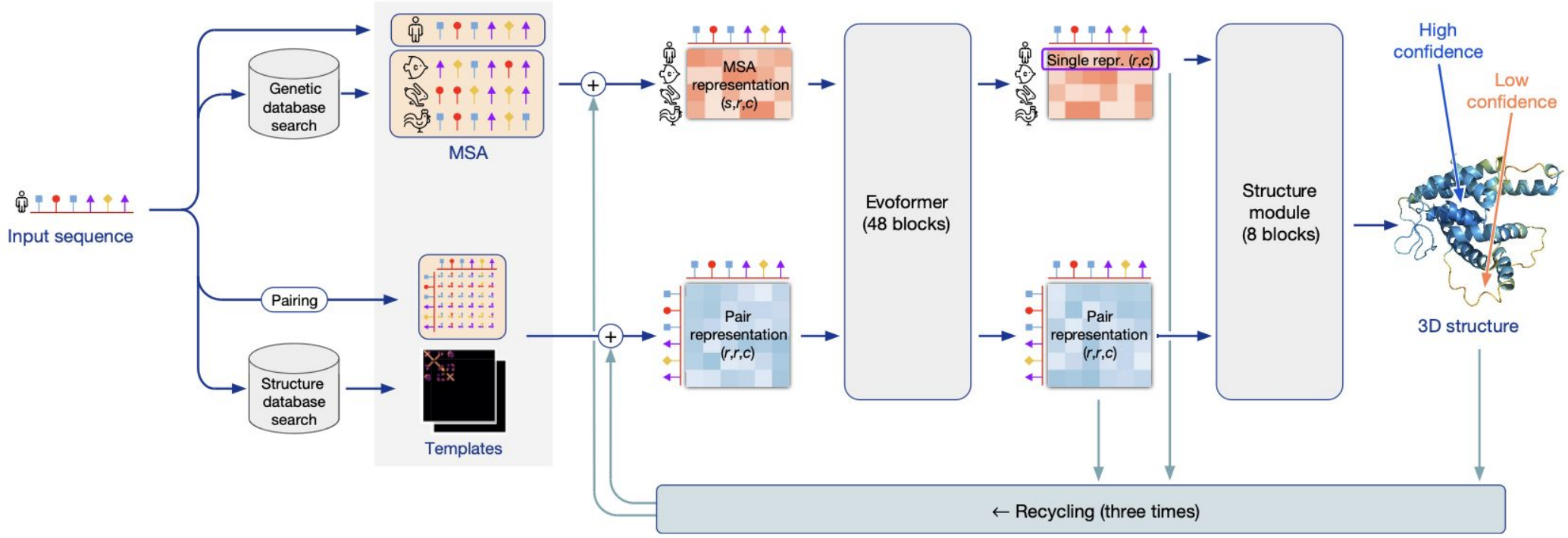
Protein Structure Prediction with Language Models

Shentong Mo

Dec 8, 2022

Recap AlphaFold2



- Multiple Sequence Alignment (MSA) + Templates of Similar Protein Structures
- Evoformer
- Structure Module

Follow-up Papers

- ESMFold (Meta AI, Alexander Rives)
  - *Lin et al*, Language models of protein sequences at the scale of evolution enable accurate structure prediction
- OmegaFold (Helixon, Jian Peng)
  - *Wu et al*, High-resolution *de novo* structure prediction from primary sequence
- HelixFold (Baidu & Biomap, Le Song)
  - *Fang et al*, HelixFold-Single: MSA-free Protein Structure Prediction by Using Protein Language Model as an Alternative

Presentation Line

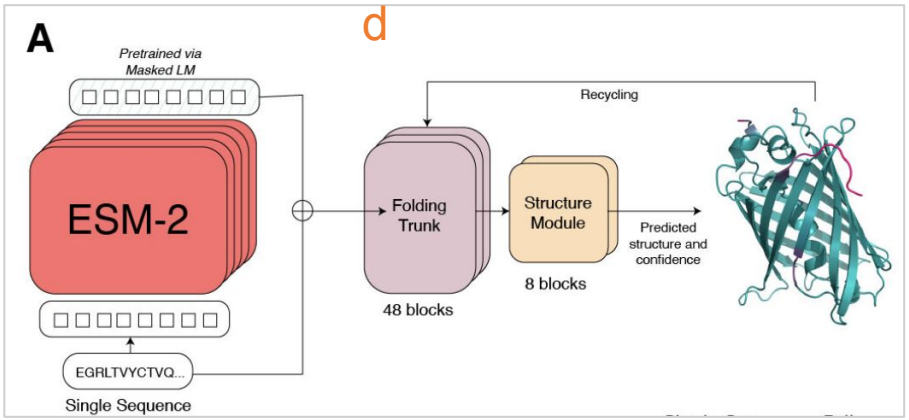
- PLM architecture
- PLM training methods
- PLM training data
- Model size
- Evoformer counterpart
- Structure finetuning data

Comparisons

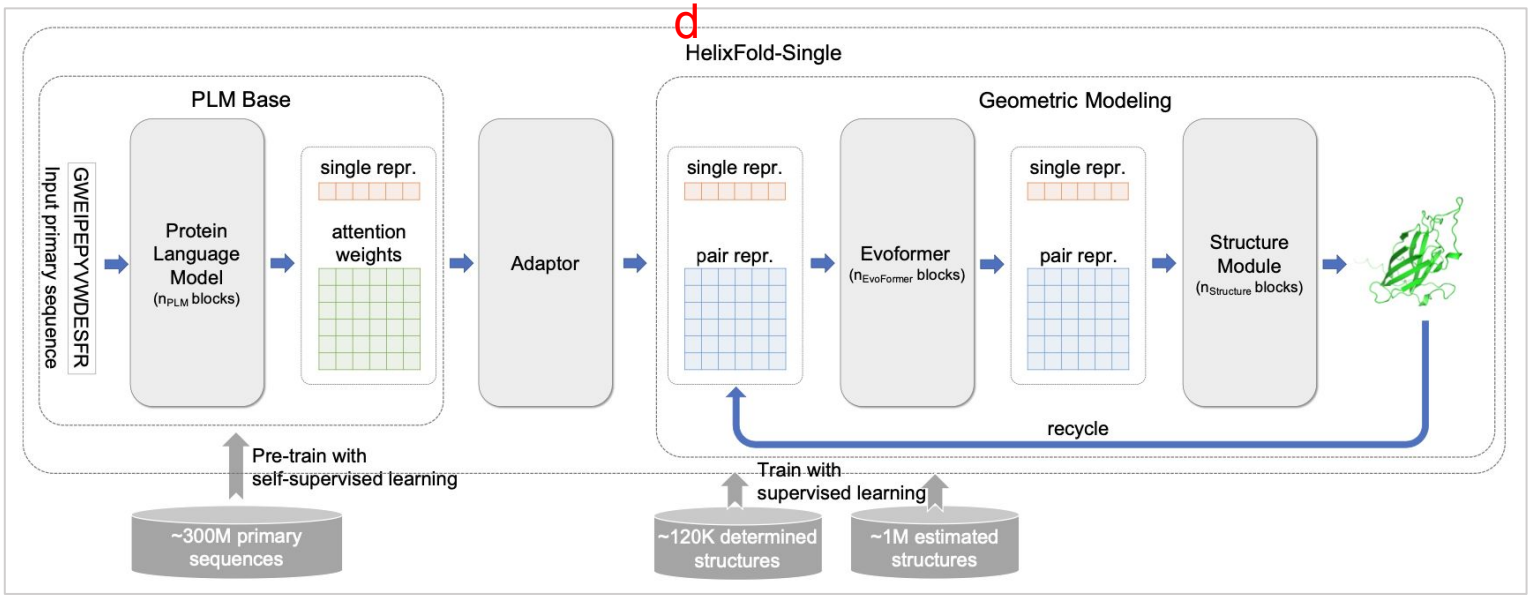
Method	PLM architectures	PLM training data	masked pretraining objective	Evoformer Counterpart	Structure fine-tuning data	Structure validation data	Freeze PLM or not
ESMFold	ESM-2 (15B)	UniRef50 (60M)	residue	ESM Folding Block	PDB + Alphafold2 Structure	CAMEO CASP14	Freeze
OmegaFold	OmegaPLM (670M)	UniRef50 (60M)	residue, motif, subsequence	Geoformer (Geometric Modeling)	PDB	CAMEO CASP13, 14	Freeze
HelixFold	PLM (1B) + Adapter	UniRef30 (260M)	residue	Revised Evoformer (Geometric Modeling)	PDB+ Uniclust30+ Alphafold2 Structure	CAMEO CASP14 MSA Depth Test	Not freeze

Overall Architecture

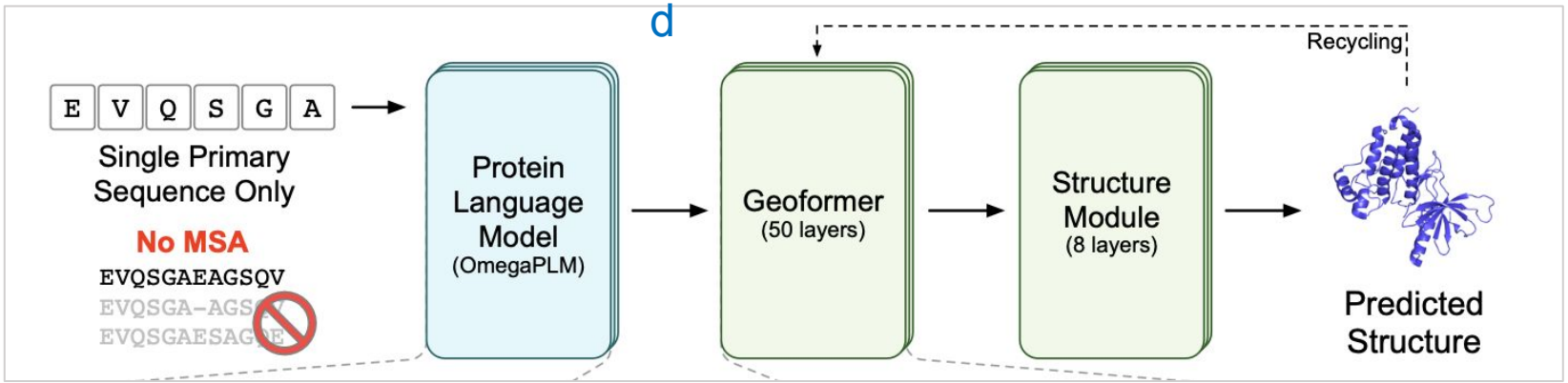
ESMFold



HelixFold

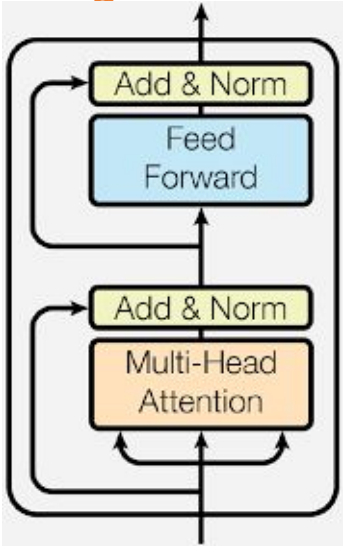


OmegaFold

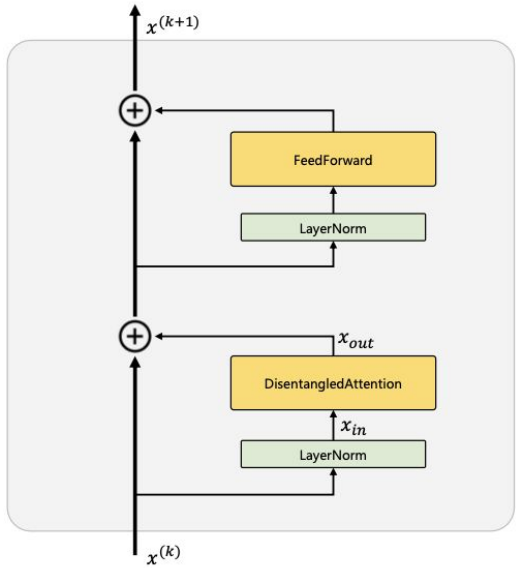


PLM Architecture

ESMFold — Standard Attention



HelixFol — Disentangled Attention

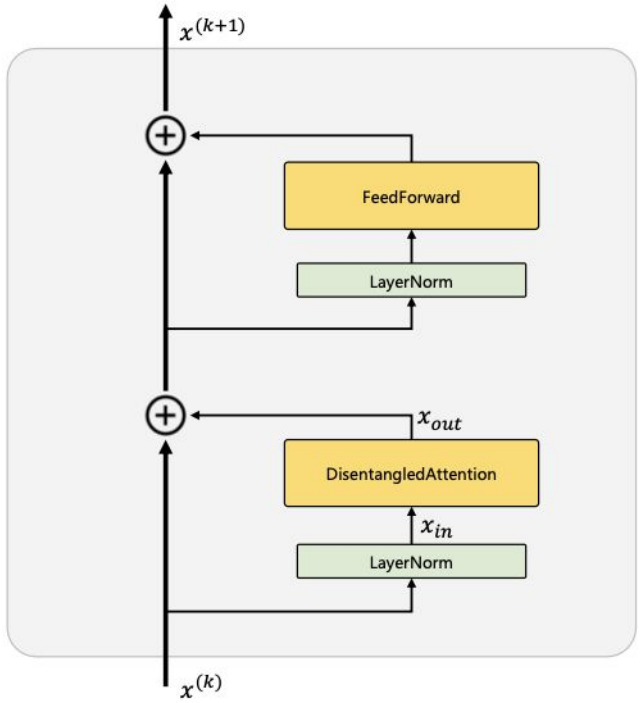


OmegaFold — Gated Attention Unit

Algorithm 1: Protein language models based on the Gated Attention Module (GAU)

def OmegaPLM($\{\boldsymbol{n}_i\}$, d_k =256, d =1,280, N_{stack} =66, d_v =2,560) **:**
  for $l \in [1, \ldots, N_{stack}]$ **do**
    $\boldsymbol{r}_i = \text{LayerNorm}(\boldsymbol{n}_i)$
    $\boldsymbol{u}_i, \boldsymbol{v}_i, \boldsymbol{g}_i = \text{SiLU}\left(\text{Linear}\left(\boldsymbol{r}_i\right)\right)$
    $\{\boldsymbol{q}_i\} = \text{RoPE}(\{\boldsymbol{w}_q \odot \boldsymbol{u}_i + \boldsymbol{b}_q\})$
    $\{\boldsymbol{k}_i\} = \text{RoPE}(\{\boldsymbol{w}_k \odot \boldsymbol{u}_i + \boldsymbol{b}_k\})$
    $\alpha_{ij} = \text{softmax}_j\left(\frac{\log n}{\sqrt{d_k}}\left(\boldsymbol{q}_i^T \boldsymbol{k}_j\right) + b_{i-j}\right)$
    $\boldsymbol{o}_i = \boldsymbol{g}_i \odot \sum_j \alpha_{ij} \boldsymbol{v}_j$
    $\boldsymbol{n}_i \mathrel{+}= \text{Linear}(\boldsymbol{o}_i)$
  end
return $\{\boldsymbol{n}_i\}$

Disentangled Attention Transformer



$$\boldsymbol{q} = \boldsymbol{x}_{in}\boldsymbol{W}_{\boldsymbol{q}}, \quad \boldsymbol{k} = \boldsymbol{x}_{in}\boldsymbol{W}_{\boldsymbol{k}}, \quad \boldsymbol{v} = \boldsymbol{x}_{in}\boldsymbol{W}_{\boldsymbol{v}}, \quad \boldsymbol{p} = \boldsymbol{e}_{p}\boldsymbol{W}_{\boldsymbol{p}}$$

$$A_{i,j} = \underbrace{\boldsymbol{q}_i\boldsymbol{k}_j^{\mathsf{T}}}_{\text{(a) residue-to-residue}} + \underbrace{\boldsymbol{q}_i\boldsymbol{p}_{\delta(i,j)}^{\mathsf{T}}}_{\text{(b) residue-to-position}}$$

$$\boldsymbol{x}_{out} = \mathrm{softmax}\left(\frac{\boldsymbol{A}}{\sqrt{2d_{PLM}}}\right)\boldsymbol{v}$$

Standard vs Disentangled Attention

Standard Attention

$$\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{q}}, \boldsymbol{K} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{k}}, \boldsymbol{V} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{v}}, \boldsymbol{A} = \frac{\boldsymbol{Q}\boldsymbol{K}^{\intercal}}{\sqrt{d}}$$

$$\boldsymbol{H}_{\boldsymbol{o}} = \mathrm{softmax}(\boldsymbol{A})\boldsymbol{V}$$

Disentangled Attention

$$\boldsymbol{Q}_{\boldsymbol{c}} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{c}}, \boldsymbol{K}_{\boldsymbol{c}} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{k},\boldsymbol{c}}, \boldsymbol{V}_{\boldsymbol{c}} = \boldsymbol{H}\boldsymbol{W}_{\boldsymbol{v},\boldsymbol{c}}, \boldsymbol{Q}_{\boldsymbol{r}} = \boldsymbol{P}\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{r}}, \boldsymbol{K}_{\boldsymbol{r}} = \boldsymbol{P}\boldsymbol{W}_{\boldsymbol{k},\boldsymbol{r}}$$

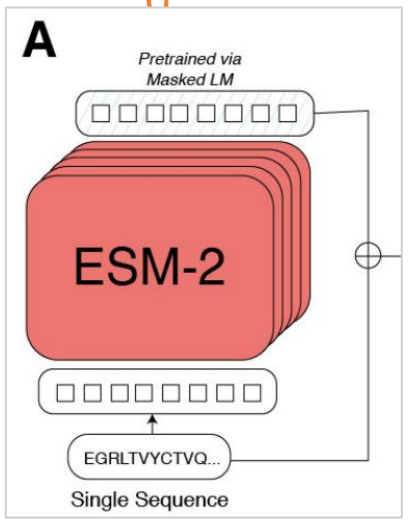
$$\tilde{A}_{i,j} = \underbrace{\boldsymbol{Q}_{\boldsymbol{i}}^{\boldsymbol{c}}\boldsymbol{K}_{\boldsymbol{j}}^{\boldsymbol{c}\intercal}}_{\text{(a) content-to-content}} + \underbrace{\boldsymbol{Q}_{\boldsymbol{i}}^{\boldsymbol{c}}\boldsymbol{K}_{\boldsymbol{\delta}(\boldsymbol{i},\boldsymbol{j})}^{\boldsymbol{r}}}^{\intercal}_{\text{(b) content-to-position}} + \underbrace{\boldsymbol{K}_{\boldsymbol{j}}^{\boldsymbol{c}}\boldsymbol{Q}_{\boldsymbol{\delta}(\boldsymbol{j},\boldsymbol{i})}^{\boldsymbol{r}}}^{\intercal}_{\text{(c) position-to-content}}$$

$$\boldsymbol{H}_{\boldsymbol{o}} = \mathrm{softmax}(\frac{\tilde{\boldsymbol{A}}}{\sqrt{3d}})\boldsymbol{V}_{\boldsymbol{c}}$$

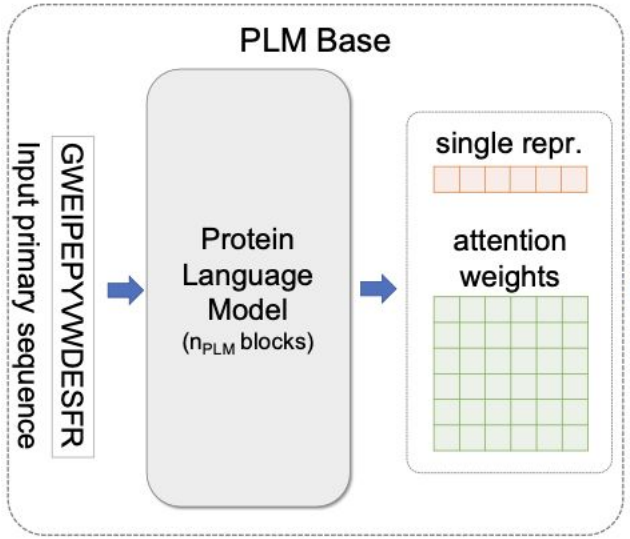
He *et al*. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, 2021 ICLR.

PLM training objectives

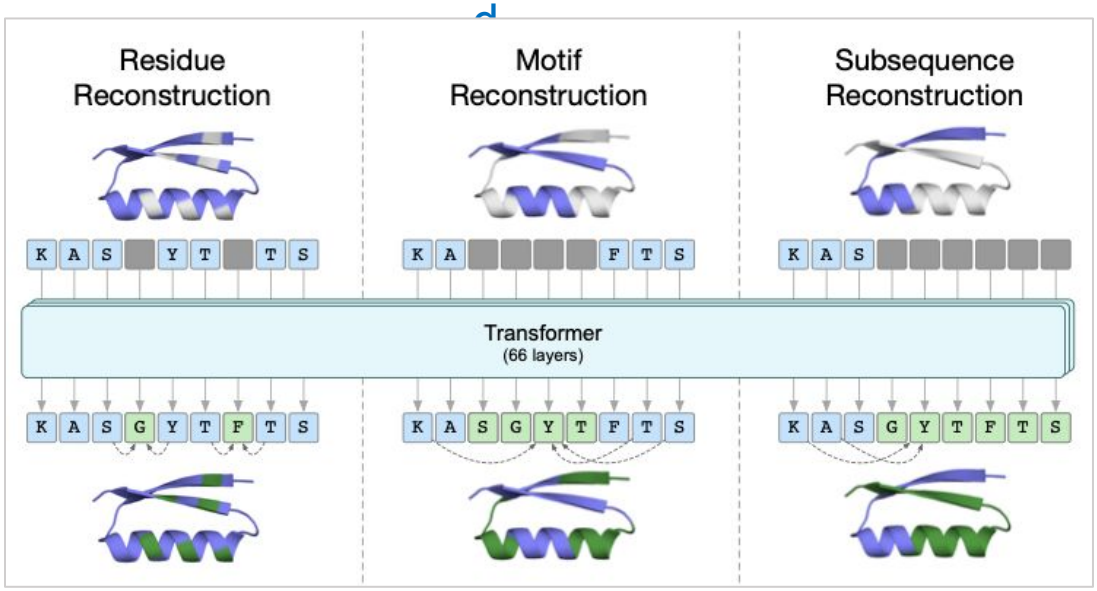
ESMFold



HelixFold



OmegaFold



PLM model size

ESMFold

	8M	35M	150M	650M	3B	15B
Dataset	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D
Number of layers	6	12	30	33	36	48
Embedding dim	320	480	640	1280	2560	5120
Attention heads	20	20	20	20	40	40
Training steps	500K	500K	500K	500K	500K	270K
Learning rate	4e-4	4e-4	4e-4	4e-4	4e-4	1.6e-4
Weight decay	0.01	0.01	0.01	0.01	0.01	0.1
Clip norm	0	0	0	0	1.0	1.0
Distributed backend	DDP	DDP	DDP	DDP	FSDP	FSDP

OmegaFold

No. Layers	66
d	1280
d_k	256
d_v	2560
No. Attention Head	1
Tying Embeddings for input & prediction head	True
Cosine normalization with learned scale (*28, 36*) at output	True
Clipping thresholds of relative positional bias	[-64, 64]
Normalization type	LayerNorm (*37*)
Pre- or Post-Norm	Pre-Norm
Learnable parameters in normalization	False

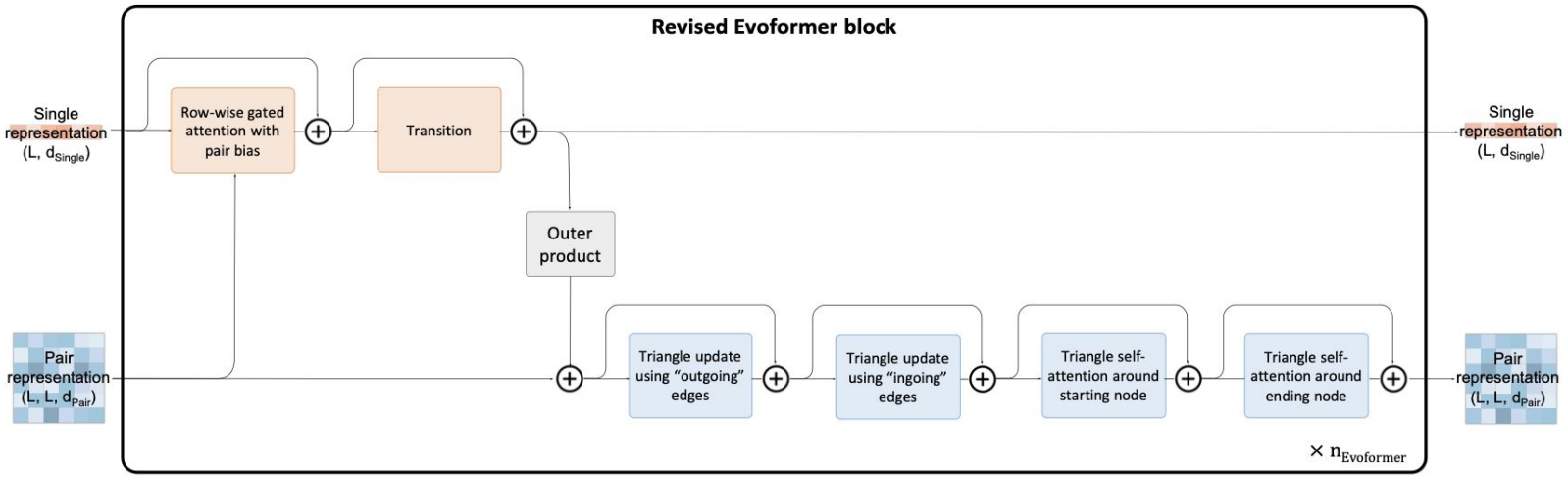
HelixFold

Components	Model size	Layer num	Hidden size	Intermediate size	Head num
PLM-1B	1.09B	$n_{PLM} = 20$	$d_{PLM} = 2048$	8192	$h_{PLM} = 16$
PLM-100M	100M	$n_{PLM} = 12$	$d_{PLM} = 768$	3072	$h_{PLM} = 12$

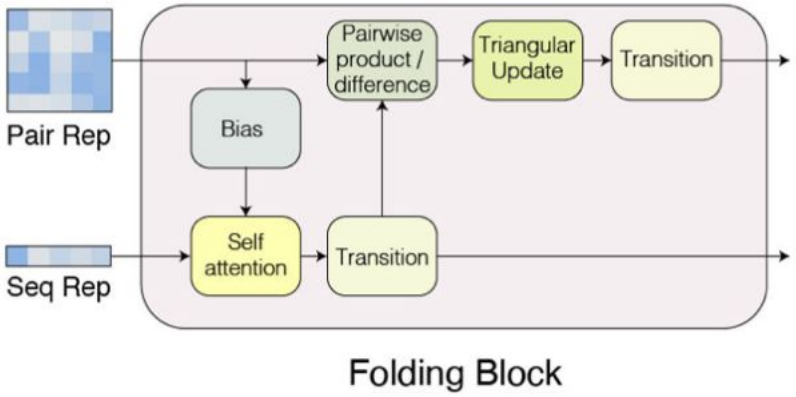
- *Disentangled Attention Transformer*

Evoformer Counterpart

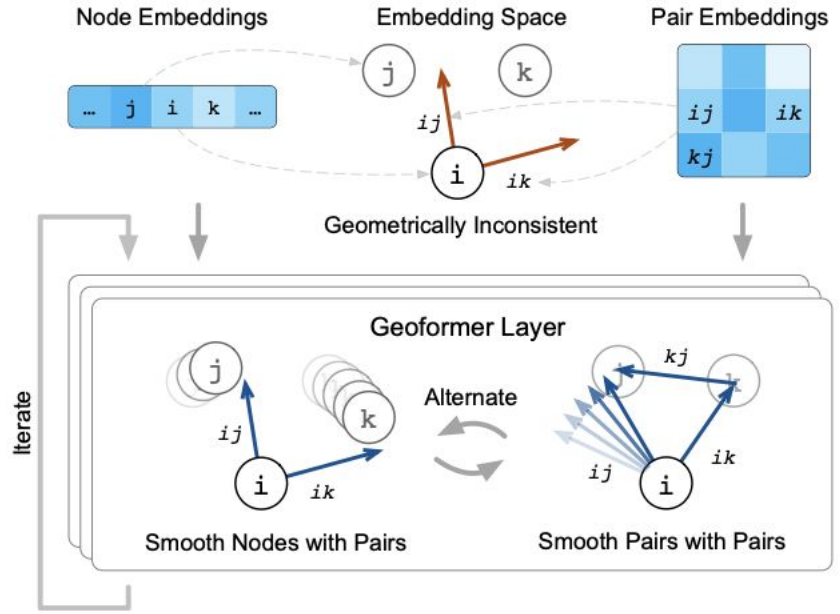
HelixFol



ESMFol



OmegaFold



Block Algorithm

ESMFold

```
Algorithm 1:
FoldingBlock(s, z)
b = Linear(z)
s = s + MultiHeadSelfAttention(s, bias=b)
s = s + MLP(s)
z = z + Linear(Concat([OuterProduct(s), OuterDifference(s)]))
z = z + TriangularMultiplicativeUpdateOutgoing(z)
z = z + TriangularMultiplicativeUpdateIncoming(z)
z = z + TriangularSelfAttentionOutgoing(z)
z = z + TriangularSelfAttentionIncoming(z)
z = z = MLP(z)
return s, z
```

HelixFold

No much details about the algorithm,

but similar to ESMFold block,

removing

column-wise attention in AlphaFold2

OmegaFold

Algorithm 3: Geometric Transformer

def Geoformer($A_{aa(i)}, \{\boldsymbol{n}_i\}, \{\boldsymbol{w}_{ij}\}, N_1 = 50, N_2 = 8, d_n = 256, d_w = 128$):
  for $l \in [1, \ldots, N_1]$ **do**
    $\{\boldsymbol{n}_i\}$ += NodeAttention($\{\boldsymbol{n}_i\}, \{\boldsymbol{w}_{ij}\}$)
    $\{\boldsymbol{n}_i\}$ += NodeTransition($\{\boldsymbol{n}_i\}$)
    $\{\boldsymbol{w}_{ij}\}$ += Node2Edge($\{\boldsymbol{n}_i\}$))
    for $k \in [1, 2]$ **do**
      $\{\boldsymbol{w}_{ij}\}$ += EdgeAttention($\{\boldsymbol{w}_{ij}\}$)
    end
    $\{\boldsymbol{w}_{ij}\}$ += EdgeTransition($\{\boldsymbol{w}_{ij}\}$))
  end
  for $l \in [1, \ldots, N_2]$ **do**
    $\{\boldsymbol{n}_i\}, \{\vec{x}_i\}$ += StructureModule($\{\boldsymbol{n}_i\}, \{\boldsymbol{w}_{ij}\}$)
    $\{\boldsymbol{w}_{ij}\}$ += 3Dprojection($A_{aa(i)}, \{\vec{x}_i\}$)
    $\{\boldsymbol{w}_{ij}\}$ += EdgeAttention($\{\boldsymbol{w}_{ij}\}$)
  end
return $\{\boldsymbol{n}_i\}, \{\boldsymbol{w}_{ij}\}$

Geometric Transformer

Node Attention

$$
\begin{aligned}
\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i &= \text{Linear}(\boldsymbol{n}_i^{(\ell-1)}) \\
\boldsymbol{b}_{ij} &= \text{Linear}(\boldsymbol{w}_{ij}) \\
\alpha_{ij} &= \text{softmax}_j\left(\frac{1}{\sqrt{c}}\boldsymbol{q}_i^T\boldsymbol{k}_j + \boldsymbol{b}_{ij}\right) \\
\boldsymbol{o}_i &= \text{sigmoid}(\text{Linear}(\boldsymbol{n}_i^{\ell-1})) \odot \sum_j \alpha_{ij}\boldsymbol{v}_j \\
\boldsymbol{n}_i^{\ell} &= \text{Linear}\left(\boldsymbol{o}_i\right)
\end{aligned}
$$

Edge Attention

$$
\begin{aligned}
\tilde{\boldsymbol{a}}_{ij}, \tilde{\boldsymbol{b}}_{ij} &= \text{sigmoid}\left(\text{Linear}(\boldsymbol{w}_{ij})\right) \odot \text{Linear}(\boldsymbol{w}_{ij}) \\
\boldsymbol{q}_{ij}, \boldsymbol{k}_{ij}, \boldsymbol{v}_{ij}, \boldsymbol{b}_{ij} &= \text{Linear}(\boldsymbol{w}_{ij}) \\
\boldsymbol{g}_{ij} &= \text{sigmoid}(\text{Linear}(\boldsymbol{w}_{ij})) \\
\alpha_{itj} &= \text{softmax}_t\left(\frac{1}{\sqrt{c}}\boldsymbol{q}_{ij}^T(\boldsymbol{k}_{it} + \boldsymbol{k}_{tj}) + \boldsymbol{b}_{it} + \boldsymbol{b}_{tj}\right) \\
\boldsymbol{o}_{ij} &= \boldsymbol{g}_{ij} \odot \left(\sum_k \alpha_{itj}(\boldsymbol{v}_{it} + \boldsymbol{v}_{tj} + \tilde{\boldsymbol{a}}_{ti} \odot \tilde{\boldsymbol{b}}_{tj})\right) \\
\boldsymbol{w}_{ij} &= \text{Linear}\left(\boldsymbol{o}_{ij}\right)
\end{aligned}
$$

Geometric Transformer (cont.)

Algorithm 4: 3Dprojection

def `3Dprojection(` $A_{aa(i)} \in \mathbb{R}^L, \{\vec{x}_i\} \in \mathbb{R}^{L\times 14\times 3}, n = 3$`):`

for $i, j \in [1, L]$ **do**

`// Embed all atom pair distances:`

$d_{ij} = ||\vec{x}_i - \vec{x}_j||$

`// Embed all atom frames`

$f^{(n)}_{aa(i)} = \text{Frame}_n(\vec{x}_i)$

`// Embed distances in both rough and fine bins`

$\boldsymbol{a}_{ij} = \text{Linear}(\text{OneHot}(d_{ij}, bins = [3.25A, 20.75A, 16]))$

$\boldsymbol{b}_{ij} = \text{Linear}(\text{OneHot}(d_{ij}, bins = [2.3125A, 21.6875A, 64]))$

`// Embed frame-position directed distance in Euclidean space:`

$\boldsymbol{c}^{(n)}_{ij} = \text{OneHot}(\text{R}(\vec{x}_i, f^{(n)}_{aa(j)}), \text{bins} = [-16\text{Å}, 16\text{Å}, 64], \text{space} = Euclidean)$

`// Embed frame-position angle distance in spherical space:`

$\boldsymbol{e}^{(n)}_{ij} = \text{OneHot}(\text{R}(\vec{x}_i, f^{(n)}_{aa(j)}), \text{bins}_\phi = [0, 2\pi, 12], \text{bins}_\psi = [0, 2\pi, 12], \text{space} = Spherical)$

`// Embed pair amino acid types:`

$\boldsymbol{g}_{ij} = \text{Linear}(\boldsymbol{a}_{ij} + \boldsymbol{b}_{ij})$

$\boldsymbol{h}_{ij} = \text{Linear}(conact_n(\{\boldsymbol{c}^{(n)}_{ij}\}) + \text{conact}_n(\{\boldsymbol{e}^{(n)}_{ij}\}))$

$\boldsymbol{z}_{ij} = A_{aa(i)} + A_{aa(j)}$

$\boldsymbol{w}_{ij} = \text{Linear}(\boldsymbol{z}_{ij} \otimes (\boldsymbol{g}_{ij} + \boldsymbol{h}_{ij}))$

end

return $\{\boldsymbol{w}_{ij}\}$

Structure training loss

- AlphaFold2:

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

- ESMFold:

$$\mathcal{L} = \mathcal{L}_{\text{FAPE}} + \mathcal{L}_{\text{dist}}$$

- OmegaFold:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{FAPE} + 0.5\mathcal{L}_{aux} + 0.3\mathcal{L}_{dist} + 0.01\mathcal{L}_{conf} & \text{first stage} \\ \mathcal{L}_{FAPE} + 0.5\mathcal{L}_{aux} + 0.3\mathcal{L}_{dist} + 0.01\mathcal{L}_{conf} + 1.0\mathcal{L}_{viol} & \text{second and last stage} \end{cases}$$

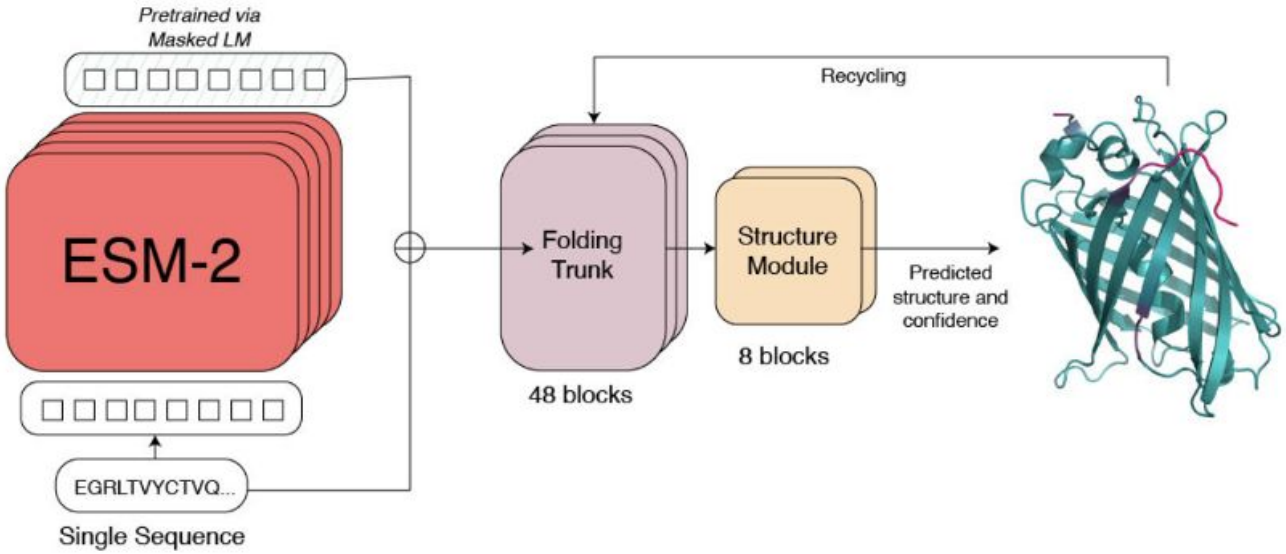
- HelixFold: removing msa loss in AlphaFold2

Structure training data

- ESMFold: (60M)
  - **RCSB PDB**: all PDB chains until 2020-05-01 with resolution greater than or equal to 9Å and length greater than 20, cluster resulting in sequences at 40% sequence identity
  - **AlphaFold2 Structure Database**: predicted lDDT greater than 70%
- OmegaFold: (number not given)
  - **RCSB PDB**: all PDB chains, cluster resulting in sequences at 40% sequence identity
- HelixFold: (260M)
  - **RCSB PDB**: all PDB chains until 2020-05-14 with resolution greater than or equal to 9Å and length greater than 20, and number of amino acids larger than 10, cluster resulting in sequences at 40% sequence identity
  - **Distillation-Uniclust30**: Uniclust30 structure predicted by AlphaFold2
  - **Distillation-EBI**: predicted lDDT greater than 50%

Thanks!

Model Structure



- ESM-2 (Replace MSA and templates in AlphaFold2)
- Folding Block (Change evoformer block in AlphaFold2)
- Structure Module

ESM-2

- Architecture: BERT-style encoder with transformers
- Modifications:
  - number of layers
  - number of attention heads
  - hidden size
  - feed-forward hidden size
  - sinusoidal positional encoding > learnable positional embedding
  - RoPE: rotary positional embedding (good for small models, bad for large models)
- Training Objective: unsupervised contact prediction with logistic regression

Unsupervised Contact Prediction

- Let c_{ij} be a boolean random variable which is true if amino acids i, j are in contact
- Suppose our transformer has L layers and K attention heads per layer.
- Let A_{kl} be the symmetrized and Average Product Correction (APC)-corrected attention map for the k-th attention head in the l-th layer of the transformer,
- and α_{ij}^{kl} be the value of that attention map at position i, j.

$$p(c_{ij}) = (1 + \exp(-\beta_0 - \sum_{l=1}^{L}\sum_{k=1}^{K}\beta_{kl}\alpha^{kl}_{ij}))^{-1}$$

Perplexity Estimation

- To measure a language model' s uncertainty of a sequence and defined as the exponential of the negative log-likelihood of the sequence
- the perplexity over a large dataset (non-deterministic)

$$Perplexity(x) = \exp\left\{-\log p(x_{i \in M} | x_{j \notin M} \cup \hat{x}_{i \in M}\right\}$$

where the mask M be a random variable denoting a set of tokens from input sequence x

- the pseudo-perplexity over a single sequence (deterministic)

$$PseudoPerplexity(x) = \exp\left\{-\frac{1}{L}\sum_{i=1}^{L}\log p(x_i | x_{j \neq i})\right\}$$

where L is the length of the sequence

ESM-2 Parameters

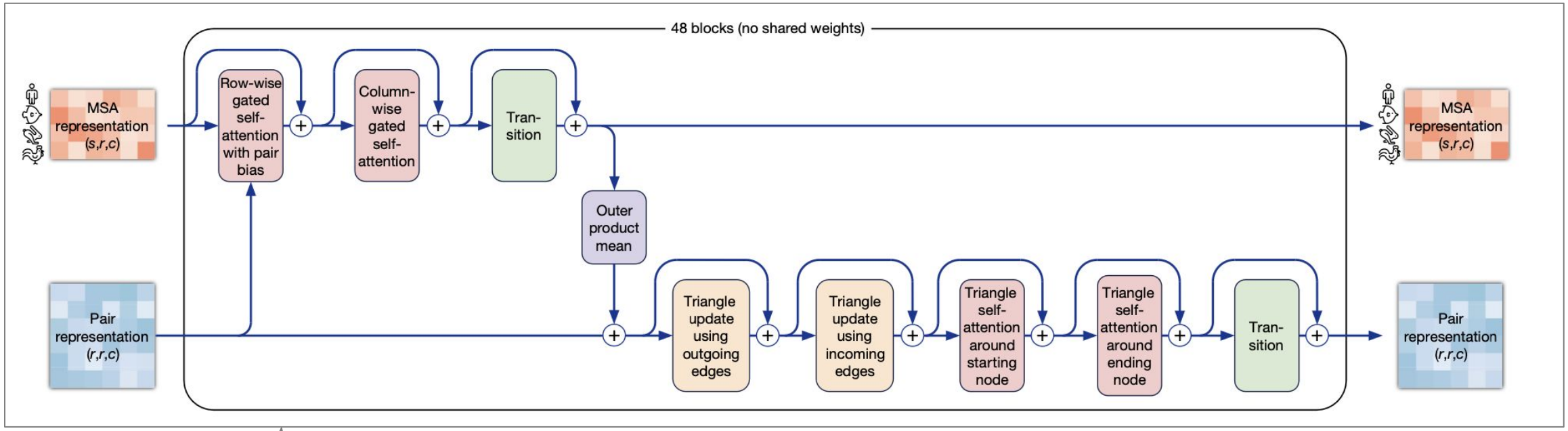
	8M	35M	150M	650M	3B	15B
Dataset	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D	UR50/D
Number of layers	6	12	30	33	36	48
Embedding dim	320	480	640	1280	2560	5120
Attention heads	20	20	20	20	40	40
Training steps	500K	500K	500K	500K	500K	270K
Learning rate	4e-4	4e-4	4e-4	4e-4	4e-4	1.6e-4
Weight decay	0.01	0.01	0.01	0.01	0.01	0.1
Clip norm	0	0	0	0	1.0	1.0
Distributed backend	DDP	DDP	DDP	DDP	FSDP	FSDP

Table S1: ESM-2 model parameters at different scales

Training data

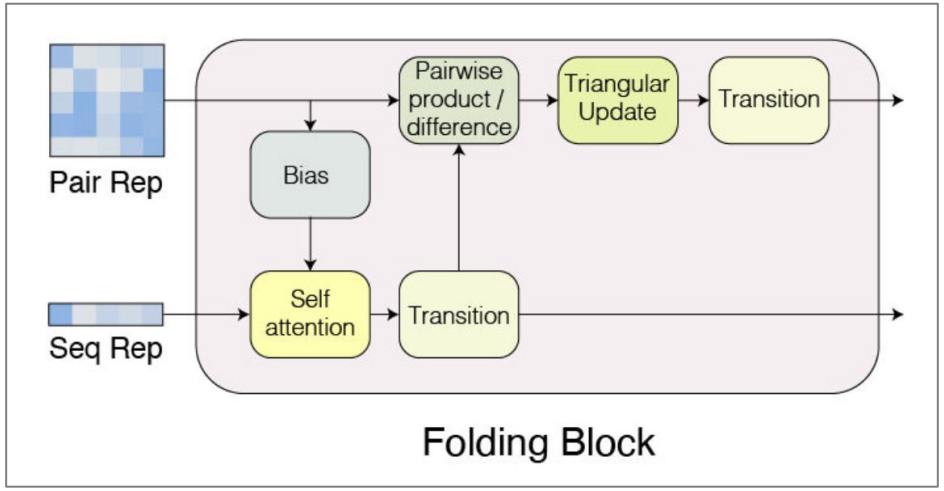
- **Data source**: UniRef50 & UniRef90 (60M protein sequences for training, 250K for validation)
  - MMseqs search to remove all train sequences matching a validation sequence with 50% identity.
- **Filtering de-novo designed proteins**:
  - remove any sequence in UniRef50 and UniRef90 that was annotated as“artificial sequence” by a taxonomy search on the UniProt website
  - use jackhmmer to remove all hits around a manually curated set of 81 de-novo proteins
- **Amount and diversity**:
  - sampled a minibatch of UniRef50 sequences for each training update
  - replaced each sequence with a sequence sampled uniformly from the corresponding UniRef90 cluster

Folding Block vs Evoformer Block

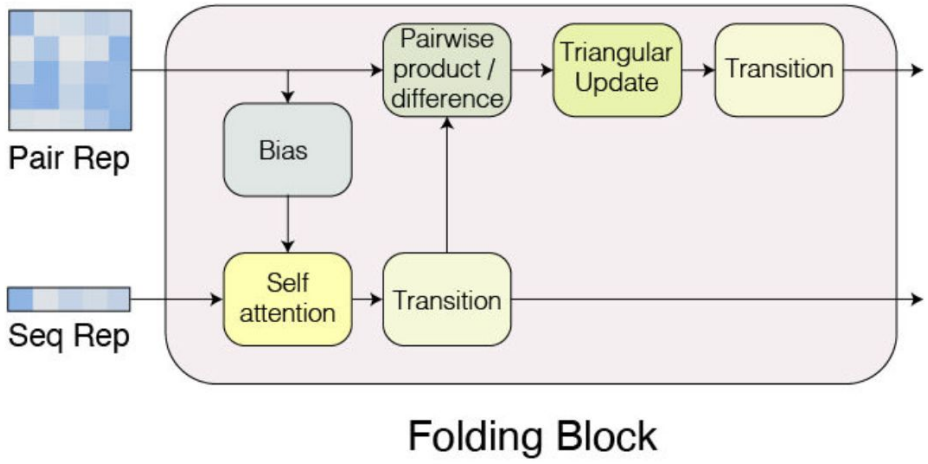


Evoformer Block

Folding Block

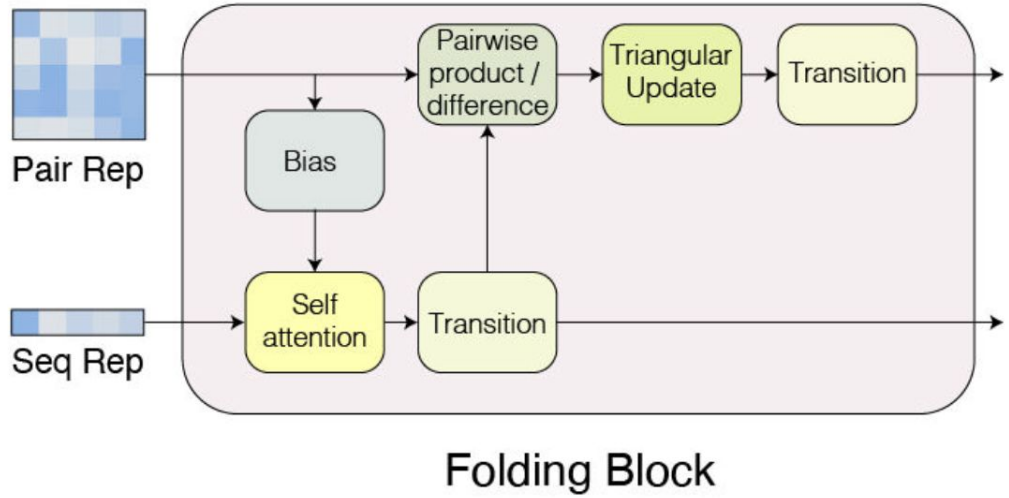


Two main changes



- use **standard attention** over this feature space, as the language model features are one dimensional
    - *Evoformer block*: employs axial attention over the columns and rows of the MSA, as MSAs are two-dimensional.
- input the **attention maps** from the language model for structure information
    - *Evoformer block*: pass template information to the model as pairwise distances, input to the residue-pairwise embedding

Folding Block Algorithm



Folding Block

```
Algorithm 1:
FoldingBlock(s, z)
b = Linear(z)
s = s + MultiHeadSelfAttention(s, bias=b)
s = s + MLP(s)
z = z + Linear(Concat([OuterProduct(s), OuterDifference(s)]))
z = z + TriangularMultiplicativeUpdateOutgoing(z)
z = z + TriangularMultiplicativeUpdateIncoming(z)
z = z + TriangularSelfAttentionOutgoing(z)
z = z + TriangularSelfAttentionIncoming(z)
z = z = MLP(z)
return s, z
```

ESMFold Algorithm

```
Algorithm 2:
esm_c_s: number of channels in ESM hidden representation
c_s = 1024
c_z = 128
ESMFold(sequence)
s = ESM_hiddens(sequence) # num_layers x Length x esm_c_s
s = (softmax(layer_weights) * s).sum(0)
s = MLP(s)
z = PairwiseRelativePositionalEncoding(Length)
for b in folding_blocks:
    s, z = b(s, z)
return StructureModule(s, z)
```

ESMFold output

- The lDDT head is output from the hidden representation of the StructureModule.
- The TM head uses the pairwise representation z.
- The distogram is predicted from the pairwise representation z.

ESMFold training loss

- AlphaFold2:

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

- ESMFold:

$$\mathcal{L} = \mathcal{L}_{\text{FAPE}} + \mathcal{L}_{\text{dist}}$$

Training structure data

- **Real Structure**:
  - all PDB chains until 2020-05-01 with resolution greater than or equal to 9Å and length greater than 20
  - cluster resulting in sequences at 40% sequence identity
- **Sampling**:
  - sampling cluster evenly
  - Rejection sampling to train longer proteins more frequently
- **Predicted Structure**:
  - 13,477,259 structures predicted using AlphaFold2 on MSAs (predicted lDDT greater than 70)
  - 75% predicted structures and 25% real structures during training

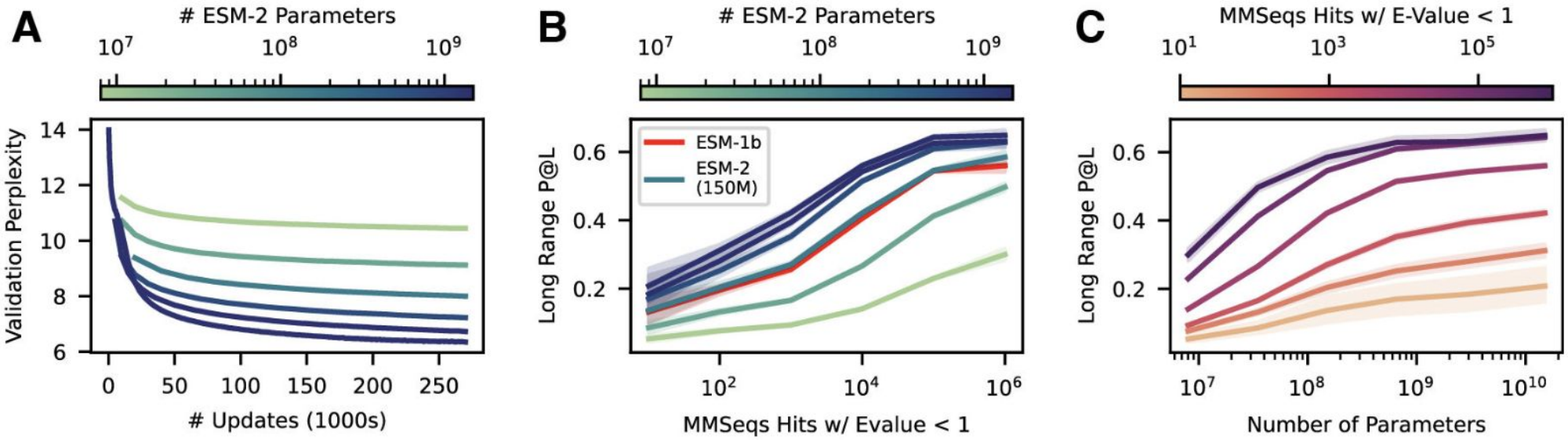
Validation & Test structure data

- Validation
  - Continuous Automated Model EvaluatiOn (CAMEO) (August 2021 to January 2022)
- Test:
  - CAMEO (194 test proteins from April 01, 2022 through June 25, 2022)
  - CASP14 competition (51 targets)
  - No filtering is performed on these test sets, even included length-2166 target T1044.

Metrics

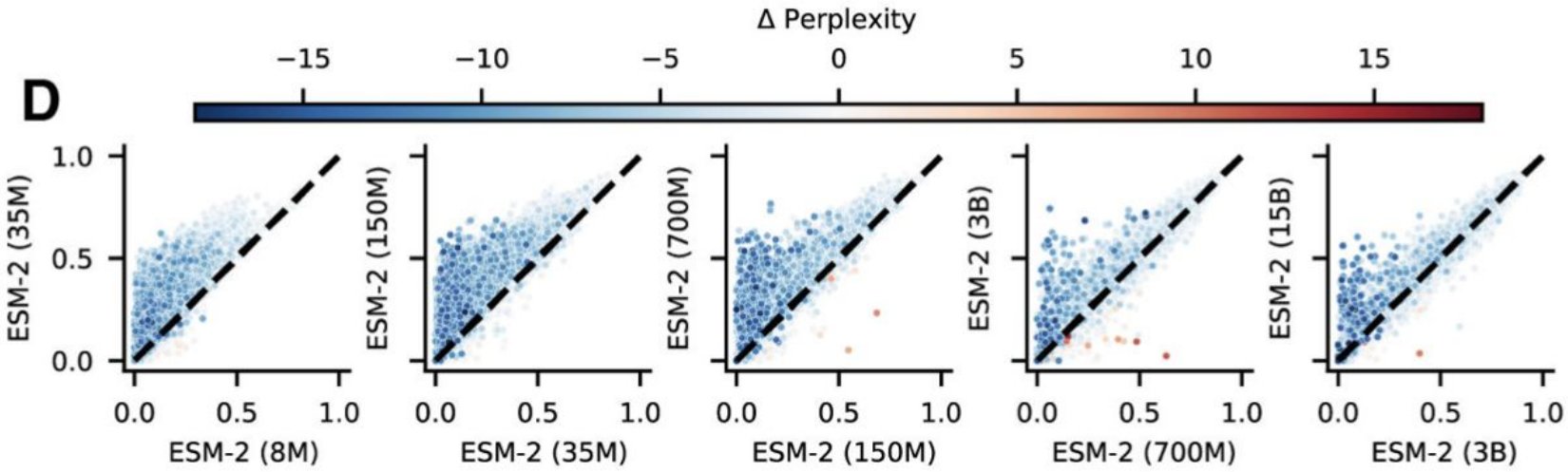
- **Validation Perplexity**: exponential of the negative log-likelihood over the validation set or a single sequence (lower is better)
- **P@L**: long-range precision @ L for unsupervised contact prediction performance (higher is better)
- **RMSD**: Root Mean Square Deviation (smaller is better)
- **TM-score**: Template Modeling score (higher is better)
- **pLDDT**: Model confidence prediction (higher is better)

Scaling up to 15B parameters



- Larger models perform better at all levels
- 150M parameter ESM-2 model performs comparably with the 650M parameter ESM-1b model.
- The largest improvement is seen for sequences with O(10^4) MMseqs hits

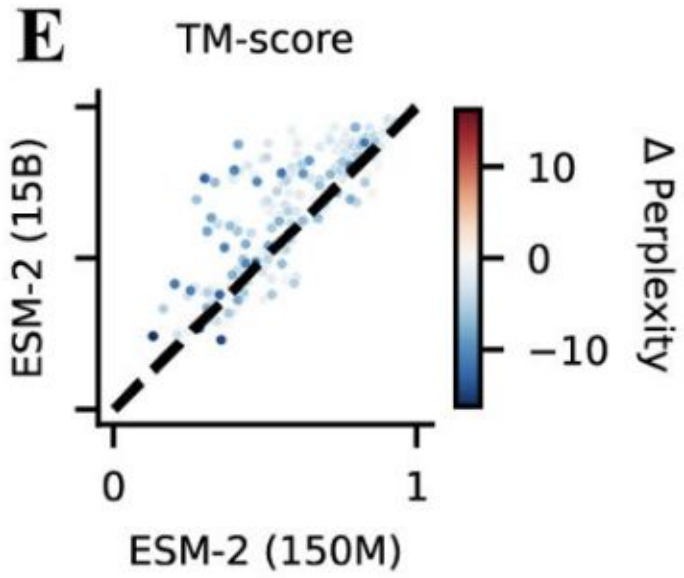
Scaling up to 15B parameters (cont.)



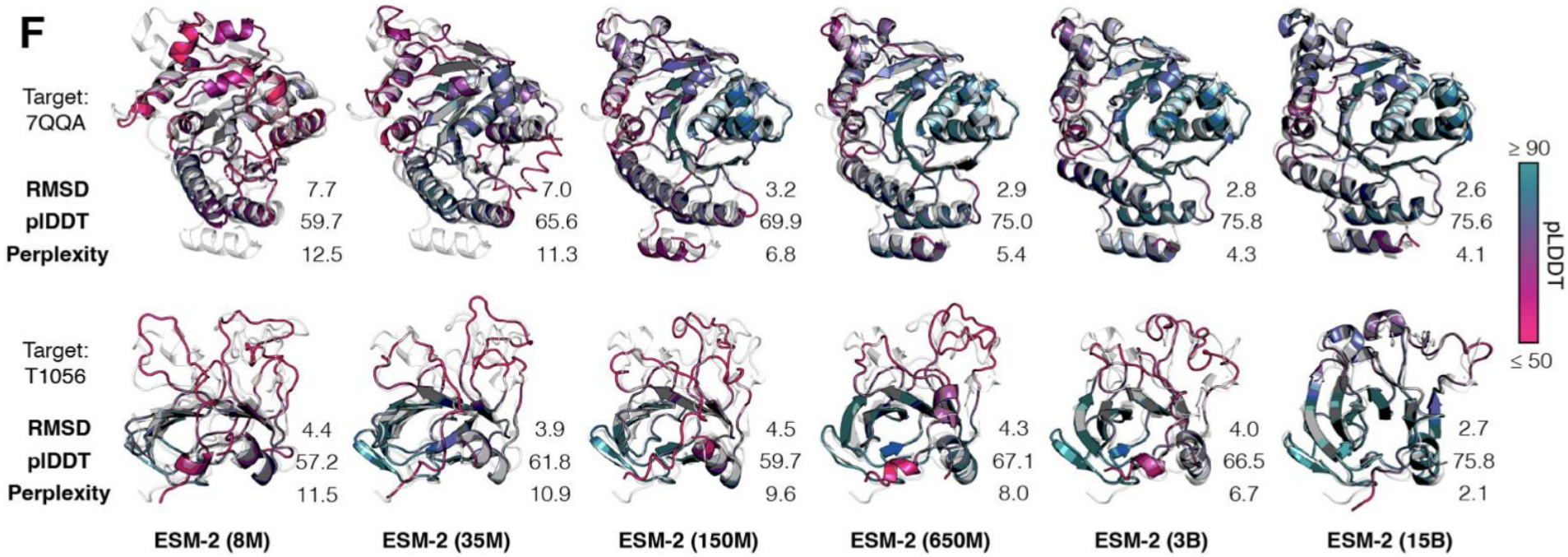
D: Left-to-right shows models from 8M to 15B parameters, consecutively comparing the smaller model (x-axis) against the next larger model (y-axis) in terms of unsupervised contact precision.

- Sequences with large changes in contact prediction performance exhibit large changes in language model understanding measured by pseudo-perplexity.

TM-score on combined CASP14 and CAMEO test sets



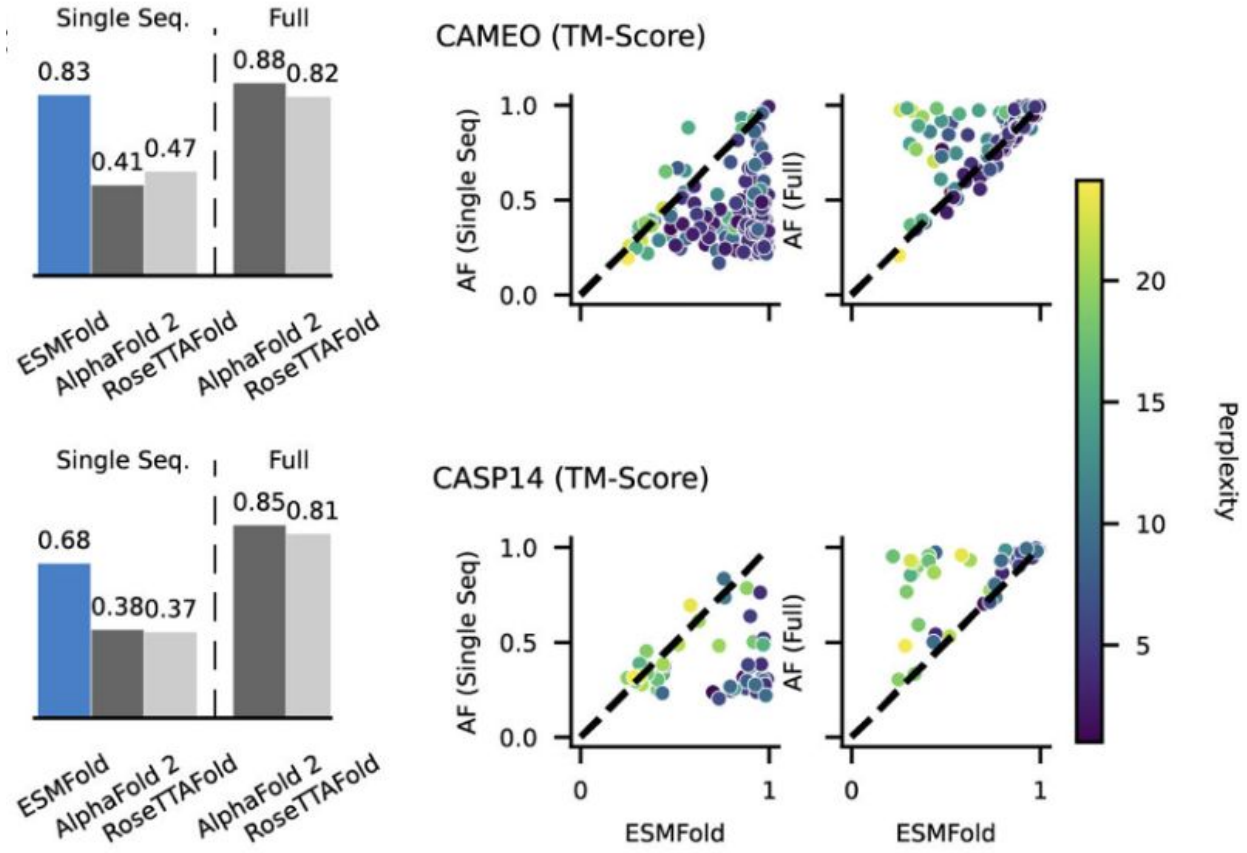
structure predictions on CAMEO structure 7QQA and CASP target 1056



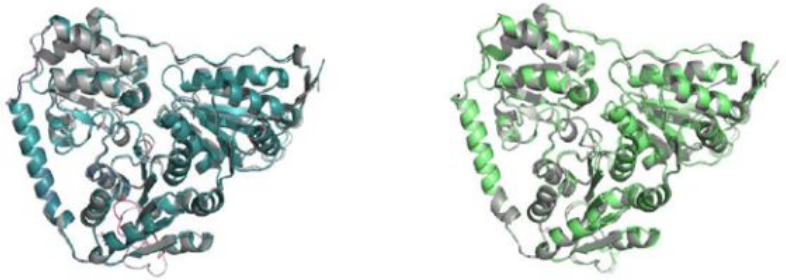
More comparisons

Model	# Params	Validation Perplexity	LR P@L	CASP14	CAMEO
ESM-2	8M	10.33	0.17	0.37	0.48
	35M	8.95	0.30	0.41	0.56
	150M	7.75	0.44	0.49	0.65
	650M	6.95	0.52	0.51	0.70
	3B	6.49	**0.54**	0.52	**0.72**
	15B	**6.37**	**0.54**	**0.55**	**0.72**
ESM-1b[1]	650M	–	0.41	0.42	0.64
Prot-T5-XL-UR50 (*19*)	3B	–	0.48	0.50	0.69
Prot-T5-XL-BFD (*19*)	3B	–	0.36	0.46	0.63
CARP (*44*)	640M	–	–	0.42	0.59

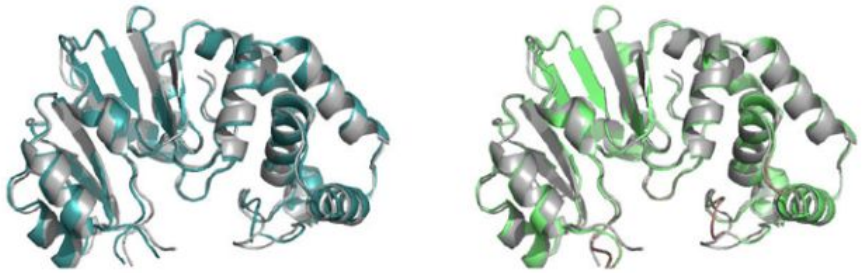
Comparison with AlphaFold2



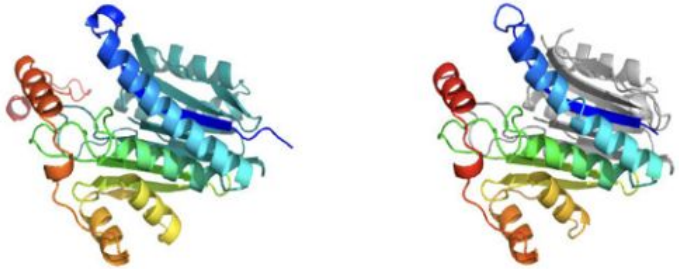
Structure prediction comparison



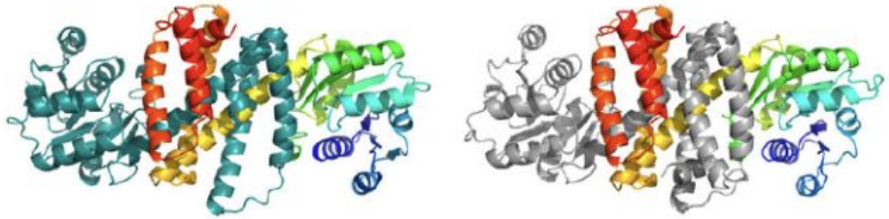
CASP14 T1076 (6XN8)
TM-score ESMFold: 0.98
TM-score Alphafold: 0.99



CASP14 T1057 (7M6B)
TM-score ESMFold: 0.98
TM-score Alphafold: 0.97



Imine Reductase (7A3W)
TM-score ESMFold: 0.956

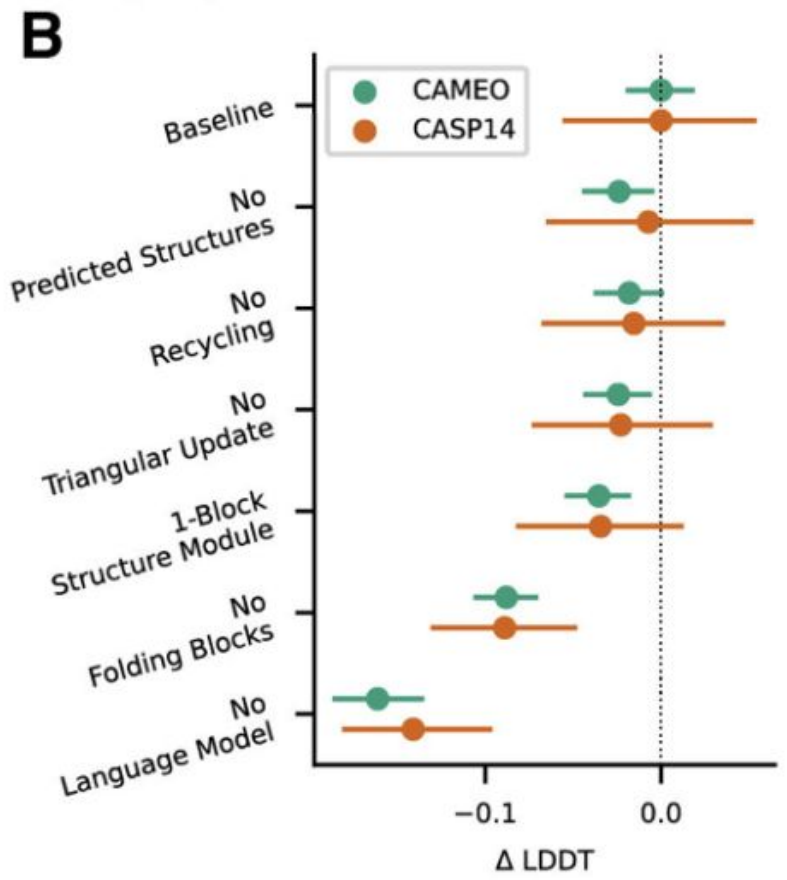


L-asparaginase (6QQ8)
TM-score ESMFold: 0.985

Ablation Study on ESM-2

	LR P@L	LR P@L/5	Validation Perplexity
Baseline	0.381	0.626	8.42
No RoPE	0.365	0.599	8.62
Older UniRef Data	0.368	0.599	7.98
No UR90 Sampling	0.387	0.631	8.40

Ablation Studies on ESMFold



Inference Time