上海科技大学
ShanghaiTech University

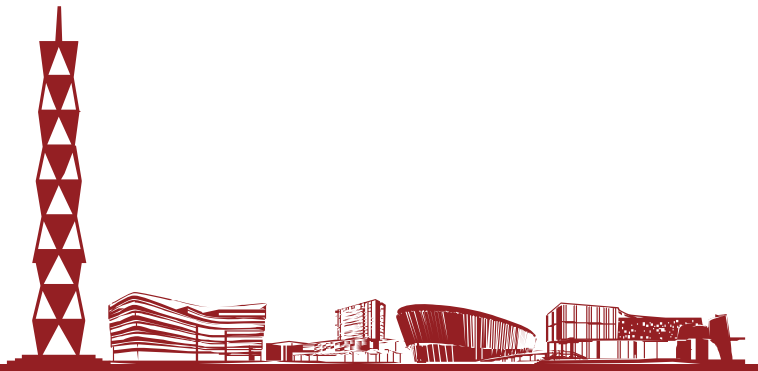
Simulating 500 Million Years of Evolution with a language Model

Yifan Qin

立志成才报国裕民

Intro: ESM2

- Only trained on the sequence information
- Transformer based structure
- BERT-like Masked Language Task

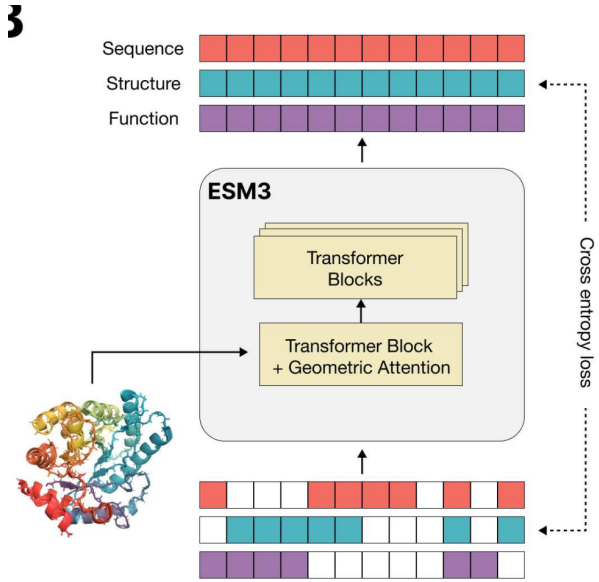


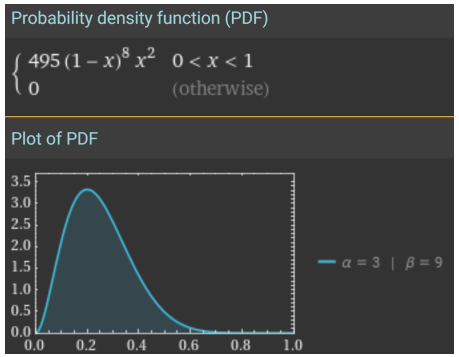
ESM3

- Trained on structure, function annotation, and sequence etc.
- Transformer blocks with attention tailored
- Masked language task with special mask strategy

ESM3

- Geometric attention to encode coordinates
- Special sequence mask strategy
  - 80% time, ratio~ $\beta(3,9)$
  - 20% time, ratio~ $\mathrm{Uniform}(0,1)$

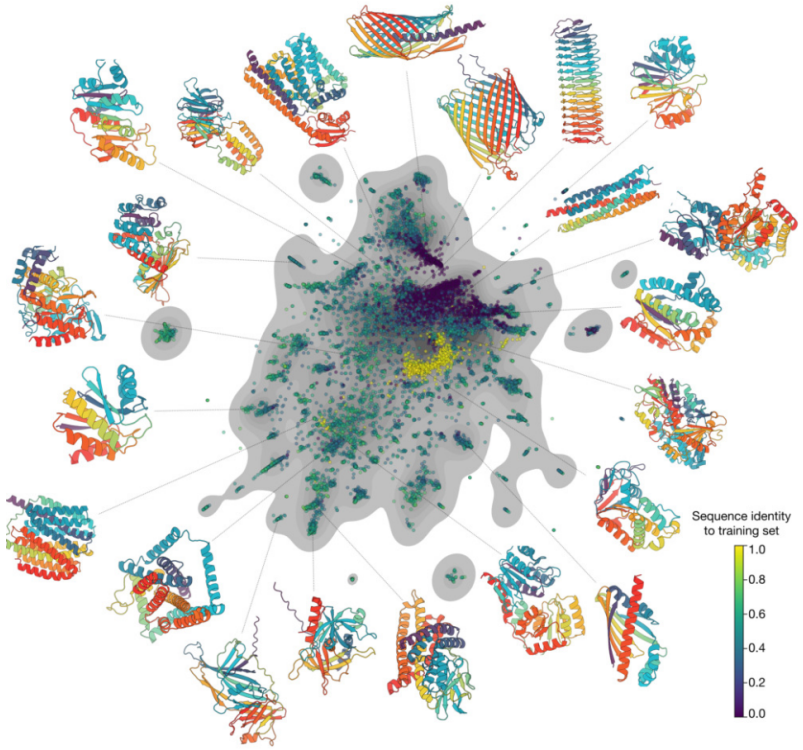


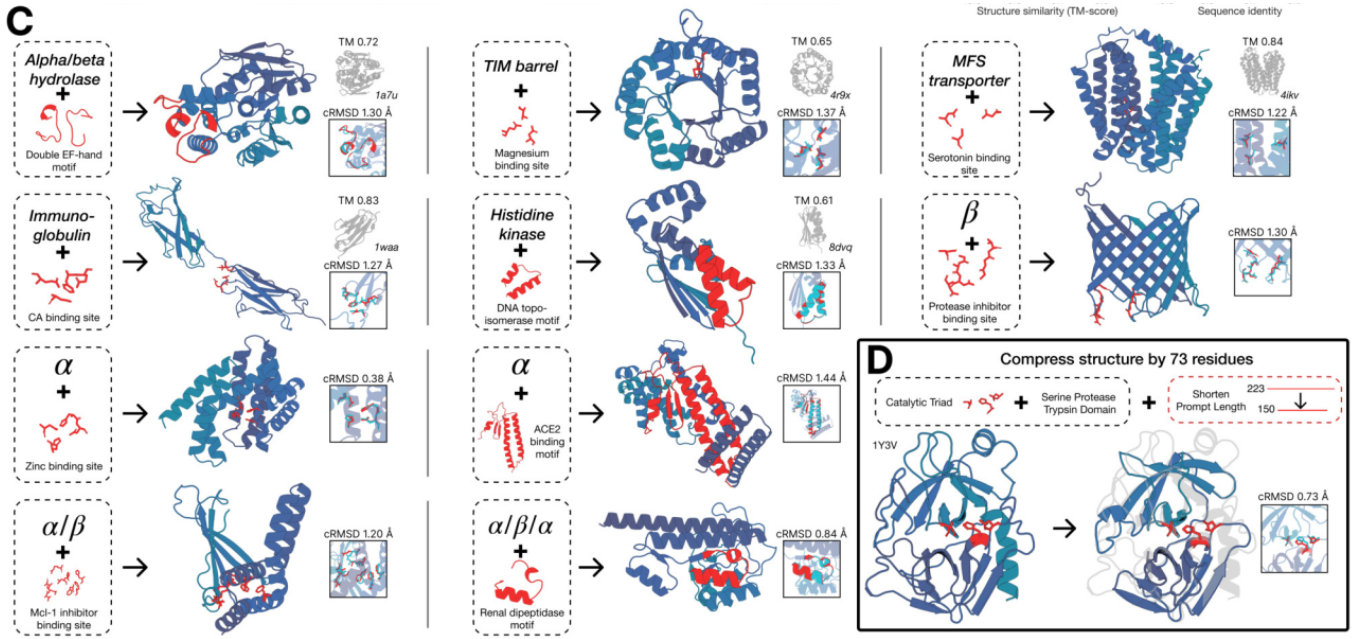


Insights & Results

上海科技大学
ShanghaiTech University

- Network architecture
- Unconditional design & Conditioned design



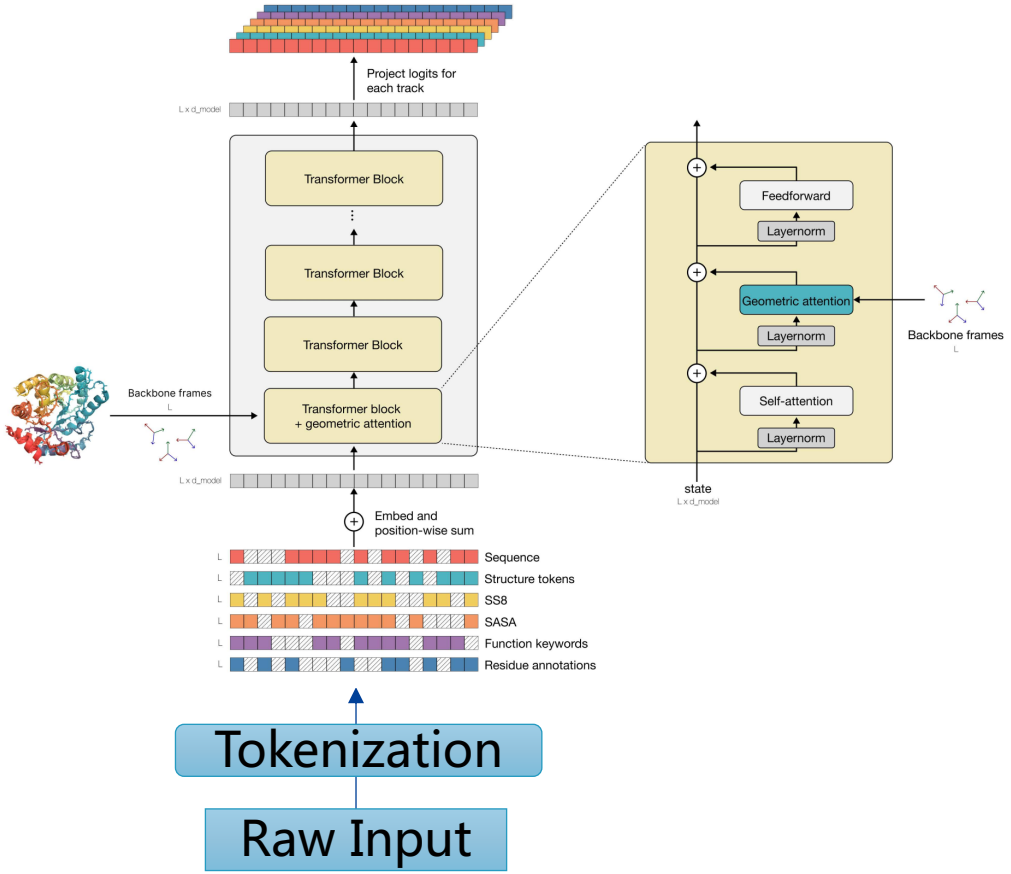


ESM3: Architecture

- Raw Input
  - Sequence info
  - Structure coordinates
  - Secondary structure(SS) annotation
  - Solvent Accessible Surface Area(SASA)
  - Functional annotation
  - Residue annotation

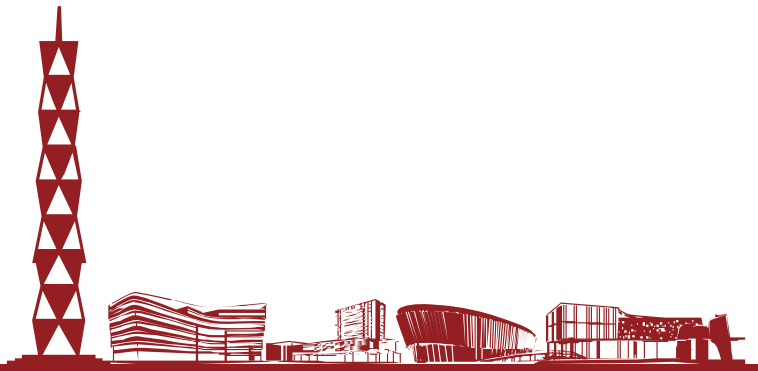
ESM3: Architecture

- Tokenization: raw input -> tokens
- Transformer Trunk: transformer directly apply on tokens
- Decoder: tokens -> each track of prediction



ESM3: Architecture: Tokenization

- Sequence info: 20 standard aa + 4 non-standard aa + BOS, EOS, mask, pad, unknown = 29 types
- SS annotation: 8 predefined SS type + mask, unknown = 10 types



ESM3: Architecture: Tokenization

- SASA: continuous area value -> 16 fixed bin + mask, unknown = 18 types
- Residue annotation: InterPro labels -> 1478 multi-hot
- Tokens:
  - Sequence: $\{0 \ldots 28\}^L$, SS: $\{0 \ldots 10\}^L$
  - SASA: $\{0 \ldots 17\}^L$, Residue annotation: $\{0,1\}^{L \times 1478}$

ESM3: Structure Tokenization

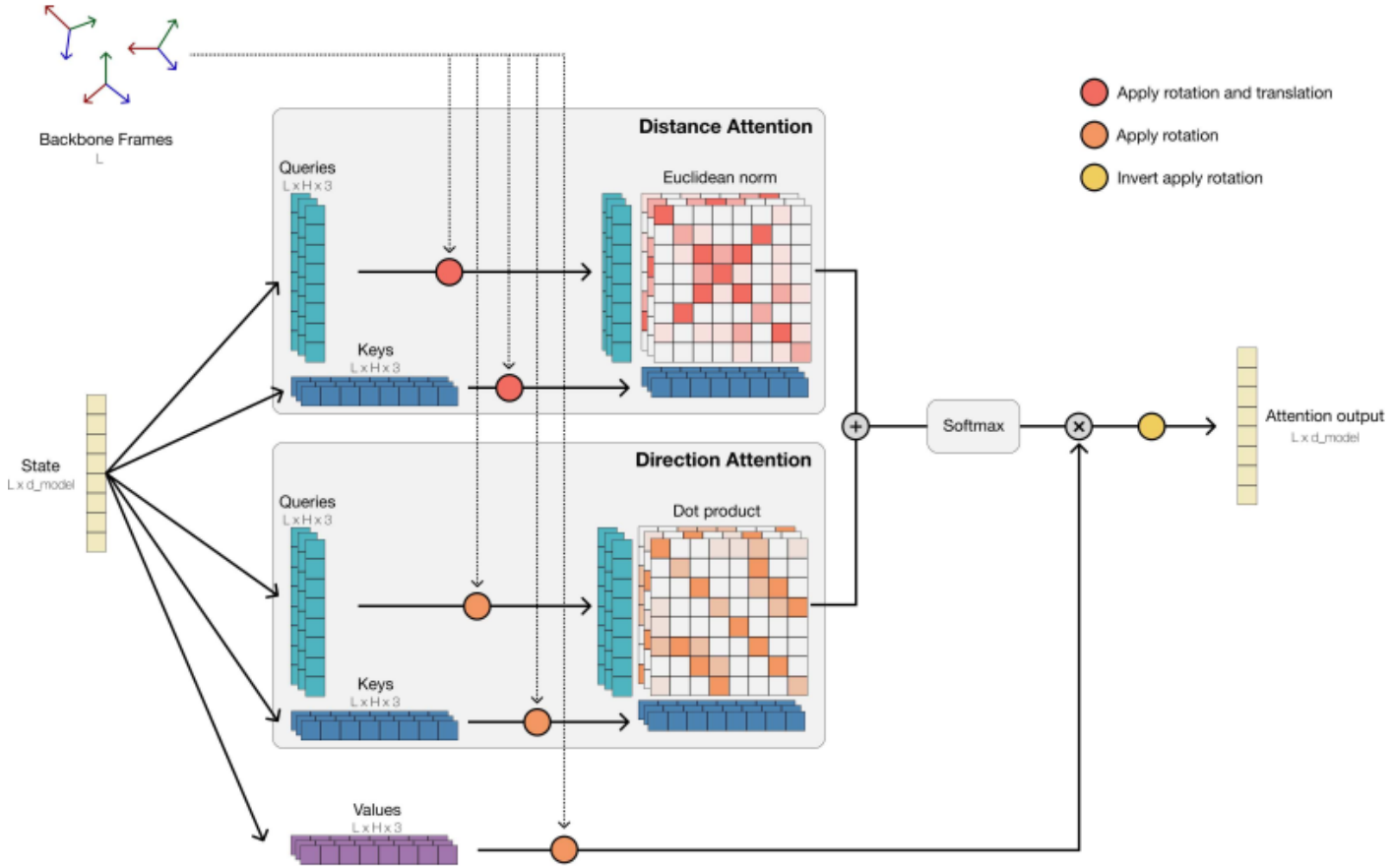
- Target: 3D coordinates to tokens for transformer input
- Use a VQ-VAE model to learn encode and decode of the coordinates
- Use encode part of VQ-VAE to get tokens for transformer input

ESM3: Structure Tokenization

- Frame & Geometric attention
- Frame $T \in SE(3)$
  - Given N, CA, C coords, get rotation and translation by gram_schmidt
  - Use rotation & translation to change between local and global coords

ESM3: Geometric Attention

- Overview



ESM3: Geometric Attention

- Given latent & frames
- Map latent to coords
- Convert coords to global distance frame and rotation frame , seen as Q, K, V

Algorithm 6 `geometric_mha`

Input: $X \in \mathbb{R}^{L \times d}, T \in SE(3)^L$

$Q_r, K_r, Q_d, K_d, V = \text{Linear}(X)$ ▷ $(\mathbb{R}^{L \times h \times 3})_{\times 5}$

$(\mathbf{R}_i, \mathbf{t}_i) = T_i$ ▷ $(SO(3)^L, \mathbb{R}^{L \times 3})$

$[Q_r]_{i,h,:} = \mathbf{R}_i([Q_r]_{i,h,:})$ ▷ $\mathbb{R}^{L \times h \times 3}$

$[K_r]_{i,h,:} = \mathbf{R}_i([K_r]_{i,h,:})$ ▷ $\mathbb{R}^{L \times h \times 3}$

$[Q_d]_{i,h,:} = T_i([Q_d]_{i,h,:})$ ▷ $\mathbb{R}^{L \times h \times 3}$

$[K_d]_{i,h,:} = T_i([K_d]_{i,h,:})$ ▷ $\mathbb{R}^{L \times h \times 3}$

$[R]_{i,j,h} = \frac{1}{\sqrt{3}}[q_r]_{i,h,:} \cdot [k_r]_{j,h,:}$ ▷ $\mathbb{R}^{L \times L \times h}$

$[D]_{i,j,h} = \frac{1}{\sqrt{3}}\|[q_r]_{i,h,:} - [k_r]_{j,h,:}\|_2$ ▷ $\mathbb{R}^{L \times L \times h}$

$A = \text{softplus}(\bar{w}_r)R - \text{softplus}(\bar{w}_d)D$ ▷ $\mathbb{R}^{L \times L \times h}$

$A = \text{softmax}_j(A)$

$[V]_{i,h,:} = \mathbf{R}_i([V]_{i,h,:})$

$O = A \cdot V$ ▷ $\mathbb{R}^{L \times h \times 3}$

$[O]_{i,h,:} = \mathbf{R}_i^{-1}([O]_{i,h,:})$

$X = X + \text{Linear}(O)$ ▷ $\mathbb{R}^{L \times d}$

ESM3: Geometric Attention

上海科技大学
ShanghaiTech University

- Attention within distance frame and rotation frame
- Combine attention and change V under rotation frame
- Map back to latent

Algorithm 6 `geometric_mha`

Input: $X \in \mathbb{R}^{L\times d}, T \in SE(3)^L$

$Q_r, K_r, Q_d, K_d, V = \text{Linear}(X)$ ▷ $(\mathbb{R}^{L\times h\times 3})_{\times 5}$

$(\mathbf{R}_i, \mathbf{t}_i) = T_i$ ▷ $(SO(3)^L, \mathbb{R}^{L\times 3})$

$[Q_r]_{i,h,:} = \mathbf{R}_i([Q_r]_{i,h,:})$ ▷ $\mathbb{R}^{L\times h\times 3}$

$[K_r]_{i,h,:} = \mathbf{R}_i([K_r]_{i,h,:})$ ▷ $\mathbb{R}^{L\times h\times 3}$

$[Q_d]_{i,h,:} = T_i([Q_d]_{i,h,:})$ ▷ $\mathbb{R}^{L\times h\times 3}$

$[K_d]_{i,h,:} = T_i([K_d]_{i,h,:})$ ▷ $\mathbb{R}^{L\times h\times 3}$

$[R]_{i,j,h} = \frac{1}{\sqrt{3}}[q_r]_{i,h,:} \cdot [k_r]_{j,h,:}$ ▷ $\mathbb{R}^{L\times L\times h}$

$[D]_{i,j,h} = \frac{1}{\sqrt{3}}\|[q_r]_{i,h,:} - [k_r]_{j,h,:}\|_2$ ▷ $\mathbb{R}^{L\times L\times h}$

$A = \text{softplus}(\bar{w}_r)R - \text{softplus}(\bar{w}_d)D$ ▷ $\mathbb{R}^{L\times L\times h}$

$A = \text{softmax}_j(A)$

$[V]_{i,h,:} = \mathbf{R}_i([V]_{i,h,:})$

$O = A \cdot V$ ▷ $\mathbb{R}^{L\times h\times 3}$

$[O]_{i,h,:} = \mathbf{R}_i^{-1}([O]_{i,h,:})$

$X = X + \text{Linear}(O)$ ▷ $\mathbb{R}^{L\times d}$

ESM3: Structure Tokenization Encode

- Find nearest neighbor in 3D (structure distance)
- Get latent from sequence distance
- Geometric attn f_{enc}
- Quantize by VQVAE codebook

Algorithm 7 `structure_encode`

Input: $x_{C_\alpha} \in \mathbb{R}^{L\times 3}, T \in SE(3)^L$

1: $N_{\text{idx}} = \text{knn}(x_{C_\alpha})$ ▷ $\{0..L-1\}^{L\times 16}$
2: $T_{\text{knn}} = T[N_{\text{idx}}]$ ▷ $SE(3)^{L\times 16}$
3: $\Delta i = \text{clamp}(N_{\text{idx}} - i, -32, 32)$
4: $N = \text{embed}(\Delta i)$ ▷ $\mathbb{R}^{L\times 16\times d}$
5: $N = f_{enc}(N, T_{\text{knn}})$ ▷ $\mathbb{R}^{L\times 16\times d}$
6: $z = \text{Linear}(N_{:,0,:})$ ▷ $\mathbb{R}^{L\times d'}$
7: $z =$ `quantize` (z) ▷ $\{0..4095\}^L$

ESM3: Structure Tokenization Decode

- Transformer on the embedding
- Direct regress backbone coords and angles for calculating torsions

Algorithm 8 `structure_decode`

Input: $z \in \{0..4099\}^{L \times 16}$

$z = \text{embed}(z)$ $\quad \triangleright \mathbb{R}^{L \times d}$

$z = f_{dec}(z)$ $\quad \triangleright \mathbb{R}^{L \times d}$

$\vec{t}, \vec{x}, \vec{y}, \overline{\sin\theta}, \overline{\cos\theta} = \text{proj}(z)$ $\quad \triangleright (\mathbb{R}^{L \times 3})_{\times 3}, (\mathbb{R}^{L \times 7})_{\times 2}$

$T = \texttt{gram\_schmidt}\,(\vec{t}, -\vec{x}, \vec{y})$ $\quad \triangleright SE(3)^L$

$\sin\theta = \frac{\overline{\sin\theta}}{\sqrt{\overline{\sin\theta}^2 + \overline{\cos\theta}^2}}$ $\quad \triangleright [-1, 1]^{L \times 7}$

$\cos\theta = \frac{\overline{\cos\theta}}{\sqrt{\overline{\sin\theta}^2 + \overline{\cos\theta}^2}}$ $\quad \triangleright [-1, 1]^{L \times 7}$

$T_{\text{local}} = \texttt{rot\_frames}\,(\sin\theta, \cos\theta)$ $\quad \triangleright SE(3)^{L \times 7}$

$T_{\text{global}} = \texttt{compose}\,(T_{\text{local}}, T)$ $\quad \triangleright SE(3)^{L \times 14}$

$\vec{X} = T_{\text{global}}(\vec{X_{ref}})$ $\quad \triangleright \mathbb{R}^{L \times 14 \times 3}$

ESM3: Structure Tokenization Decode

- Backbone coords to frame
- Torsion angle to local frame
- Use backbone frame to global frame and rotate ref positions

Algorithm 8 `structure_decode`

Input: $z \in \{0..4099\}^{L \times 16}$

$z = \text{embed}(z)$ ▷ $\mathbb{R}^{L \times d}$

$z = f_{dec}(z)$ ▷ $\mathbb{R}^{L \times d}$

$\vec{t}, \vec{x}, \vec{y}, \overline{\sin\theta}, \overline{\cos\theta} = \text{proj}(z)$ ▷ $(\mathbb{R}^{L \times 3})_{\times 3}, (\mathbb{R}^{L \times 7})_{\times 2}$

$T = \texttt{gram\_schmidt}(\vec{t}, -\vec{x}, \vec{y})$ ▷ $SE(3)^L$

$\sin\theta = \frac{\overline{\sin\theta}}{\sqrt{\overline{\sin\theta}^2 + \overline{\cos\theta}^2}}$ ▷ $[-1, 1]^{L \times 7}$

$\cos\theta = \frac{\overline{\cos\theta}}{\sqrt{\overline{\sin\theta}^2 + \overline{\cos\theta}^2}}$ ▷ $[-1, 1]^{L \times 7}$

$T_{\text{local}} = \texttt{rot\_frames}(\sin\theta, \cos\theta)$ ▷ $SE(3)^{L \times 7}$

$T_{\text{global}} = \texttt{compose}(T_{\text{local}}, T)$ ▷ $SE(3)^{L \times 14}$

$\vec{X} = T_{\text{global}}(\vec{X_{ref}})$ ▷ $\mathbb{R}^{L \times 14 \times 3}$

ESM3: Structure Tokenization Train

- Two stage
- Stage I
  - Encoder and codebook are learnt with small decoder
  - 5 losses
  - Backbone distance loss, Backbone direction loss
  - Binned dir classification loss, Distogram loss, Inverse folding loss

ESM3: Structure Tokenization Train

- Stage II
  - Encoder and codebook are frozen
  - Larger decoder applied
  - Sequence embedding added at input to decoder
  - All atom distance loss and All atom direction loss
  - pLDDT head and pAE head

ESM3: Function Tokenization

- Find frequency of keywords for annotations -> TF-IDF vector
- Max pool vector to create vector per residue
- Locality sensitive hashes quantize vector per residue -> 8 tokens, each in range 0,…,255
- Still cannot figure out, code inspection

ESM3: Tokenization

- Tokens:
  - Sequence: $\{0 \dots 28\}^L$, SS: $\{0 \dots 10\}^L$
  - SASA: $\{0 \dots 17\}^L$, Residue annotation: $\{0,1\}^{L\times 1478}$
  - Structure: $\{0 \dots 4099\}^L$, func: $\{0 \dots 258\}^{L\times 8}$
- Before transformer, they are embedded to $[L, d_{model}]$ and fused by summing with each other

ESM3: Transformer

- Transformer applies to the embedded and summed tokens
- Attention in first block is changed to Geometric attention

ESM3: Decoder

- Apply regression head to get the predicted logit and token for each track
- For sequence, ss, sasa, residue annotation, no further decode is needed

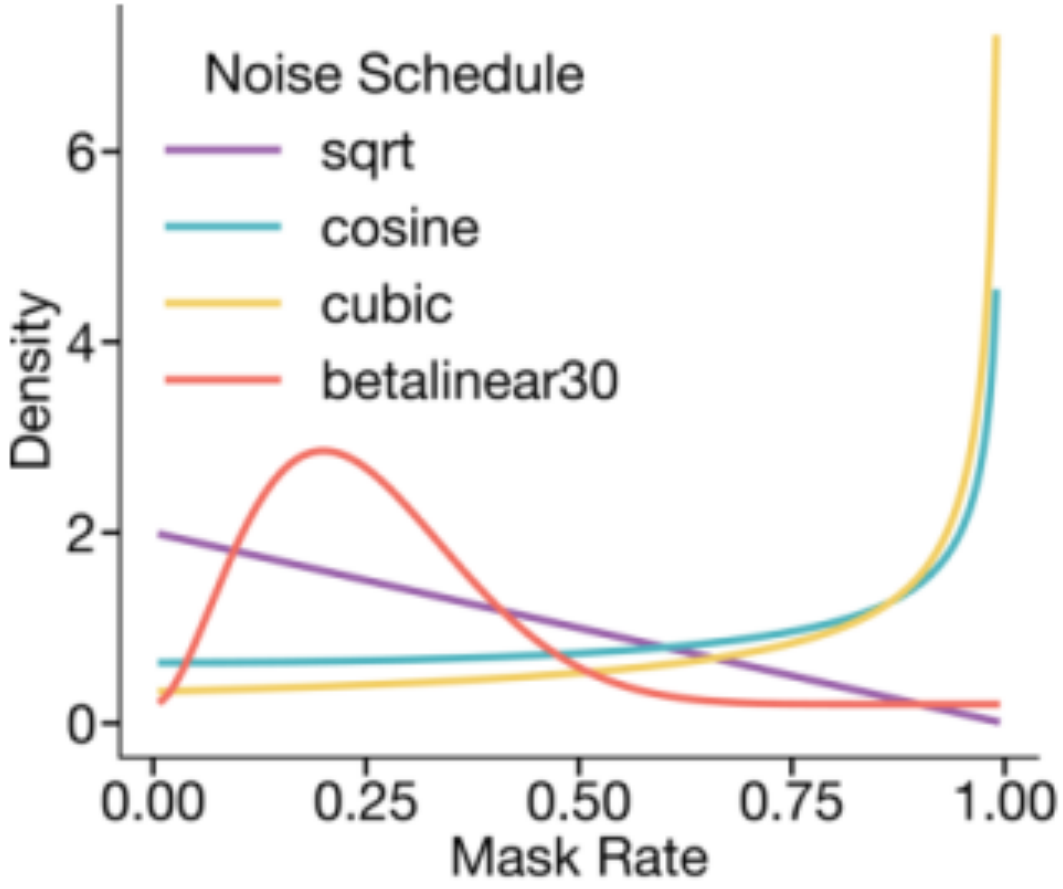
ESM3: Decoder

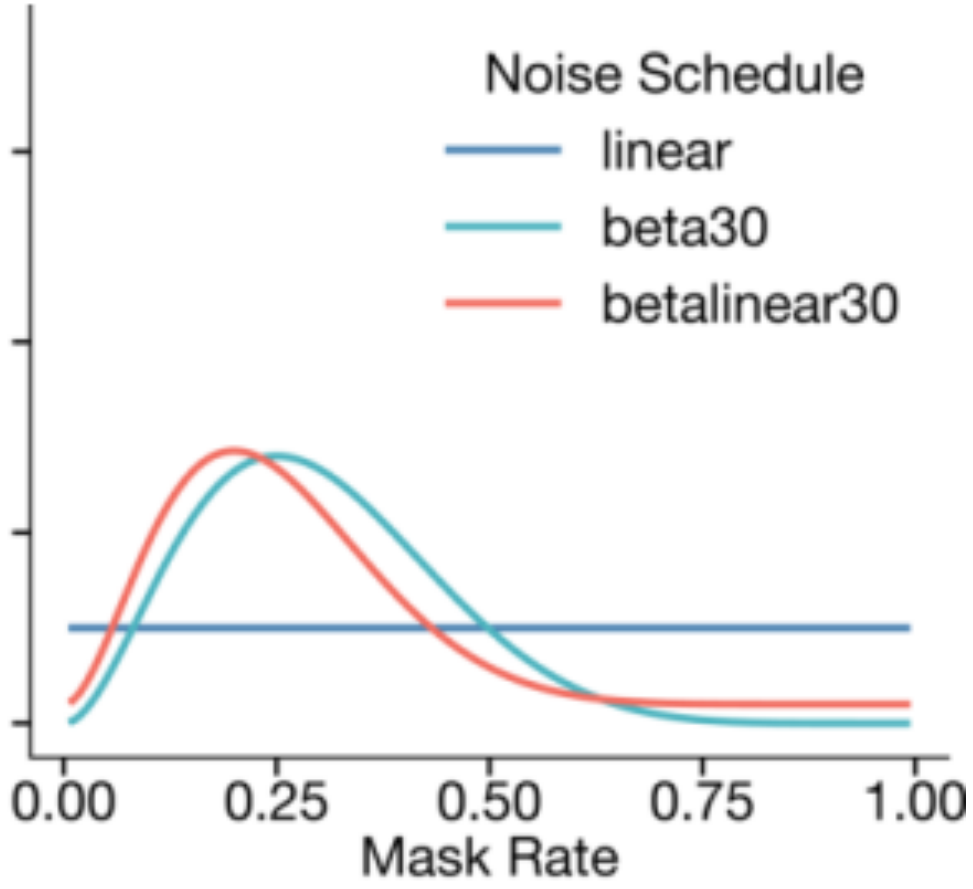
- For structure, take the token after regression head and use the decoder trained in VQVAE
- For function annotation, learn an offline 3-layer transformer as the inverse of tokenization process

上海科技大学
ShanghaiTech University

Thanks for listening

立志成才 报国裕民


Noise Schedule
sqrt
cosine
cubic
betalinear30
Density
6
4
2
0
0.00
0.25
0.50
0.75
1.00
Mask Rate


Noise Schedule
linear
beta30
betalinear30
0.00
0.25
0.50
0.75
1.00
Mask Rate

Algorithm 22 Invariant point attention (IPA)

def InvariantPointAttention($\{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, \{T_i\}, N_{\text{head}} = 12, c = 16, N_{\text{query points}} = 4, N_{\text{point values}} = 8$) :

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

$\vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$ $\quad \vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp}, \in \mathbb{R}^3,\ p \in \{1, \ldots, N_{\text{query points}}\},$ units: nanometres

$\vec{\mathbf{v}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$ $\quad \vec{\mathbf{v}}_i^{hp} \in \mathbb{R}^3,\ p \in \{1, \ldots, N_{\text{point values}}\},$ units: nanometres

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

$w_C = \sqrt{\frac{2}{9N_{\text{query points}}}},$

$w_L = \sqrt{\frac{1}{3}}$

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

$\tilde{\mathbf{o}}_i^h = \sum_j a_{ij}^h \mathbf{z}_{ij}$

$\mathbf{o}_i^h = \sum_j a_{ij}^h \mathbf{v}_j^h$

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

$\tilde{\mathbf{s}}_i = \text{Linear}\left(\text{concat}_{h,p}(\tilde{\mathbf{o}}_i^h, \mathbf{o}_i^h, \vec{\mathbf{o}}_i^{hp}, \left\|\vec{\mathbf{o}}_i^{hp}\right\|)\right)$

return $\{\tilde{\mathbf{s}}_i\}$

Algorithm 22 Invariant point attention (IPA)

def InvariantPointAttention($\{\mathbf{s}_i\}$ $, \{T_i\}, N_{\text{head}} = 12, c = 16, N_{\text{query points}} = 4, N_{\text{point values}} = 8$) :

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

$\vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$ $\quad \vec{\mathbf{q}}_i^{hp}, \vec{\mathbf{k}}_i^{hp}, \in \mathbb{R}^3,\ p \in \{1, \ldots, N_{\text{query points}}\},$ units: nanometres

$\vec{\mathbf{v}}_i^{hp} = \text{LinearNoBias}(\mathbf{s}_i)$ $\quad \vec{\mathbf{v}}_i^{hp} \in \mathbb{R}^3,\ p \in \{1, \ldots, N_{\text{point values}}\},$ units: nanometres

$w_C = \sqrt{\frac{2}{9N_{\text{query points}}}},$

$w_L = \sqrt{\frac{1}{3}}$

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

$\mathbf{o}_i^h = \sum_j a_{ij}^h \mathbf{v}_j^h$

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

$\tilde{\mathbf{s}}_i = \text{Linear}\left(\text{concat}_{h,p}(\quad \mathbf{o}_i^h, \vec{\mathbf{o}}_i^{hp}, \left\|\vec{\mathbf{o}}_i^{hp}\right\|)\right)$

return $\{\tilde{\mathbf{s}}_i\}$