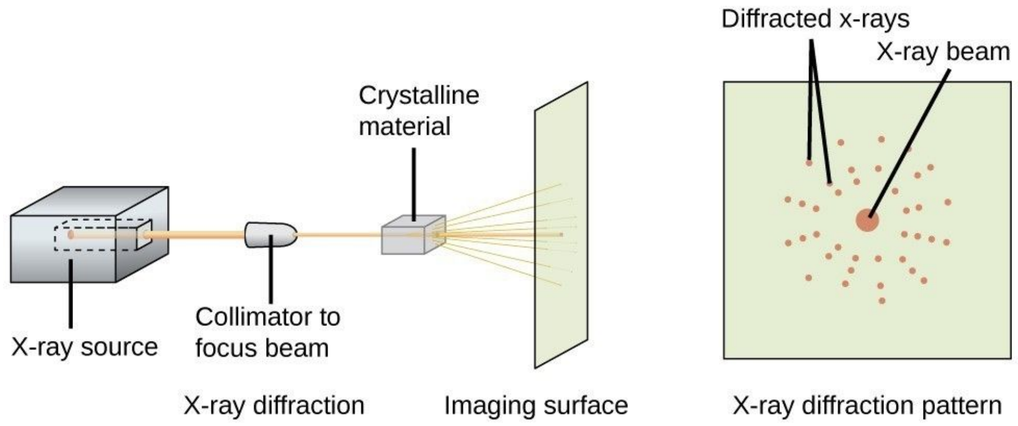
Determining structure from sequence alone should theoretically be possible from our understanding of nature, yet very difficult from our current understanding of biophysics.



Berkeley
UNIVERSITY OF CALIFORNIA

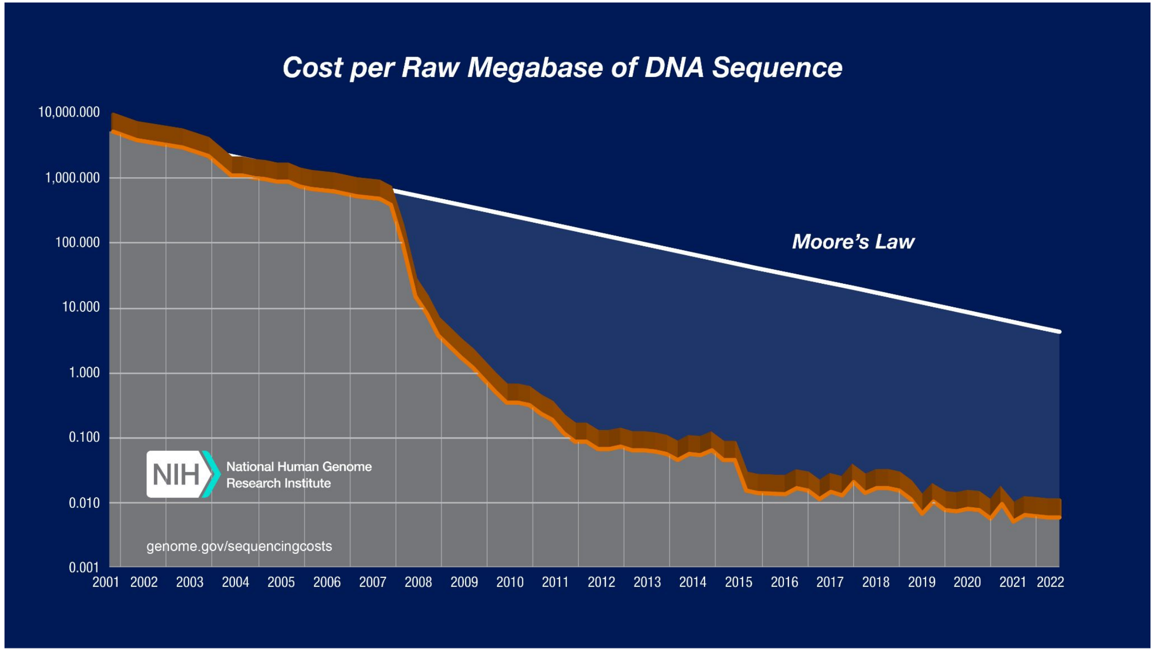
Experimentally-resolved structure is expensive & slow

- Structures typically **experimentally resolved** by X-ray crystallography, cryo-EM, etc.
- Can take up to months and years to resolve a single structure



Source:
https://university.pressbooks.pub/chemistryucf/chapter/10-8-x-ray-crystallography/

Sequence data is more abundant than structure

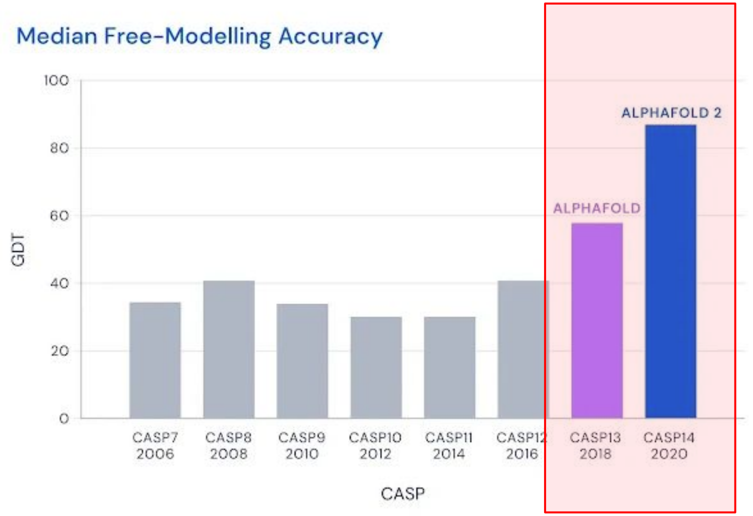


Source: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data

Berkeley UNIVERSITY OF CALIFORNIA

Benchmarks drive progress: the CASP competition

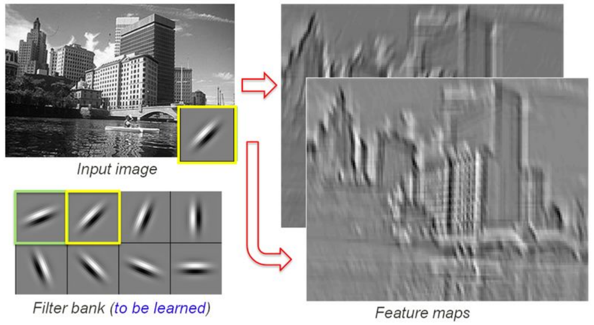
- CASP: Critical Assessment of Structure Prediction (CASP)
  - Biennial competition with structural data for the held out set generated as teams prepare their predictions.
- Traditional approaches typically leverage biophysics for predictions

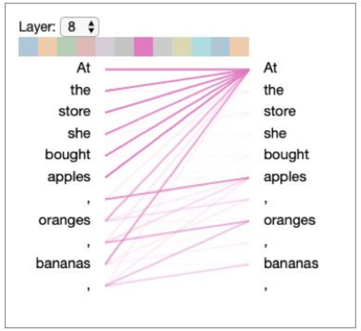


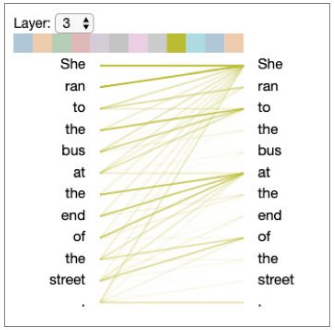
AlphaFold mechanism: Overview

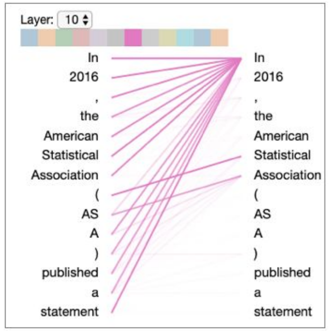
tl;dr: balancing provision of evolutionary and graph-like priors with effective end-to-end learning

Inductive biases in conventional architectures







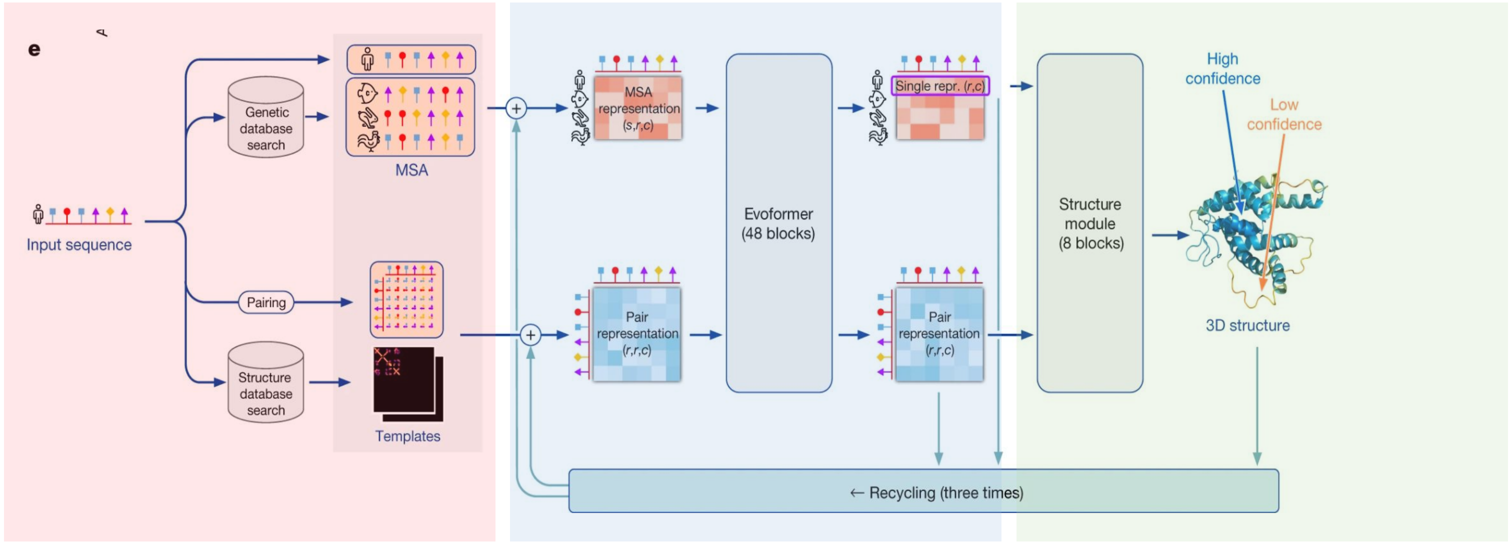


CNNs: adopts ideas from signal processing to learn composite lower level features

Transformers: attention designed to capture local structure (grammar) and global structure (content coherence)

What are the inductive biases to capture for protein structure prediction?

Designing an architecture for biological inductive biases

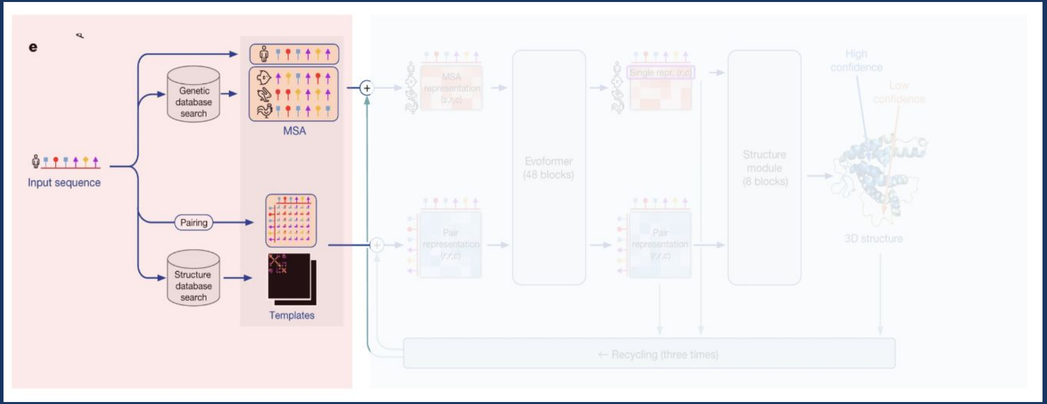

e
Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three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

harness additional sequence-based priors

learn structural features from latents

generate structures

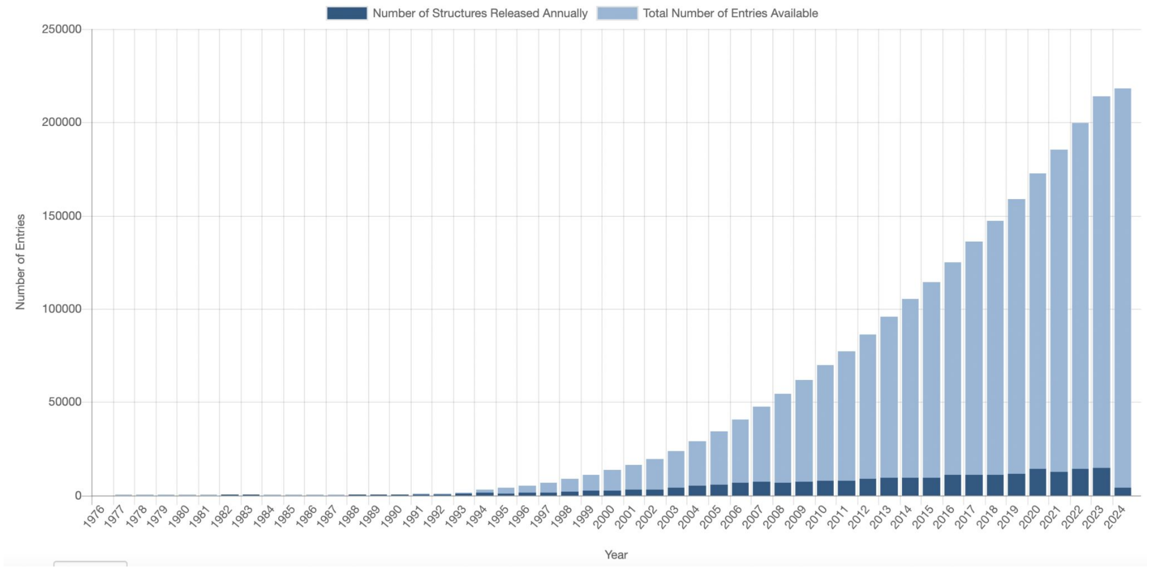
AlphaFold mechanism: Training Data



tl;dr: Labelled structures, unlabelled sequences, and self-distillation

The Protein Data Bank (PDB)

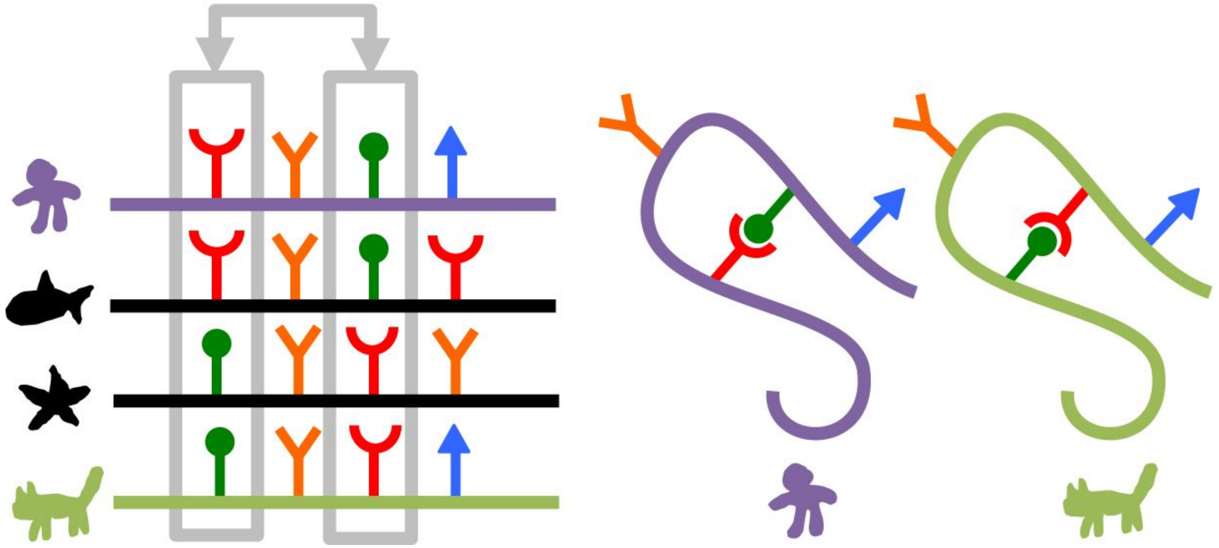
- Publicly available dataset of experimentally-resolved protein structures.



Incorporating labelled and unlabelled data

- Consider pairs of {sequence, structure} data as {input, label}.
  - As of 2024:
    - # of {sequence, structure} pairs in PDB: **218,196**
    - # of sequence-only data in UniRef50: **63,849,054**
  - Relatively small labelled dataset, but a large number of unlabelled sequence data
- Aim – improve structure prediction using unlabelled sequence data:
  - 1. Capture co-evolutionary patterns from sequence
  - 2. Self-distillation – use labelled sequences to predict labels for unlabelled sequences

Coevolution



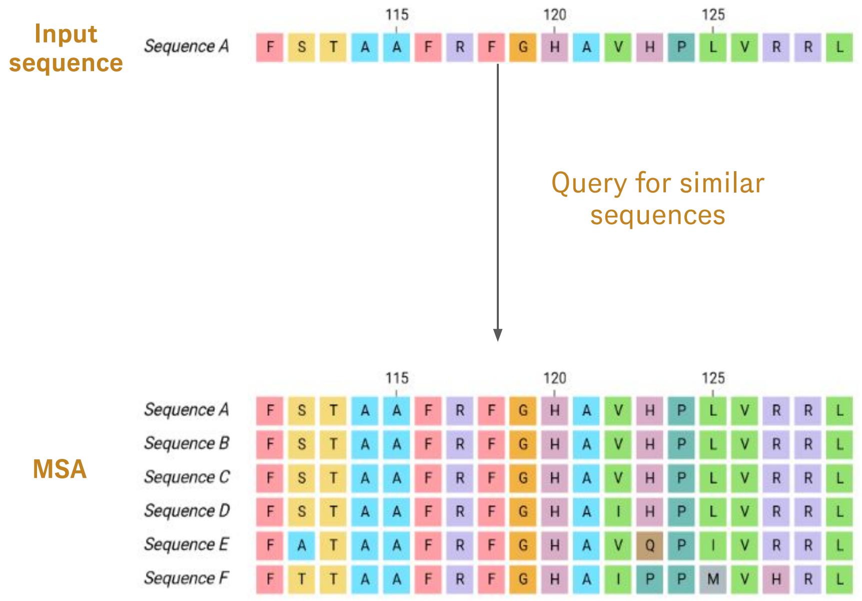
If two positions are in contact, then if one is mutated, the other also loses function.

Thus, statistically observed pairs of residues that "change together" across evolution often denotes contact.

Image credit: Hetu Kamisetty

Multiple sequence alignment (MSA) input

- Multiple sequence alignment:
  - Statistically line up similar string positions across sequences
  - Used to infer evolutionary relationships
- If a statistical sequence pattern is preserved across species, it's likely to imply function
  - Based on principles of Darwinian evolution



					115					120					125				
Sequence A	F	S	T	A	A	F	R	F	G	H	A	V	H	P	L	V	R	R	L

					115					120					125				
Sequence A	F	S	T	A	A	F	R	F	G	H	A	V	H	P	L	V	R	R	L
Sequence B	F	S	T	A	A	F	R	F	G	H	A	V	H	P	L	V	R	R	L
Sequence C	F	S	T	A	A	F	R	F	G	H	A	V	H	P	L	V	R	R	L
Sequence D	F	S	T	A	A	F	R	F	G	H	A	I	H	P	L	V	R	R	L
Sequence E	F	A	T	A	A	F	R	F	G	H	A	V	Q	P	I	V	R	R	L
Sequence F	F	T	T	A	A	F	R	F	G	H	A	I	P	P	M	V	H	R	L

A natural language analogy…

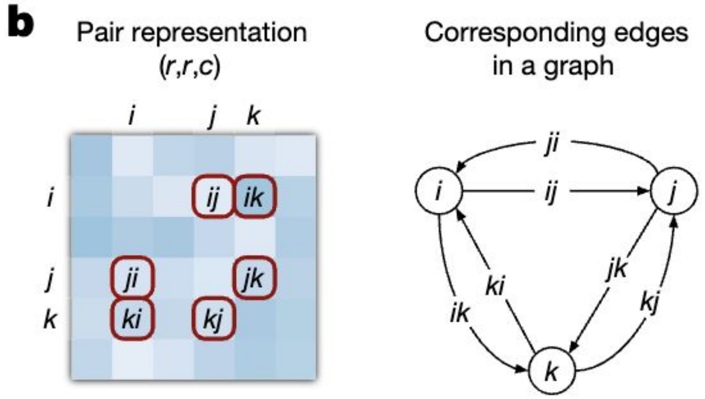
Text 1	Pad Thai is stir-fry dish made with rice noodles, shrimp, chicken (or tofu), peanuts, all sautéed together in a wok and tossed in a delicious sauce.
Text 2	…a delicious Pad Thai recipe. You'll need rice noodles, shrimp, chicken, tofu, peanuts, scrambled eggs, red peppers, and bean sprouts.
Text 3	This delicious thai dish is traditionally made up of rice noodles tossed with a deeply flavored, sweet and sour sauce. The sauce ...
Text 4	Our favorite vegan Pad Thai recipe involves stir-frying rice noodles, tofu, and peanut sauce in a wok, with bean sprouts and vegetables added to taste. You'll love…

A pad thai recipe that doesn't include `rice noodles` would likely be removed by the editor, because it would no longer be pad thai! Thus, `rice noodles` is observed to be conserved across recipes.

Berkeley
UNIVERSITY OF CALIFORNIA

Template Input

- For all sequence homologs found during MSA construction: if an existing structure exists in the PDB, use it as input.
  - In practice, few structures have available templates at inference

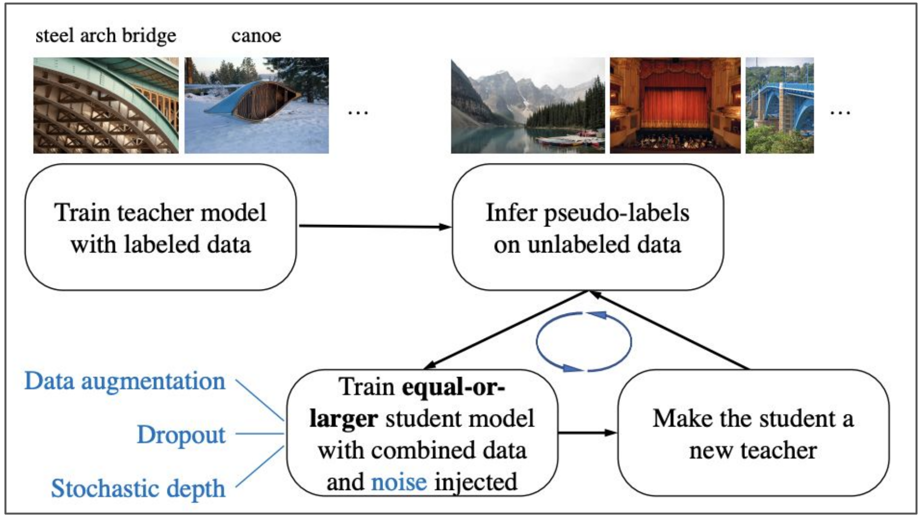


Noisy Student Training

[Submitted on 11 Nov 2019 (v1), last revised 19 Jun 2020 (this version, v4)]

Self-training with Noisy Student improves ImageNet classification

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, Quoc V. Le

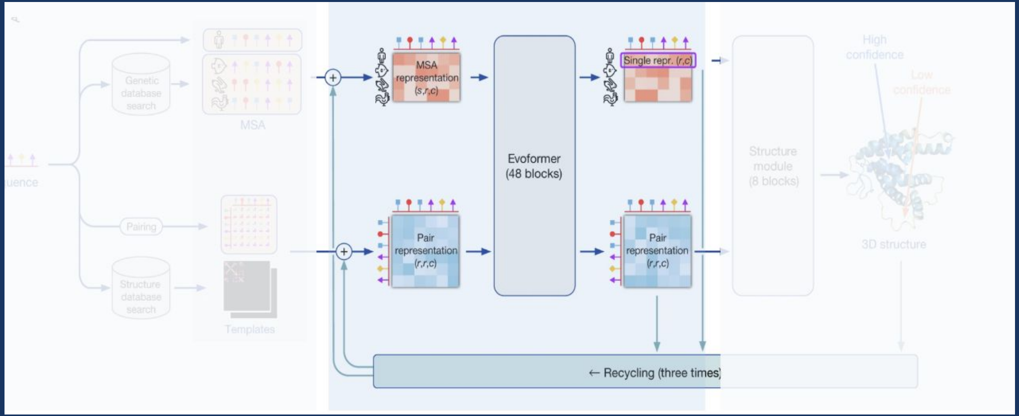

steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Data augmentation
Dropout
Stochastic depth
Train equal-or-larger student model with combined data and noise injected
Make the student a new teacher

1. Train "undistilled" model on PDB dataset only
2. Predict structure for all available sequences
3. Retrain network with distillation dataset

During student training:

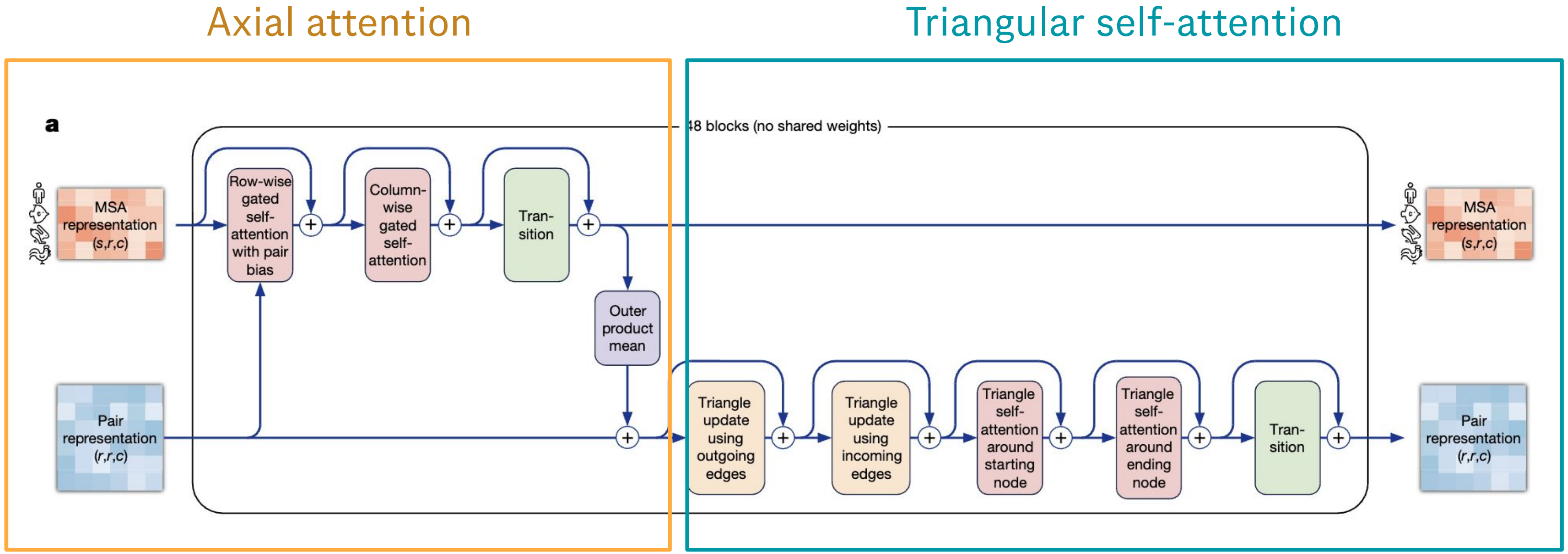
- 75% from self-distillation dataset
- 25% from PDB

AlphaFold mechanism: Evoformer


uence
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)

tl;dr: Criss-cross attention to evolutionary details + triangulate attention for 3D awareness

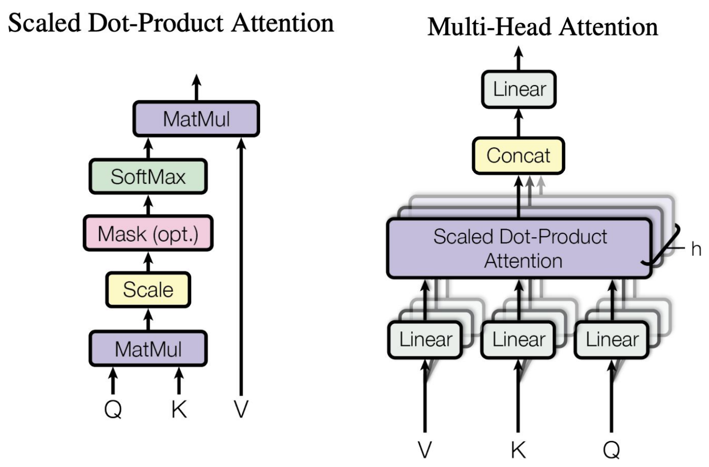
Breaking down the Evoformer



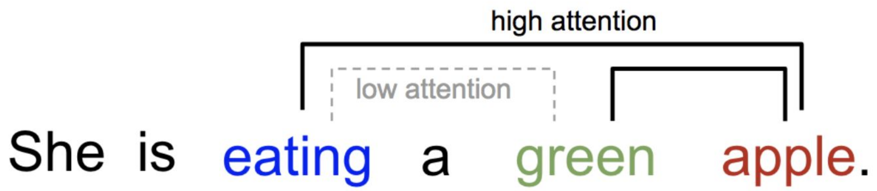
Axial attention
Triangular self-attention
a
48 blocks (no shared weights)
MSA representation (s,r,c)
Pair representation (r,r,c)
Row-wise gated self-attention with pair bias
Column-wise gated self-attention
Tran-sition
Outer product mean
Triangle update using outgoing edges
Triangle update using incoming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

48 stacked blocks

Review: transformers and self-attention



$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention weights learn relationships between sequence positions

Axial attention: encoding evolution

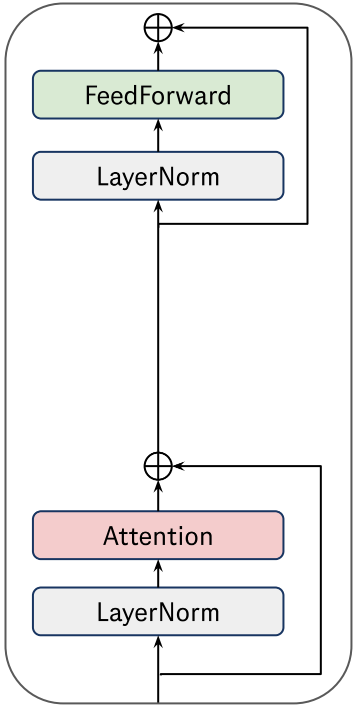
[Submitted on 28 Nov 2018 (v1), last revised 9 Jul 2020 (this version, v2)]

CCNet: Criss-Cross Attention for Semantic Segmentation

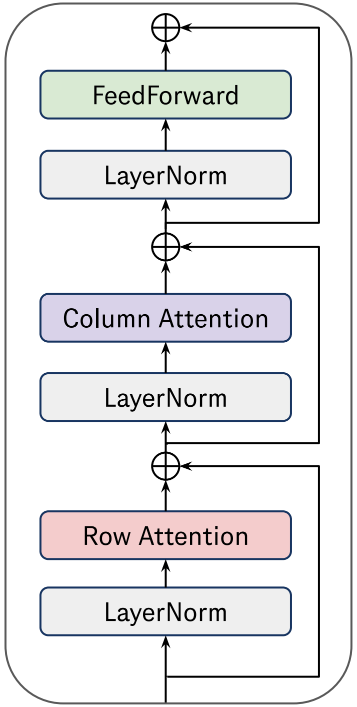
Zilong Huang, Xinggang Wang, Yunchao Wei, Lichao Huang, Humphrey Shi, Wenyu Liu, Thomas S. Huang

MSA Transformer

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

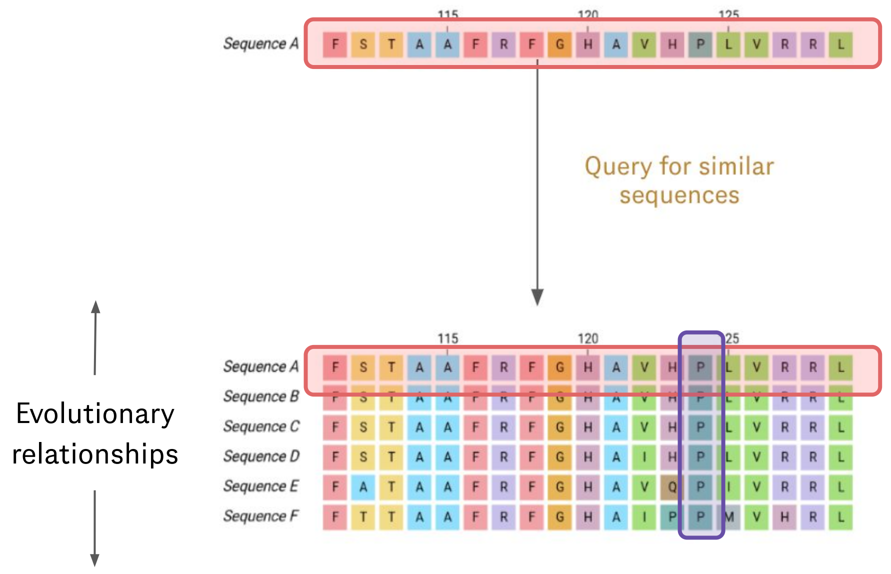


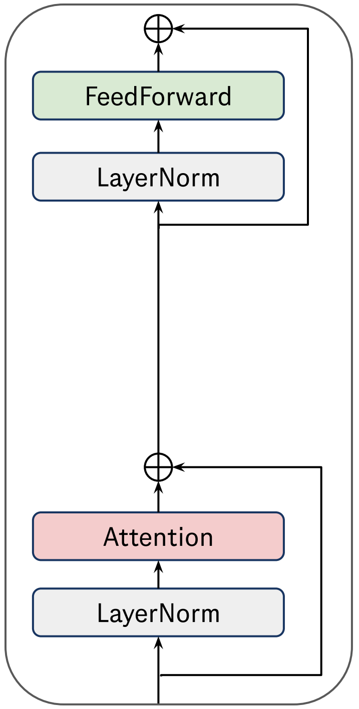
Original attention block



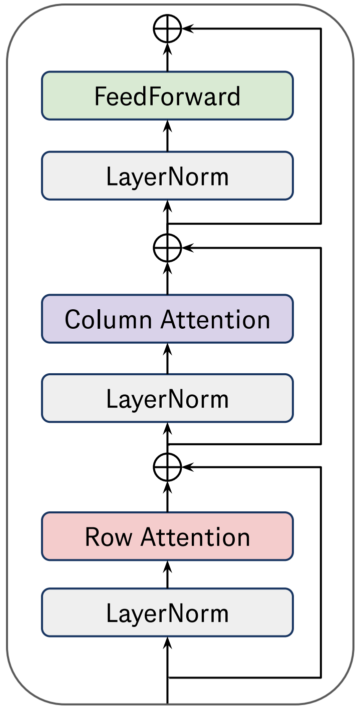
Axial transformer block

Axial attention: encoding evolution





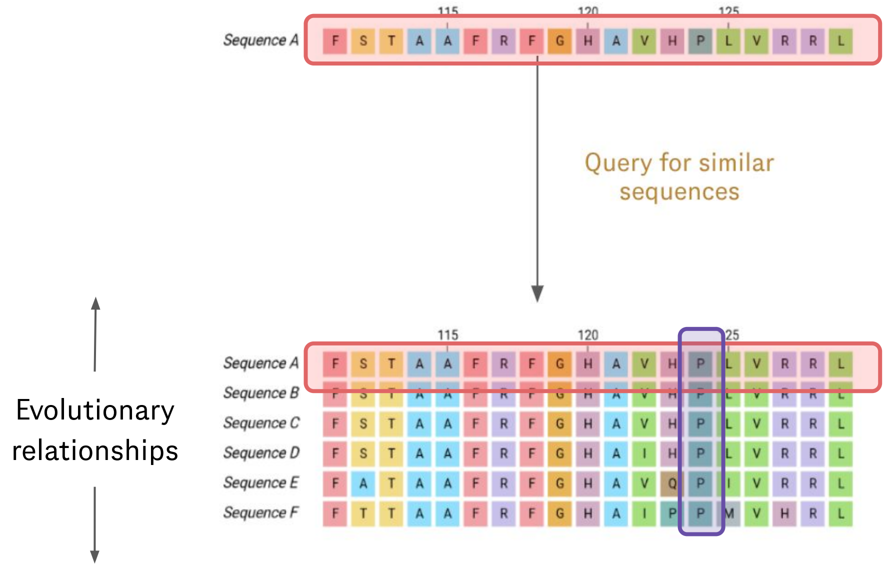
Original attention block



Axial transformer block

Berkeley UNIVERSITY OF CALIFORNIA

Axial attention: encoding evolution

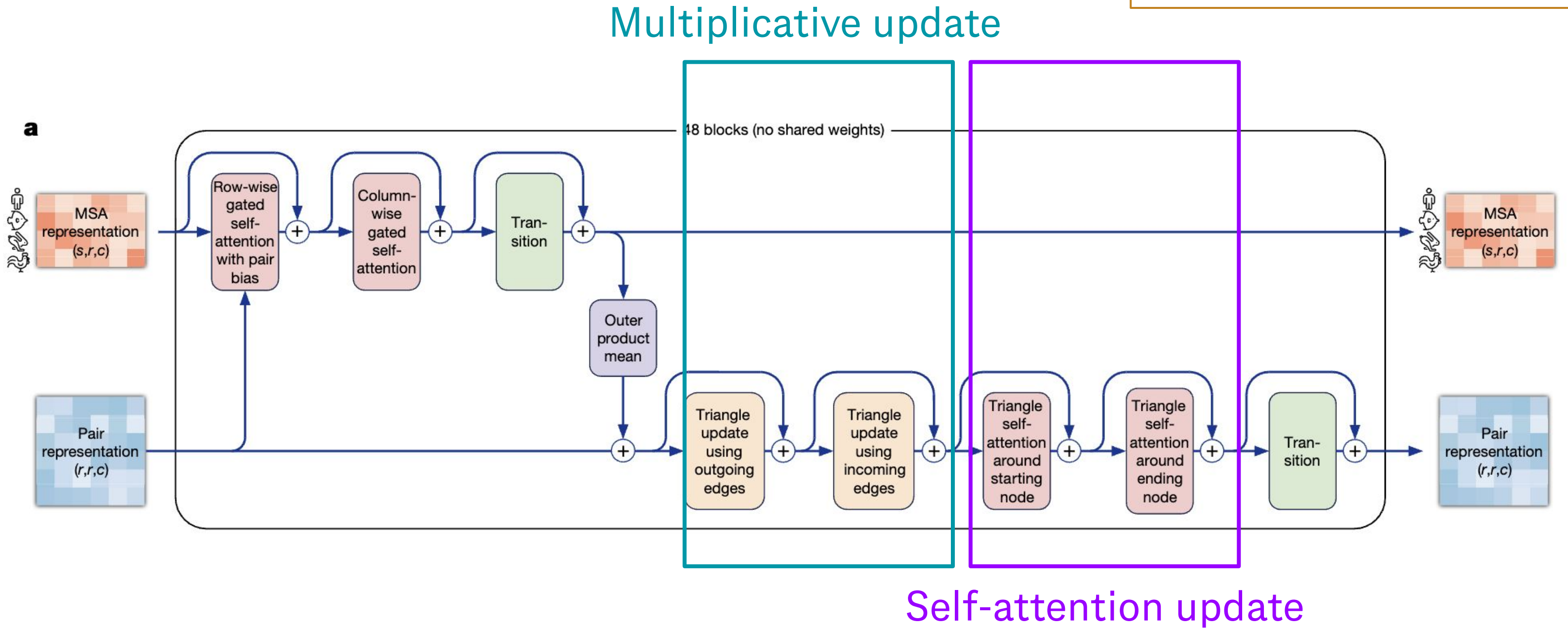


115
120
125
Sequence A
F S T A A F R F G H A V H P L V R R L
Query for similar sequences
Evolutionary relationships
Sequence A
Sequence B
Sequence C
Sequence D
Sequence E
Sequence F
F S T A A F R F G H A V H P L V R R L
F S T A A F R F G H A V H P L V R R L
F S T A A F R F G H A V H P L V R R L
F S T A A F R F G H A I H P L V R R L
F A T A A F R F G H A V Q P I V R R L
F T T A A F R F G H A I P P M V H R L

- **Row-wise attention** captures intra-sequence relationships
- **Column-wise attention** captures evolutionary relationships

Triangular self-attention

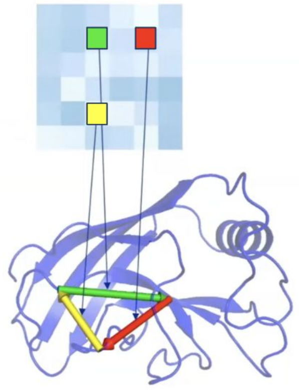
Aim: capture 3D constraints in learned embeddings.


Multiplicative update
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)
Self-attention update

Berkeley UNIVERSITY OF CALIFORNIA

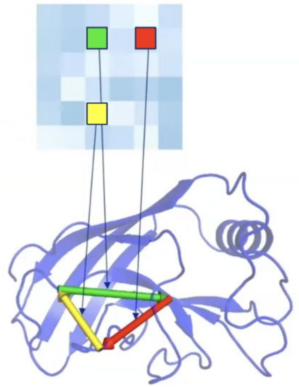
Triangular self-attention: encoding graph structure

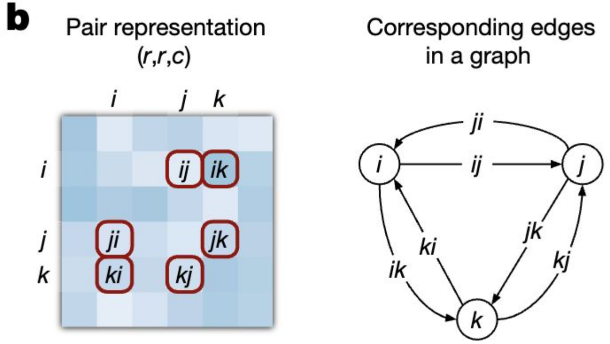
- Pairwise representation captures relationships between <u>residue positions</u>
  - I.e. captures positions in close contact



Triangular self-attention: encoding graph structure

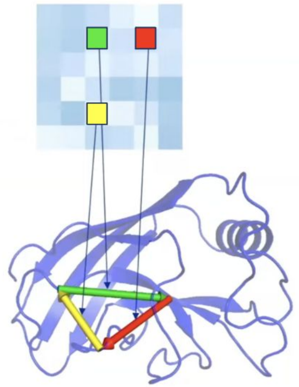
- Pairwise representation captures relationships between <u>residue positions</u>
  - I.e. captures positions in close contact
- Graph structures represent these edge relationships well
  - Aim: update transformer learning to implicitly capture graph relationships

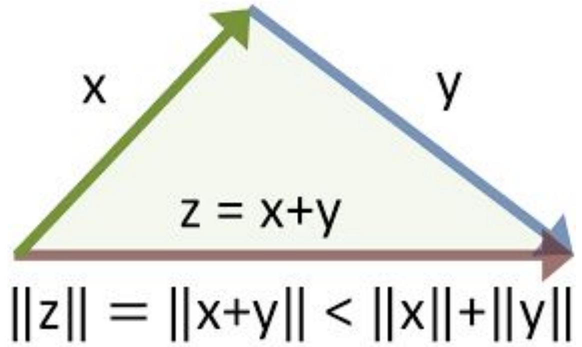




Triangular self-attention: enforcing 3D common sense

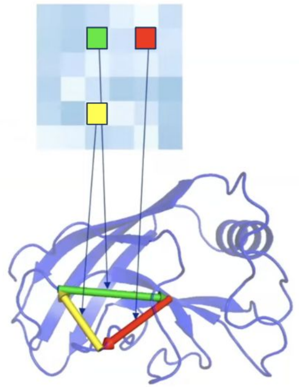
- If pairwise edges are to represent distances in 3D space, they should obey the triangle inequality
  - Note: the inequality refers to distances between coordinates rather than the 3D coordinate positions

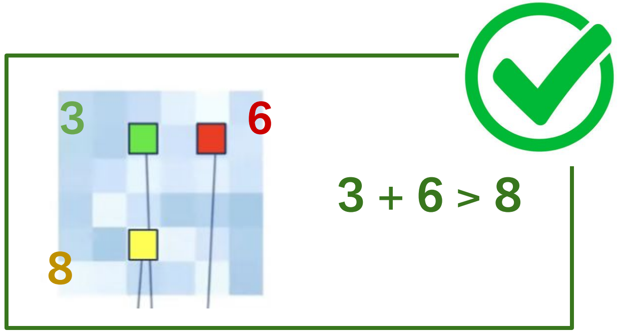


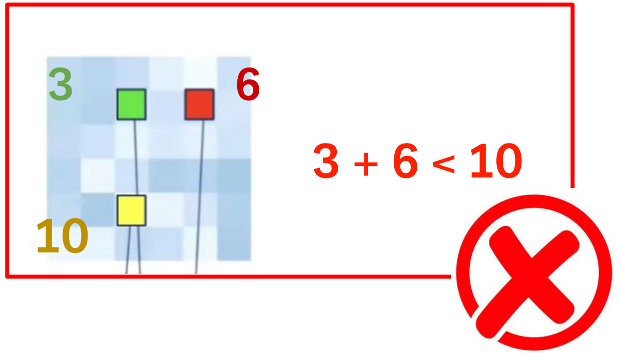


Triangular self-attention: enforcing 3D common sense

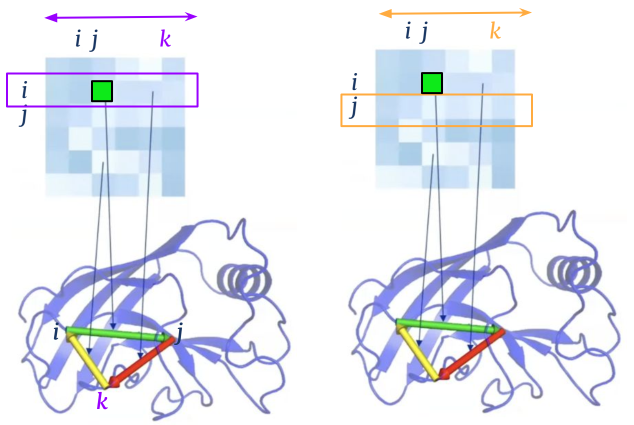
- If pairwise edges are to represent distances in 3D space, they should obey the triangle inequality
- Output values by the naive attention mechanism may violate this
  - need to implement a soft constraint



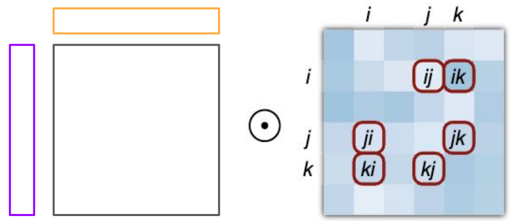




Triangular self-attention: multiplicative update



- To update pairwise representation at {i, j}, take the outer product of {i, k} and {j, k} for all k, and update with new outer product



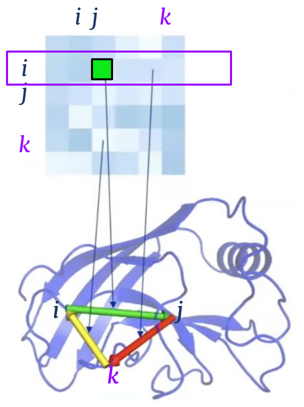
$$\mathbf{g}_{ij} = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$$

$$\tilde{\mathbf{z}}_{ij} = \mathbf{g}_{ij} \odot \text{Linear}(\text{LayerNorm}(\textstyle\sum_k \mathbf{a}_{ik} \odot \mathbf{b}_{jk}))$$

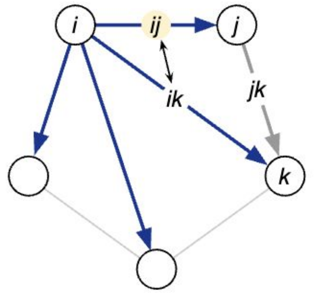
Berkeley
UNIVERSITY OF CALIFORNIA

Triangular self-attention: attention update

- To update attention value of pairwise representation at {i, j} ("starting node"), update by the attention value at {i,k} for all k.
  - Additionally: bias the query-key multiplication {i, j} by value at {i, k}

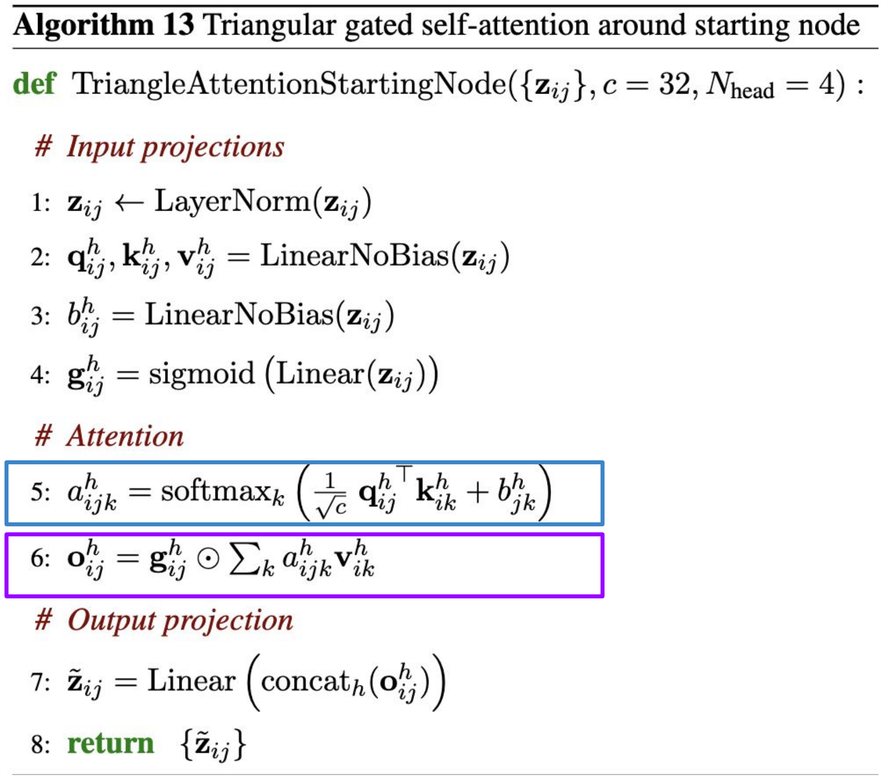


Triangle self-attention around starting node



Triangular self-attention: attention update

- To update attention value of pairwise representation at {i, j} ("starting node"), update by the attention value at {i,k} for all k.
    - Additionally: bias the query-key multiplication {i, j} by value at {i, k}

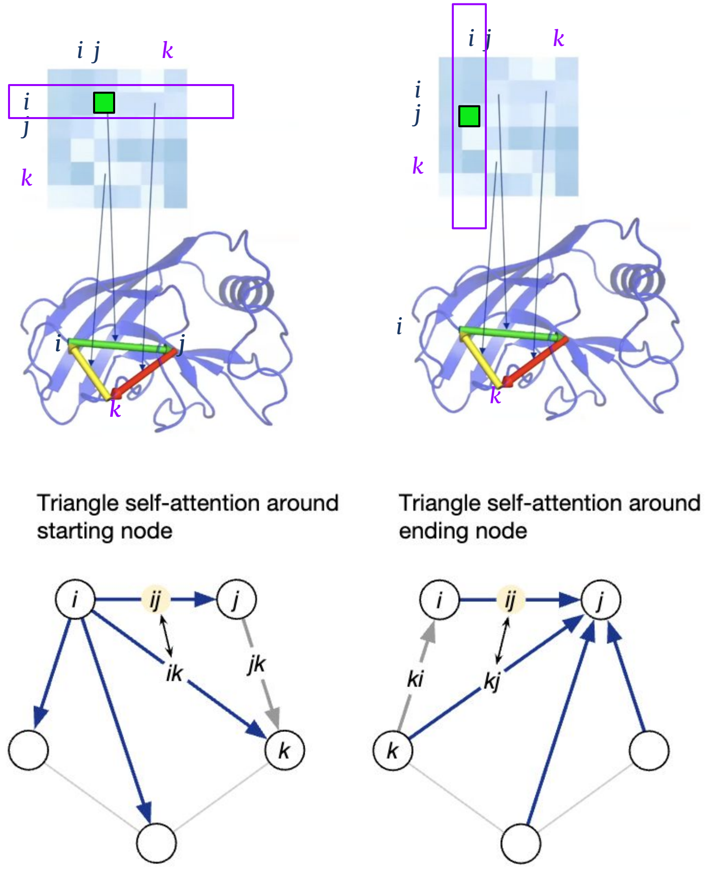


Algorithm 13 Triangular gated self-attention around starting node

```
def TriangleAttentionStartingNode($\{\mathbf{z}_{ij}\}, c = 32, N_{\text{head}} = 4$) :
  # Input projections
  1: $\mathbf{z}_{ij} \leftarrow \text{LayerNorm}(\mathbf{z}_{ij})$
  2: $\mathbf{q}_{ij}^h, \mathbf{k}_{ij}^h, \mathbf{v}_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$
  3: $b_{ij}^h = \text{LinearNoBias}(\mathbf{z}_{ij})$
  4: $\mathbf{g}_{ij}^h = \text{sigmoid}\left(\text{Linear}(\mathbf{z}_{ij})\right)$
  # Attention
  5: $a_{ijk}^h = \text{softmax}_k \left( \frac{1}{\sqrt{c}} \mathbf{q}_{ij}^{h\top} \mathbf{k}_{ik}^h + b_{jk}^h \right)$
  6: $\mathbf{o}_{ij}^h = \mathbf{g}_{ij}^h \odot \sum_k a_{ijk}^h \mathbf{v}_{ik}^h$
  # Output projection
  7: $\tilde{\mathbf{z}}_{ij} = \text{Linear}\left(\text{concat}_h(\mathbf{o}_{ij}^h)\right)$
  8: return $\{\tilde{\mathbf{z}}_{ij}\}$
```

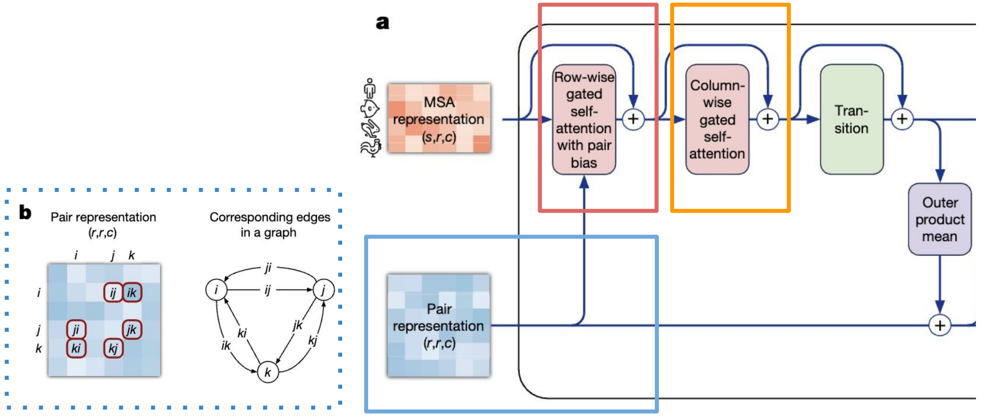
Triangular self-attention: attention update

- Repeat for "ending nodes": *j*
  - i.e. having just updated attention value at {i, j} by all {i, k} ∀ k, we now update attention value at **{j, i}** by attention values at **{j, k}** ∀ k.


i j k
i
j
k
i
j
k
i j k
i
j
k
i
k
Triangle self-attention around starting node
Triangle self-attention around ending node
i
ij
j
ik
jk
k
i
ij
j
ki
kj
k

Berkeley UNIVERSITY OF CALIFORNIA

Integrating graph and evolutionary representations


a
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)
b
Pair representation (r,r,c)
Corresponding edges in a graph
i
j
k
ij
ik
ji
jk
ki
kj

Bias from pairwise representations added to the corresponding {i, j} position in row attention, followed by outer product mean

Row attention with bias

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

Column attention

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

Outer product mean

$$\mathbf{o}_{ij} = \boxed{\mathrm{flatten}\left(\mathrm{mean}_s(\mathbf{a}_{si} \otimes \mathbf{b}_{sj})\right)}$$

Berkeley UNIVERSITY OF CALIFORNIA

Evoformer: Summary

Axial attention

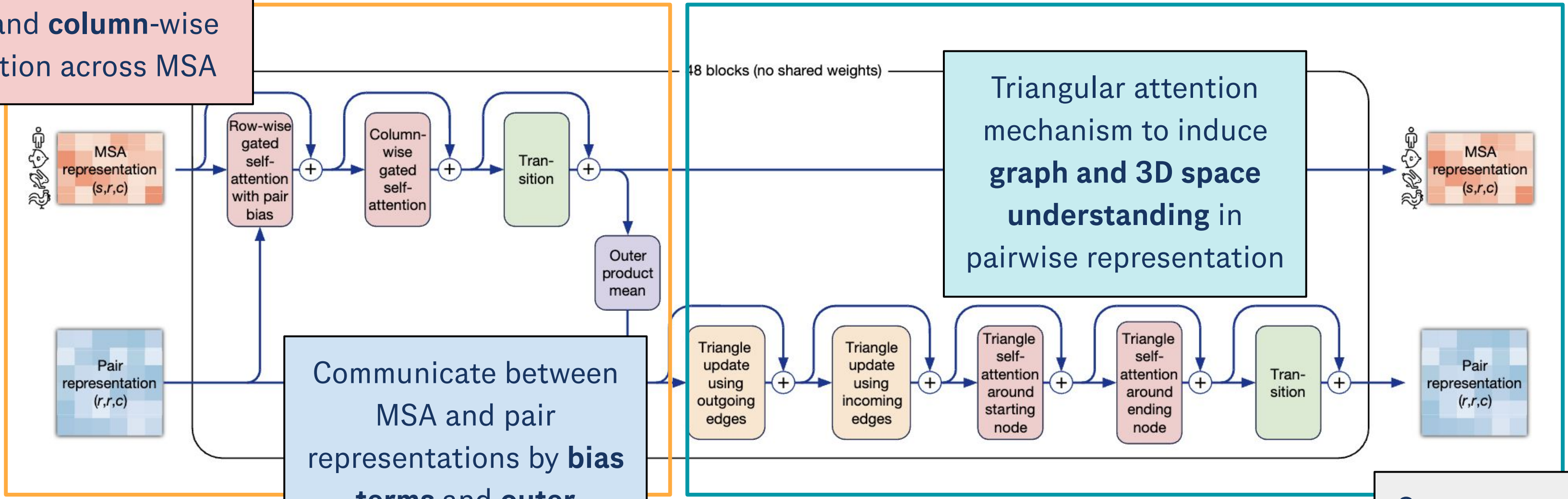
Triangular self-attention

Row and **column**-wise attention across MSA

Triangular attention mechanism to induce **graph and 3D space understanding** in pairwise representation

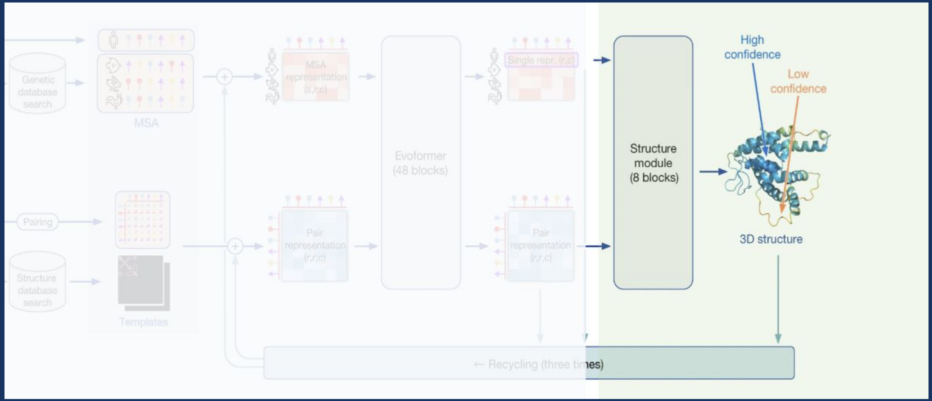
Communicate between MSA and pair representations by **bias terms** and **outer product means**

Outputs are fed into the next block


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)

AlphaFold mechanism:

Structure Module

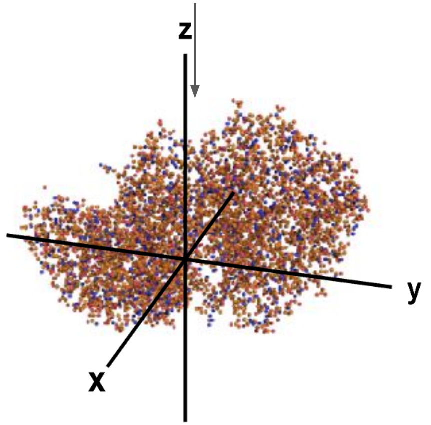


Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Tl;dr: simplify prediction by considering protein structures as rigid groups connected by "joints".

How should we represent the 3D structure?

Idea 1: As a point cloud of 3D coordinates



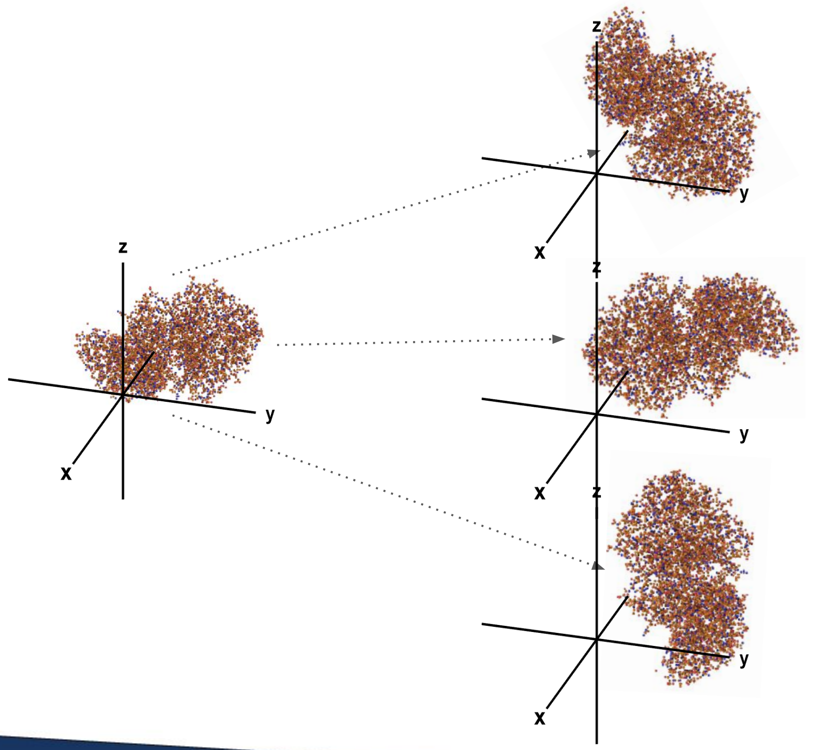
Berkeley
UNIVERSITY OF CALIFORNIA

How should we represent the 3D structure?

Idea 1: As a point cloud of 3D coordinates
→ issue: lacks 3D "equivariance"

Same object in different orientations have different representations.

We want same object in different orientations to the same representation



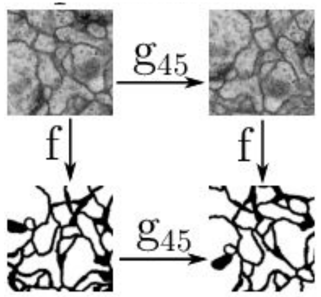
How should we represent the 3D structure?

Idea 1: As a point cloud of 3D coordinates
→ issue: lacks 3D "equivariance"

Same object in different orientations have different representations.

We want same object in different orientations to the same representation

For any rotations applied to the input, we want our model to apply that same rotation to the output:

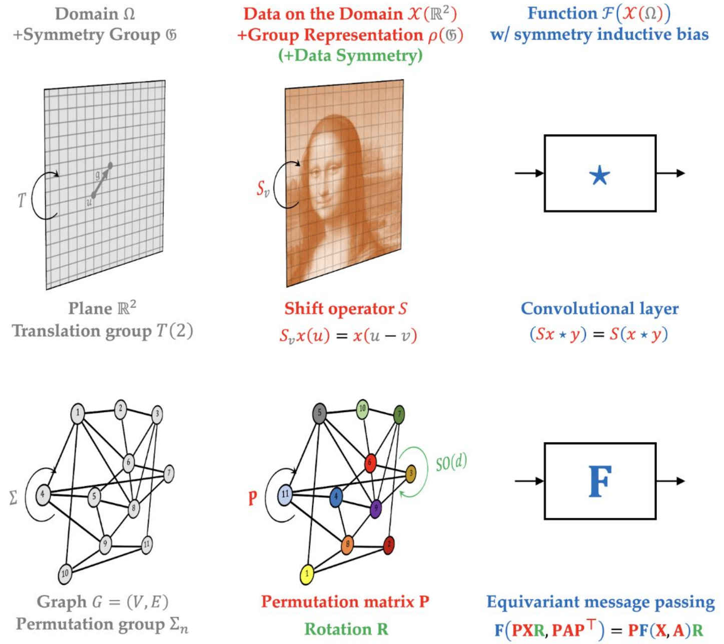


g: rotation
f: function (e.g. neural net)

How should we represent the 3D structure?

Idea 2: Use equivariant neural network architectures from geometric deep learning.

→ Details are outside the scope of this lecture, but such methods are often complex and computationally expensive.



Domain Ω
+Symmetry Group $\mathfrak{G}$
Data on the Domain $\mathcal{X}(\mathbb{R}^2)$
+Group Representation $\rho(\mathfrak{G})$
(+Data Symmetry)
Function $\mathcal{F}(\mathcal{X}(\Omega))$
w/ symmetry inductive bias
T
u
g
S_v
$\star$
Plane $\mathbb{R}^2$
Translation group $T(2)$
Shift operator S
$S_v x(u) = x(u-v)$
Convolutional layer
$(Sx \star y) = S(x \star y)$
Σ
P
$SO(d)$
F
Graph $G = (V, E)$
Permutation group Σ_n
Permutation matrix P
Rotation R
Equivariant message passing
$\mathbf{F}(\mathbf{PXR}, \mathbf{PAP}^\top) = \mathbf{PF}(\mathbf{X}, \mathbf{A})\mathbf{R}$

Source:
https://thegradient.pub/towards-geometric-deep-learning/

How should we represent the 3D structure?

Idea 3: as “rigid blobs” with translation and rotation matrices from a reference frame

→ functions built on this representation is equivariant by construction
→ easily adaptable as input to transformer

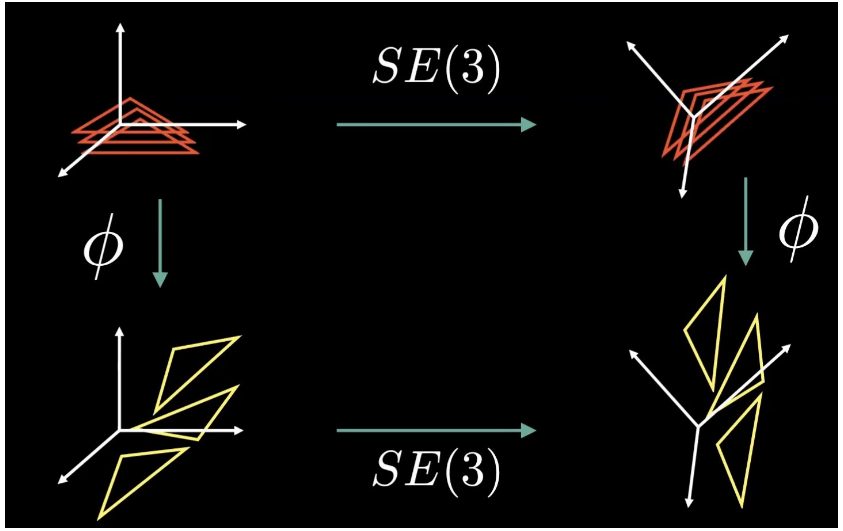


Figure by: Nazim Bouatta

AlphaFold representation: “Residue gas”

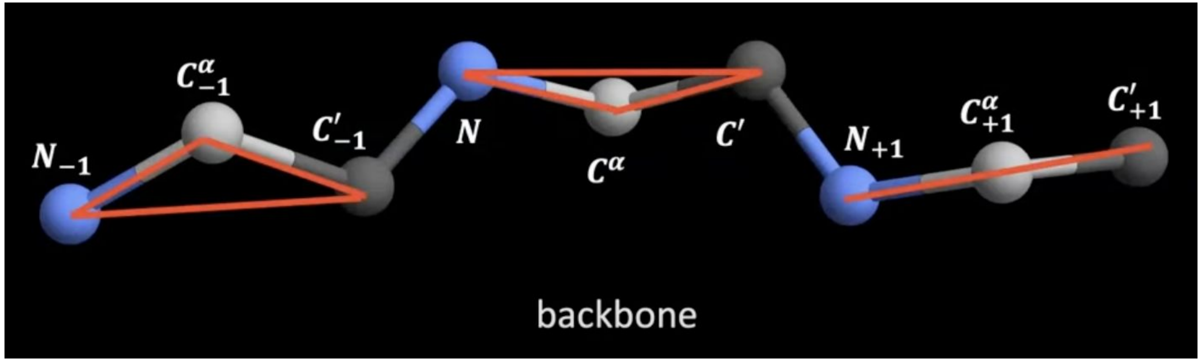


Figure by: Nazim Bouatta

- Represent each residue as a triangle with {N, Cα, C} as vertices.
- Backbone is a series of tiled triangles in 3D.

Residue gas representation

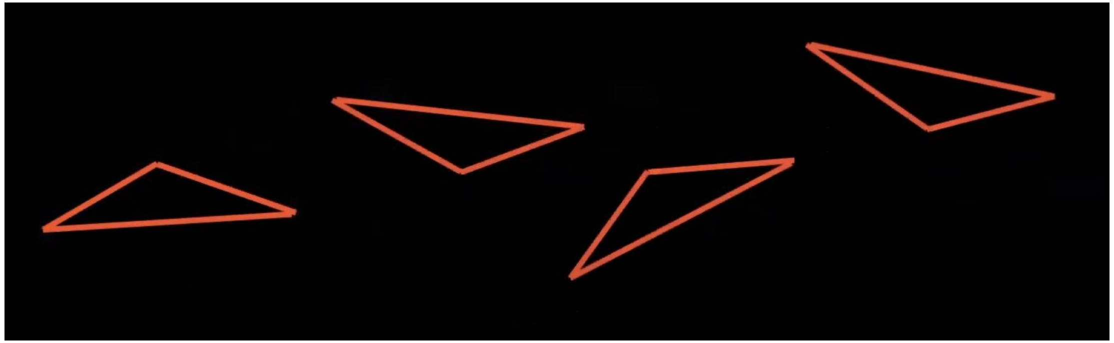


Figure by: Nazim Bouatta

- “Chunk up” the structure, and consider each residue as “independently floating”.
    - Network learns to place the backbone frames sequentially.
- All frames are initialized at the origin
    - Termed “black hole initialization”

Berkeley
UNIVERSITY OF CALIFORNIA

Representing residue positions as transformations

- Each residue is represented as a rigid frame transformed by a tuple of **rotation** (R_i) and **translation** (t_i) matrices from the origin:
  - Rotation: $R_i \in \mathbb{R}^{3x3}$
  - Translation: $T_i \in \mathbb{R}^3$
- Obtained from 3D coordinates via the Gram-Schmidt process

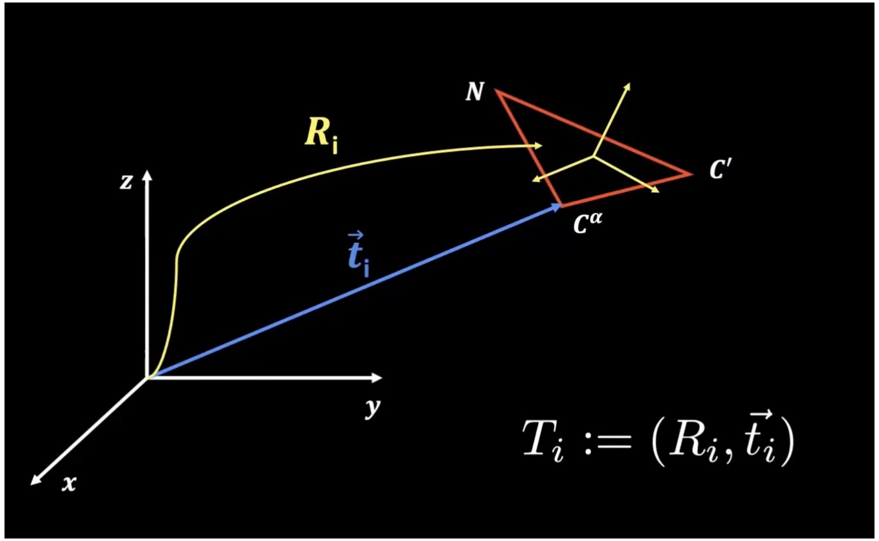
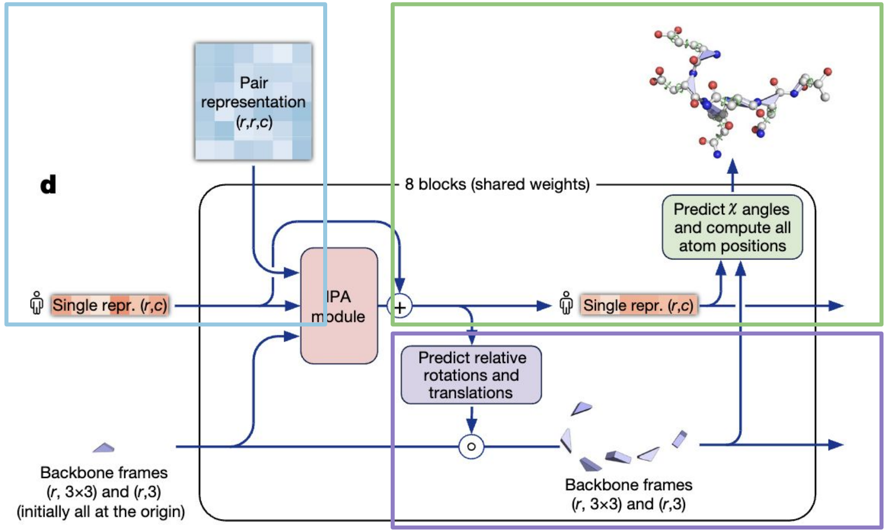


Figure by: Nazim Bouatta

Structure module: overview


Pair representation (r,r,c)
d
8 blocks (shared weights)
Predict χ angles and compute all atom positions
Single repr. (r,c)
IPA module
+
Single repr. (r,c)
Predict relative rotations and translations
∘
Backbone frames (r, 3×3) and (r,3) (initially all at the origin)
Backbone frames (r, 3×3) and (r,3)

Sequence & pair inputs from evoformer

Resolve side chain positions from predicted torsion angles

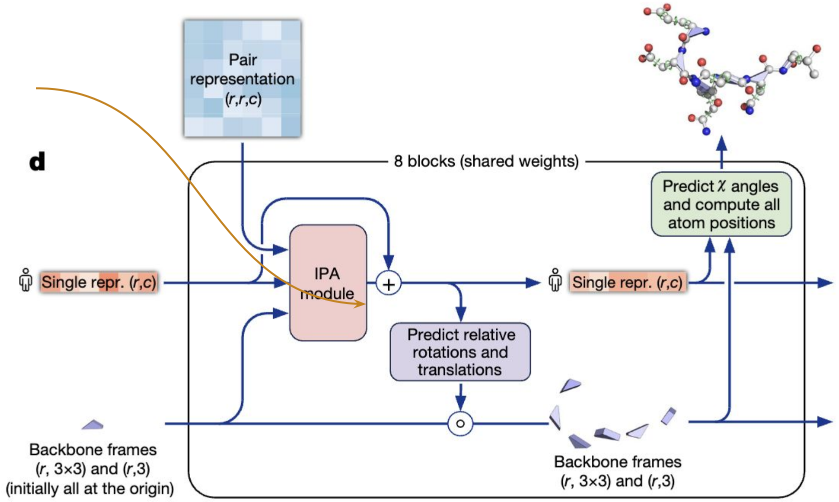
Backbone positions

Berkeley UNIVERSITY OF CALIFORNIA

Invariant Point Attention

Workhouse of the structure module: **Invariant Point Attention (IPA).**

→ A modified attention mechanism to act on $T_i = (R_i, t_i)$ tuples.



d
Pair representation (r,r,c)
8 blocks (shared weights)
Predict χ angles and compute all atom positions
Single repr. (r,c)
IPA module
+
Single repr. (r,c)
Predict relative rotations and translations
∘
Backbone frames (r, 3×3) and (r,3) (initially all at the origin)
Backbone frames (r, 3×3) and (r,3)

Invariant Point Attention

From Evoformer

$$a_{ij}^h = \text{softmax}_j \left(w_L \left(\frac{1}{\sqrt{c}} \mathbf{q}_i^{h\top} \mathbf{k}_j^h + b_{ij}^h \right.\right.$$

Query/key over single representation

Bias from pairwise representation

Invariant Point Attention

From Evoformer

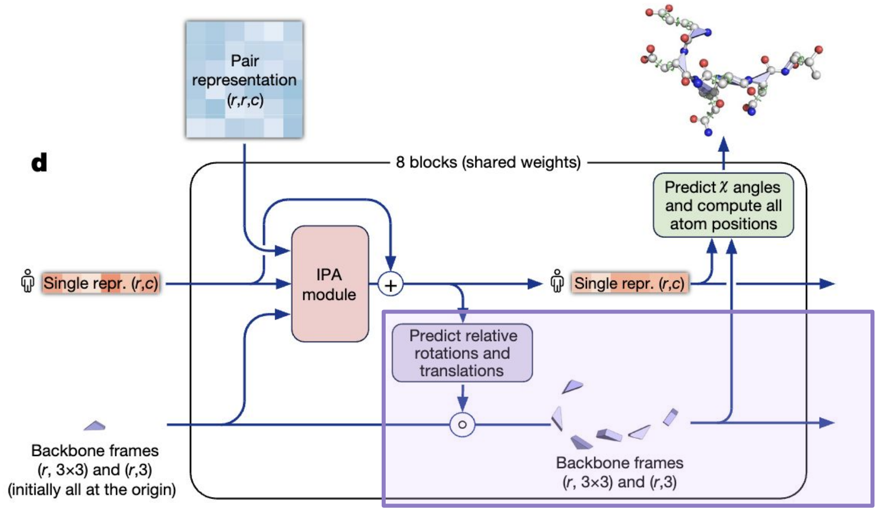
Bias by attending to frame transformations for residues *i* and *j*

$$a_{ij}^{h} = \mathrm{softmax}_j \left(w_L \left(\underbrace{\frac{1}{\sqrt{c}} \mathbf{q}_i^{h\top} \mathbf{k}_j^h}_{\text{Query/key over single representation}} + \underbrace{b_{ij}^h}_{\text{Bias from pairwise representation}} \right. - \frac{\gamma^h w_C}{2} \sum_p \left\| T_i \circ \vec{\mathbf{q}}_i^{hp} - T_j \circ \vec{\mathbf{k}}_j^{hp} \right\|^2 \right) \Bigg)$$

Note: L2-norm between Euclidean matrices is invariant by construction, since the distance between matrices remains constant if the same rotation is applied to both!

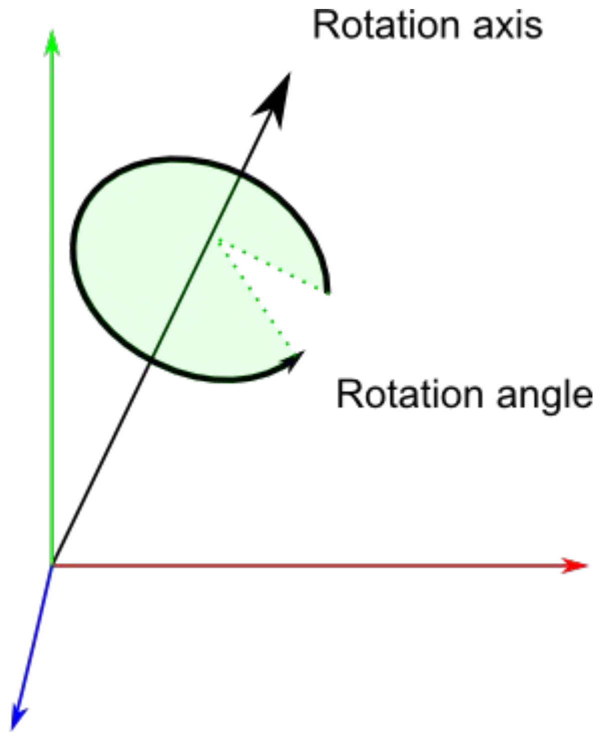
Backbone update

- Predict backbone frames from IPA module output

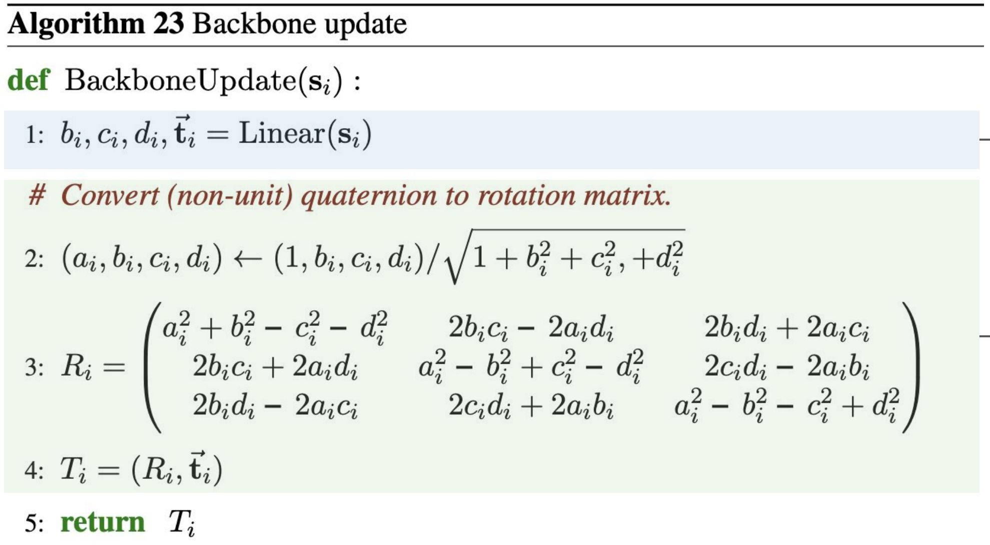

d
Pair representation (r,r,c)
8 blocks (shared weights)
Single repr. (r,c)
IPA module
+
Predict relative rotations and translations
Single repr. (r,c)
Predict χ angles and compute all atom positions
Backbone frames (r, 3×3) and (r,3) (initially all at the origin)
Backbone frames (r, 3×3) and (r,3)

Backbone update

- Rotation matrices are stored as quaternions
  - Rather than as a $\mathbb{R}^{3x3}$ matrix, use a $\mathbb{R}^4$ to denote **rotation axis vector and associated angle**
    - Only two dimensions need to be provided for the axis vector to define a unique rotation
  - Reduces memory usage
  - First component is fixed to 1, and network predicts the remaining 3 components.



Backbone update



Algorithm 23 Backbone update

```
def BackboneUpdate($\mathbf{s}_i$) :
  1: $b_i, c_i, d_i, \vec{\mathbf{t}}_i = \text{Linear}(\mathbf{s}_i)$
  # 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: $R_i = \begin{pmatrix} a_i^2 + b_i^2 - c_i^2 - d_i^2 & 2b_ic_i - 2a_id_i & 2b_id_i + 2a_ic_i \\ 2b_ic_i + 2a_id_i & a_i^2 - b_i^2 + c_i^2 - d_i^2 & 2c_id_i - 2a_ib_i \\ 2b_id_i - 2a_ic_i & 2c_id_i + 2a_ib_i & a_i^2 - b_i^2 - c_i^2 + d_i^2 \end{pmatrix}$
  4: $T_i = (R_i, \vec{\mathbf{t}}_i)$
  5: return $T_i$
```

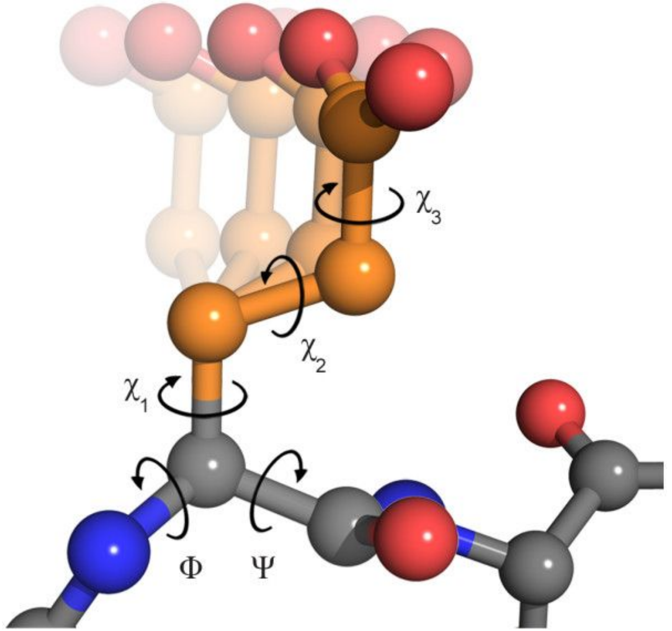
Predict rotation (quaternion components) and translation vector

Convert back to $\mathbb{R}^{3x3}$ representation

Berkeley UNIVERSITY OF CALIFORNIA

Placing the side chain atoms

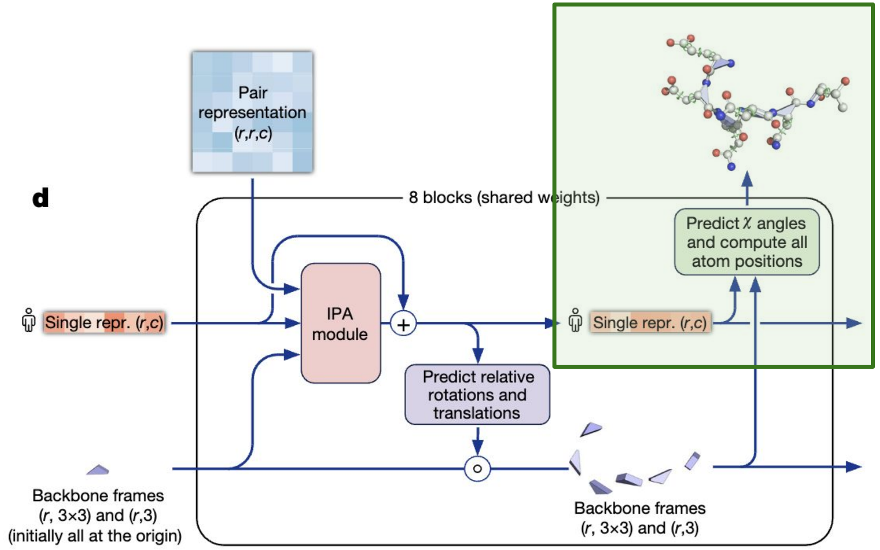
- Predicting frame transformations only places the backbone atoms
- To place the side chains, we also predict the torsion angles.
    - Consider proteins as having “moveable joints”, while other components are kept rigid.



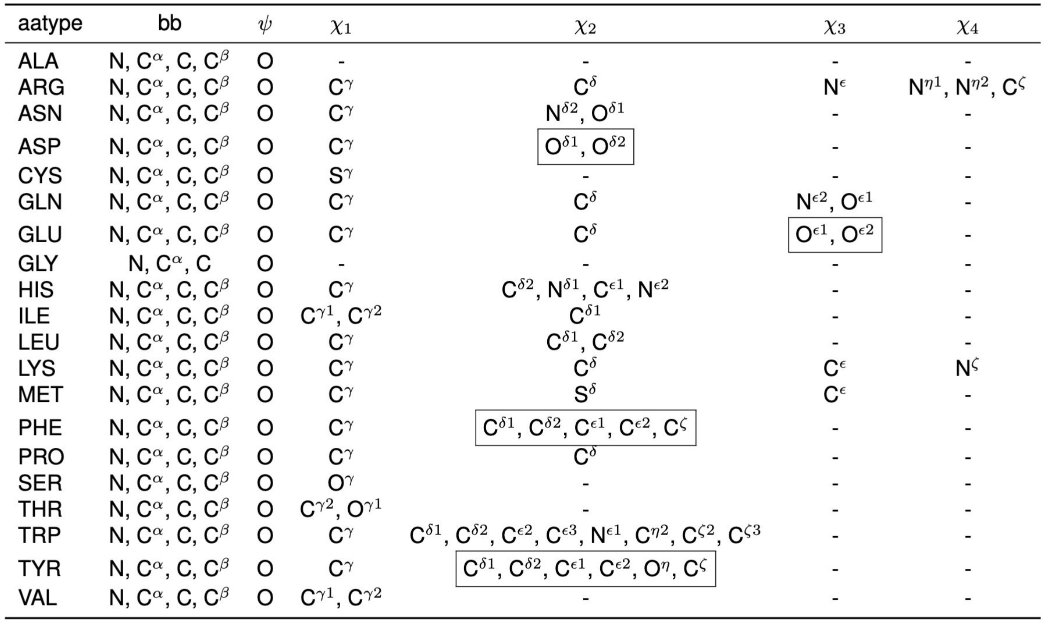
Berkeley UNIVERSITY OF CALIFORNIA

Predicting torsion angles

- Predict torsion angles $\alpha \in \mathbb{R}^2$ with a shallow ResNet
- 7 total types of rotations:
  - **Backbone: {ω,φ,ψ}**
  - **Sidechain: $\{\chi_1,\chi_2,\chi_3,\chi_4\}$**


d
Pair representation (r,r,c)
8 blocks (shared weights)
Single repr. (r,c)
IPA module
Predict relative rotations and translations
Predict χ angles and compute all atom positions
Single repr. (r,c)
Backbone frames (r, 3×3) and (r,3) (initially all at the origin)
Backbone frames (r, 3×3) and (r,3)

Predicting torsion angles



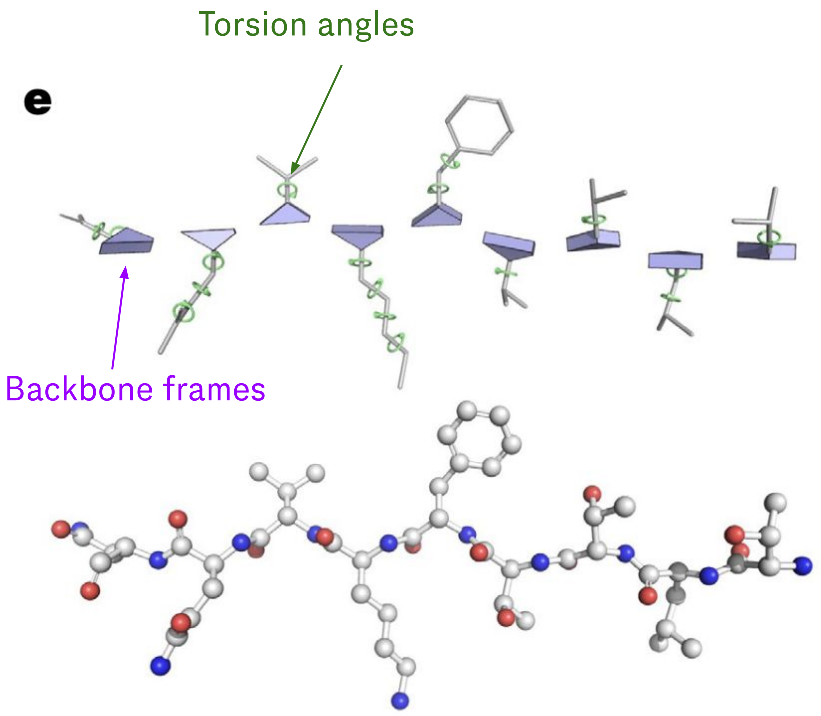
aatype	bb	ψ	χ_1	χ_2	χ_3	χ_4
ALA	N, C^{α}, C, C^{β}	O	-	-	-	-
ARG	N, C^{α}, C, C^{β}	O	C^{γ}	C^{δ}	N^{ϵ}	$N^{\eta 1}$, $N^{\eta 2}$, C^{ζ}
ASN	N, C^{α}, C, C^{β}	O	C^{γ}	$N^{\delta 2}$, $O^{\delta 1}$	-	-
ASP	N, C^{α}, C, C^{β}	O	C^{γ}	$\boxed{O^{\delta 1}, O^{\delta 2}}$	-	-
CYS	N, C^{α}, C, C^{β}	O	S^{γ}	-	-	-
GLN	N, C^{α}, C, C^{β}	O	C^{γ}	C^{δ}	$N^{\epsilon 2}$, $O^{\epsilon 1}$	-
GLU	N, C^{α}, C, C^{β}	O	C^{γ}	C^{δ}	$\boxed{O^{\epsilon 1}, O^{\epsilon 2}}$	-
GLY	N, C^{α}, C	O	-	-	-	-
HIS	N, C^{α}, C, C^{β}	O	C^{γ}	$C^{\delta 2}$, $N^{\delta 1}$, $C^{\epsilon 1}$, $N^{\epsilon 2}$	-	-
ILE	N, C^{α}, C, C^{β}	O	$C^{\gamma 1}$, $C^{\gamma 2}$	$C^{\delta 1}$	-	-
LEU	N, C^{α}, C, C^{β}	O	C^{γ}	$C^{\delta 1}$, $C^{\delta 2}$	-	-
LYS	N, C^{α}, C, C^{β}	O	C^{γ}	C^{δ}	C^{ϵ}	N^{ζ}
MET	N, C^{α}, C, C^{β}	O	C^{γ}	S^{δ}	C^{ϵ}	-
PHE	N, C^{α}, C, C^{β}	O	C^{γ}	$\boxed{C^{\delta 1}, C^{\delta 2}, C^{\epsilon 1}, C^{\epsilon 2}, C^{\zeta}}$	-	-
PRO	N, C^{α}, C, C^{β}	O	C^{γ}	C^{δ}	-	-
SER	N, C^{α}, C, C^{β}	O	O^{γ}	-	-	-
THR	N, C^{α}, C, C^{β}	O	$C^{\gamma 2}$, $O^{\gamma 1}$	-	-	-
TRP	N, C^{α}, C, C^{β}	O	C^{γ}	$C^{\delta 1}$, $C^{\delta 2}$, $C^{\epsilon 2}$, $C^{\epsilon 3}$, $N^{\epsilon 1}$, $C^{\eta 2}$, $C^{\zeta 2}$, $C^{\zeta 3}$	-	-
TYR	N, C^{α}, C, C^{β}	O	C^{γ}	$\boxed{C^{\delta 1}, C^{\delta 2}, C^{\epsilon 1}, C^{\epsilon 2}, O^{\eta}, C^{\zeta}}$	-	-
VAL	N, C^{α}, C, C^{β}	O	$C^{\gamma 1}$, $C^{\gamma 2}$	-	-	-

Note: some residues are symmetric and has two “correct” torsion angle values

- Each atom is grouped into a “rigid group” based on their dependence on a specific torsion angle
- Number of χ angles present is dependent on the amino acid identity.

From frames to coordinates

- Map rigid body coordinates onto the backbone frames with the predicted torsion angles to obtain final atom positions
  - Coordinates of rigid bodies in reference frame are defined as constants
- Molecular dynamics (non-deep learning) used on final coordinates to relax explicit clashes



Predicted LDDT (pLDDT)

- lDDT: local distance difference test
  - A per-atom measure of local accuracy
- Train model to predict lDDT-Cα using ground truth structure
  - I.e. pLDDT is a measure of predicted confidence
  - One value per residue
- Experimentally shown to correspond well to experimental accuracy

Model Confidence:

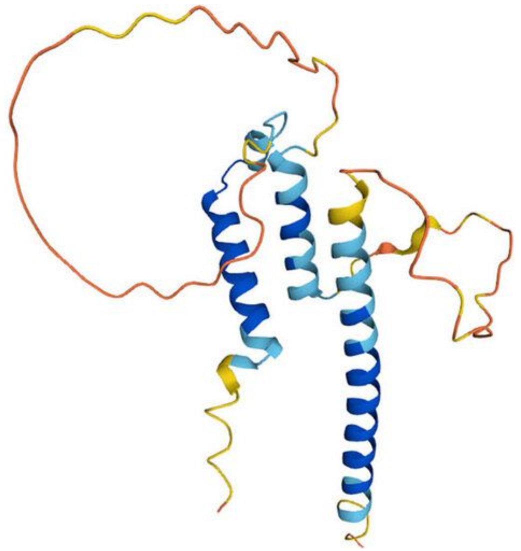
🟦 Very high (pLDDT > 90)

🟦 Confident (90 > pLDDT > 70)

🟨 Low (70 > pLDDT > 50)

🟧 Very low (pLDDT < 50)

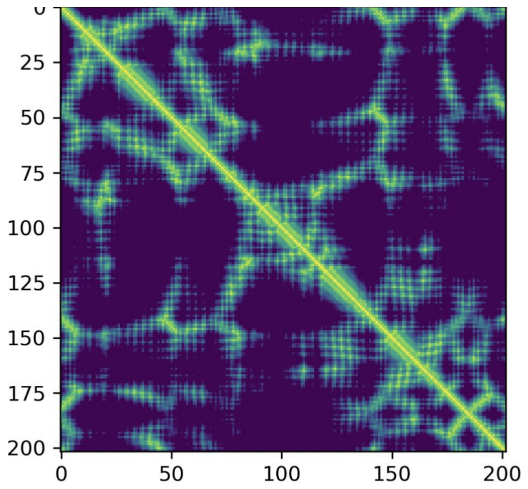
AlphaFold produces a per-residue confidence score (pLDDT) between 0 and 100. Some regions with low pLDDT may be unstructured in isolation.



Berkeley
UNIVERSITY OF CALIFORNIA

Other outputs

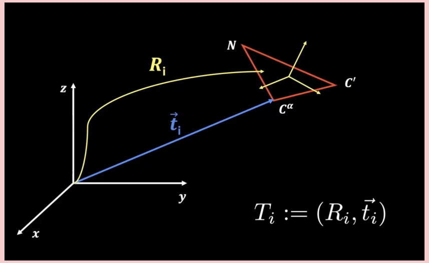
- Distogram
  - “Distance” + “histogram”
  - Bin pairwise contacts into 64 discrete bins
- Experimentally resolved prediction
  - Per-atom prediction with values between (0, 1)

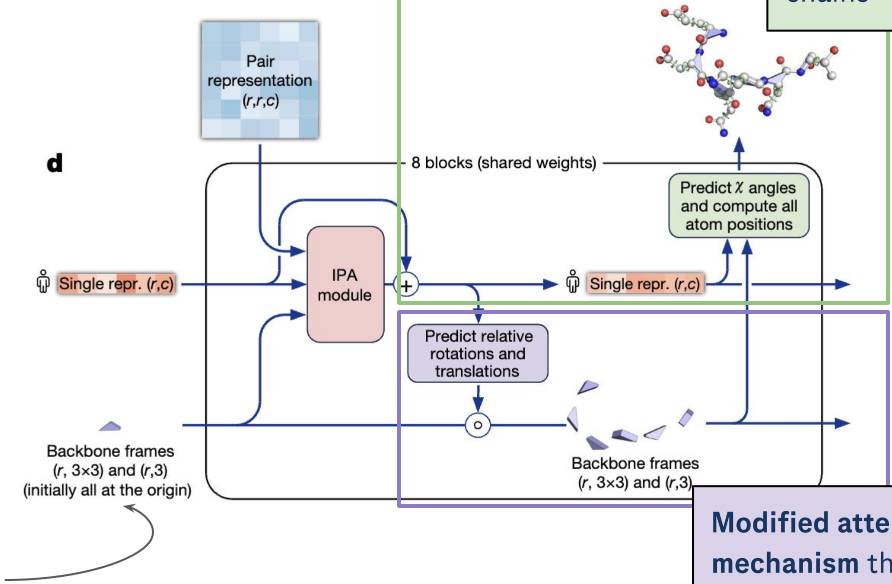


Berkeley UNIVERSITY OF CALIFORNIA

Structure module: summary

Process coordinates into **backbone frames** of rotation & translation from origin





d
Pair representation (r,r,c)
8 blocks (shared weights)
Single repr. (r,c)
IPA module
Single repr. (r,c)
Predict χ angles and compute all atom positions
Predict relative rotations and translations
Backbone frames (r, 3×3) and (r,3) (initially all at the origin)
Backbone frames (r, 3×3) and (r,3)

Predict torsion angles with separate ResNet to place side chains

Final output of **atom positions** (placed with idealized bond lengths & predicted angles) and auxiliary outputs (e.g. **pLDDT**)

Modified attention mechanism that attends to relationships between frame representations.

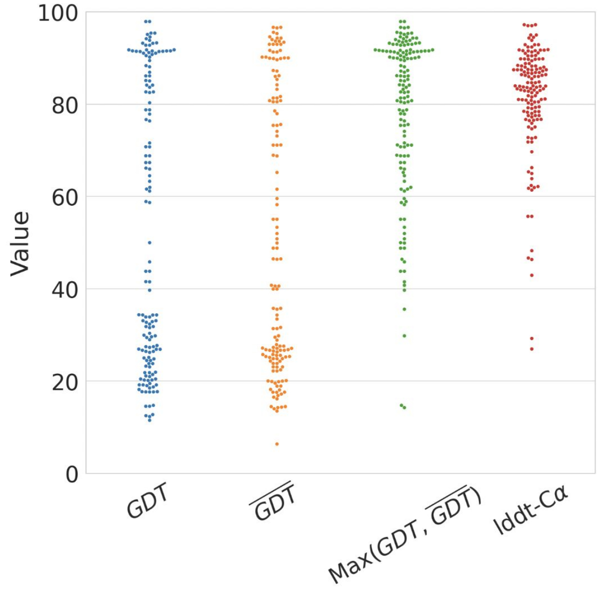
AlphaFold mechanism:

Loss Functions

Tl;dr: L2 distance on frames + losses to enforce contributions from all network components

Assessing structure prediction quality

- Desiderata for a loss function:
  - Differentiable
  - Computationally fast to calculate
- Idea 1: distance root-mean squared distance (dRMSD)
  - Take pairwise distances and calculate root-mean-squared distance
- However, it does not account for chirality
  - Recall: mirror image of a protein might have different biological functions



Bimodal errors for AlphaFold trained with the chirality-unaware dRMSD loss

Frame Adjusted Point Error (FAPE)

- Leverage the transformations associated with predicted coordinates to project back to the reference frame:

$$\vec{\mathbf{x}}_{ij} = T_i^{-1} \circ \vec{\mathbf{x}}_j$$

$$\vec{\mathbf{x}}_{ij}^{\text{true}} = T_i^{\text{true}-1} \circ \vec{\mathbf{x}}_j^{\text{true}}$$

"Undo" the predicted from-reference-frame transformation

$$d_{ij} = \sqrt{\|\vec{\mathbf{x}}_{ij} - \vec{\mathbf{x}}_{ij}^{\text{true}}\|^2 + \epsilon}$$

Calculate L2 distance for coordinates in reference frame orientation

Training loss functions

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

Training losses are designed to emphasize contributions from different architectural parts:

- L_{aux}: averaged FAPE and torsion angle losses from intermediate structure module layers
- L_{dist}: distogram prediction (cross-entropy)
- L_{msa}: masked-MSA prediction (cross-entropy)

Berkeley
UNIVERSITY OF CALIFORNIA

Training loss functions

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

Training losses are designed to emphasize contributions from different architectural parts:

- L_{aux}: averaged **FAPE and torsion angle losses** from intermediate structure module layers
- L_{dist}: distogram prediction (cross-entropy)
- L_{msa}: masked-MSA prediction (cross-entropy)

- enforces contributions from structure module
- Torsion angles calculated as unit-circle normalized L2-loss

Training loss functions

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

Training losses are designed to emphasize contributions from different architectural parts:

- L_{aux}: averaged **FAPE and torsion angle losses** from intermediate structure module layers
- L_{dist}: distogram prediction (cross-entropy)
- L_{msa}: masked-MSA prediction (cross-entropy)

- Calculates cross-entropy between real and predicted distogram
- Enforces accuracy of pairwise relationships

Berkeley
UNIVERSITY OF CALIFORNIA

Training loss functions

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

Training losses are designed to emphasize contributions from different architectural parts:

- L_{aux}: averaged **FAPE and torsion angle losses** from intermediate structure module layers
- L_{dist}: distogram prediction (cross-entropy)
- L_{msa}: masked-MSA prediction (cross-entropy)

- BERT-like masked language modelling loss
- Enforces contributions from the MSA trunk

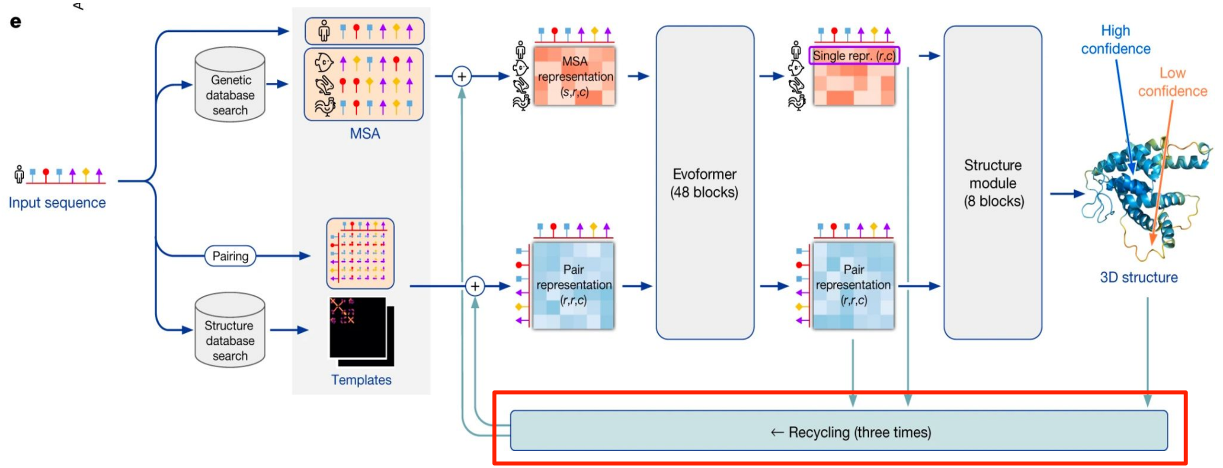
Finetuning loss functions

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

Some are only used during finetuning:

- $L_{\text{exp resolved}}$: finetunes the experimentally-resolved output with explicit labels from high-resolution structures
- L_{viol}: distogram prediction (cross-entropy)

Recycling


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)

- Output single and pair representations are reused as inputs to the Evoformer!
- Unlike stacked blocks, **weights are shared** across iterations
    - Uses stop-gradient after each iteration.

For some proteins, insufficient recycling cycles can greatly impact performance.

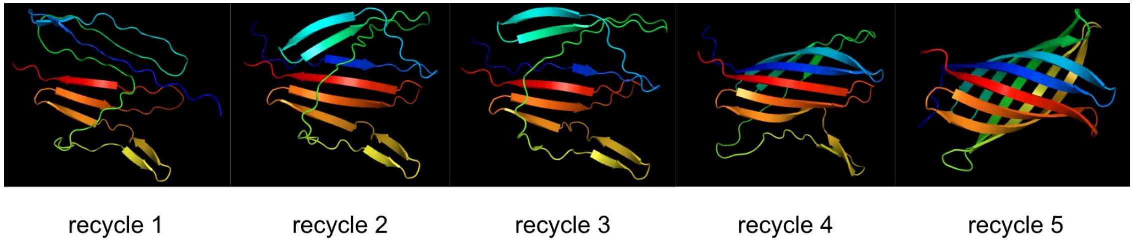
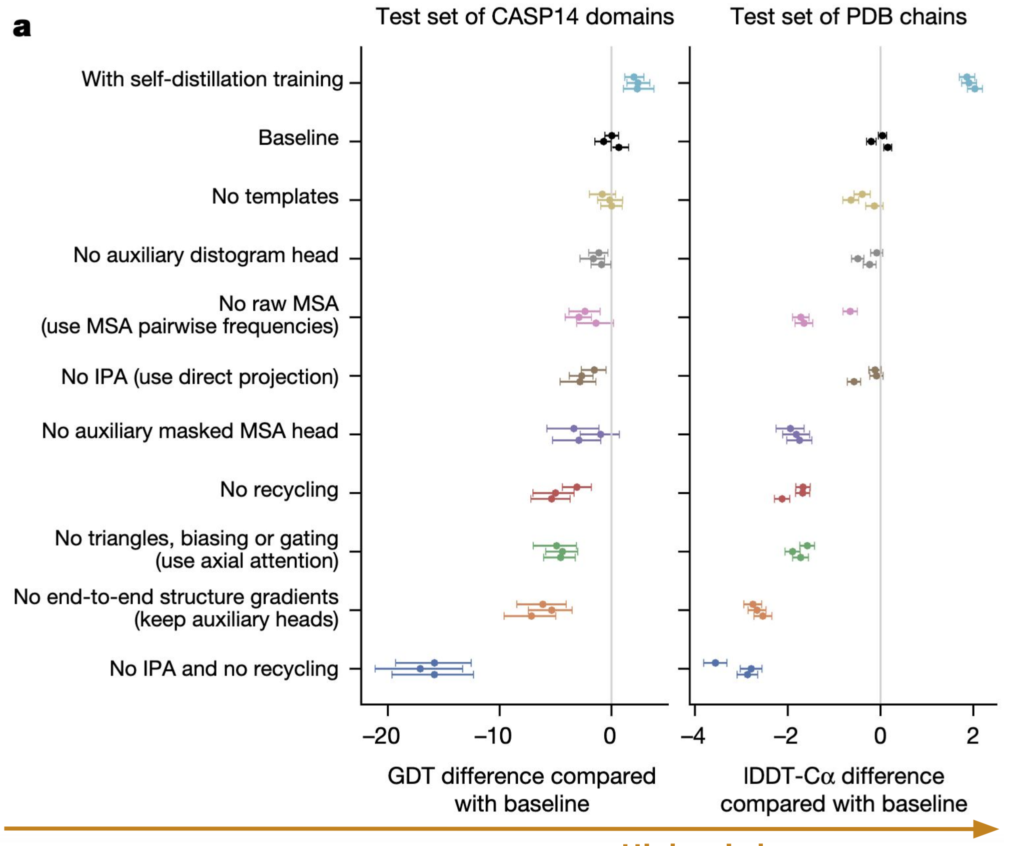


Image source: Sergey Ovchinnikov

Ablations

- “Knockdown study” to see which network components are most important



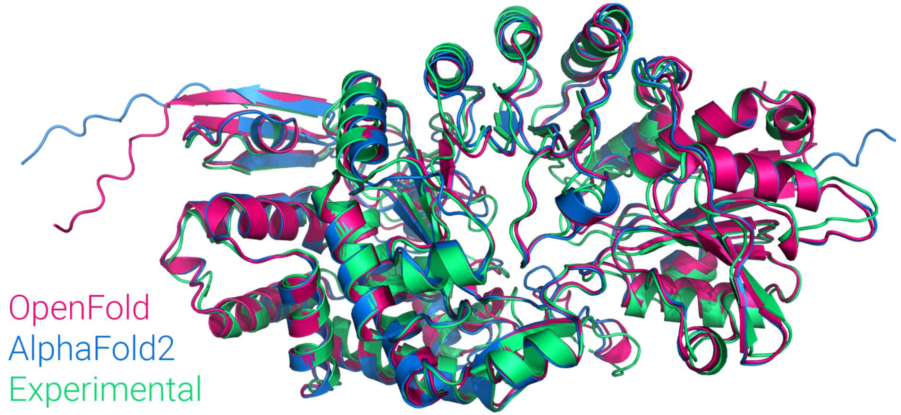
Higher is better

Limitations and Open Problems

tl;dr: generalization failures; moving towards dynamical & heterogeneous systems.

Alternative methods

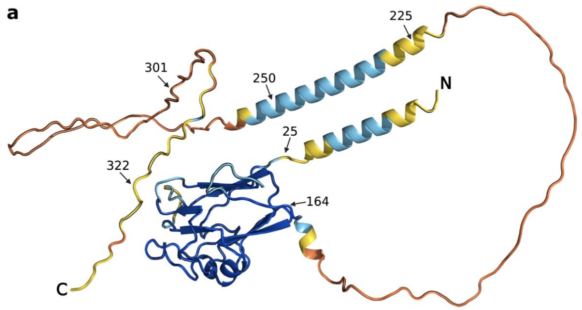
- RoseTTAFold
  - Methodologically similar to AlphaFold2, released after CASP14
- ESMFold and OmegaFold:
  - Replaces MSA input with a protein language model
- OpenFold
  - PyTorch and open-sourced reimplementation of AlphaFold



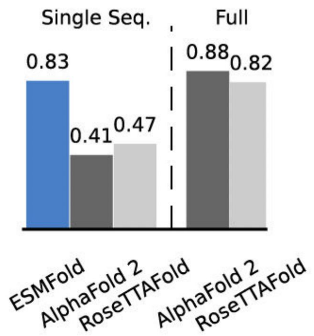
Source: https://github.com/aqlaboratory/openfold/

Limitations: Generalization failures

- PDB biases towards crystallizable proteins with stable structure
    - Performance degrades on antibody loops and intrinsically disordered regions
- MSA trunk relies on the existence of homologs
    - Performance degrades for proteins without homologs
    - Non-MSA methods such as OmegaFold and ESMFold performs better for these use cases.



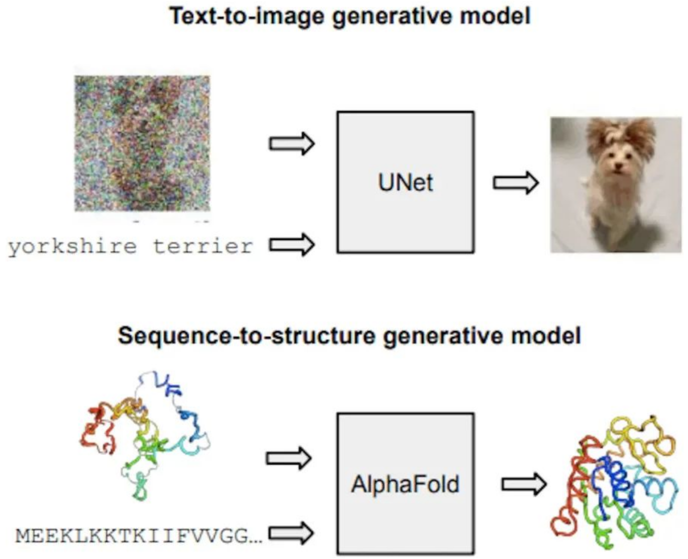
Source: Piovesan et al.



Source: Lin et al.

Limitations: Conformationally static

- Proteins are dynamic!
- Because AlphaFold is trained on crystallized structures, it is only capable of predicting static structures
- Recent works finetune AlphaFold as a generative model to predict a distribution over structures instead
    - Conceptually recapitulates the Boltzmann distribution



Text-to-image generative model
yorkshire terrier
UNet
Sequence-to-structure generative model
MEEKLKKTKIIFVVGG...
AlphaFold

Source: Jing et al.

Limitations: Modelling Complexes

- Multimers
  - AlphaFold2 at CASP14 was unable to model multi-chain protein assemblies (i.e. quaternary structure)
  - Community hacks arounds include adding a glycine linker to “glue” multiple chains into one sequence
  - AlphaFold-Multimer model released in 2022.
- Nucleic acids
  - Many important protein functions occur in tandem with nucleic acids, for e.g. protein-RNA interactions, gene editing enzymes, etc.
  - Recent work such as RoseTTAFold All-Atom (Krishna, 2024) make progress towards predicting structure for these complexes