ESM

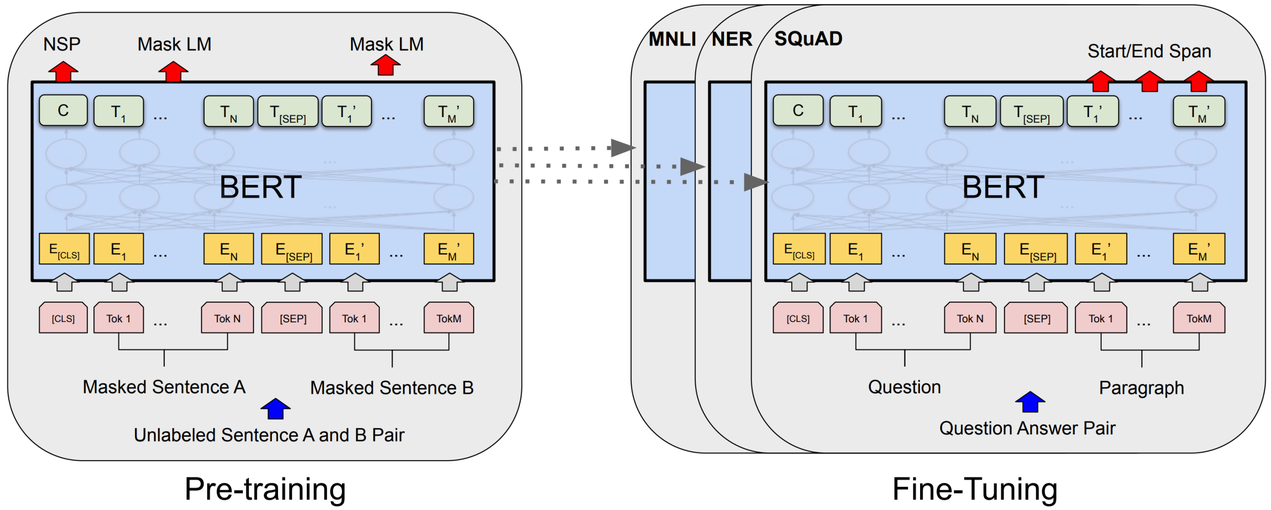
ESMC | ESMFold2 | ESM Atlas

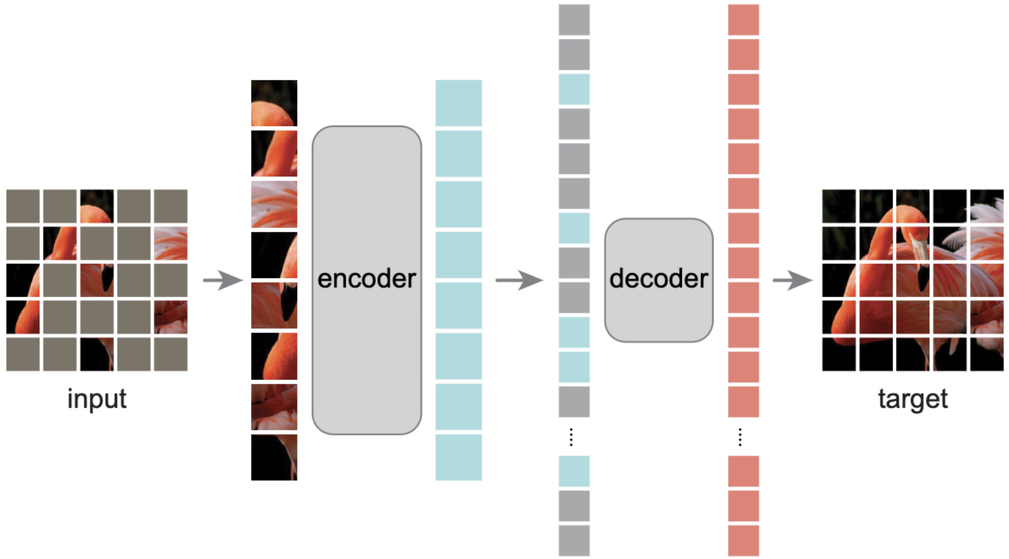
Language Modeling Materializes a World Model of Protein Biology

Team Biohub and EvolutionaryScale

Seungwoo Yoo, KAIST Visual AI Group

https://dvelopery0115.github.io/

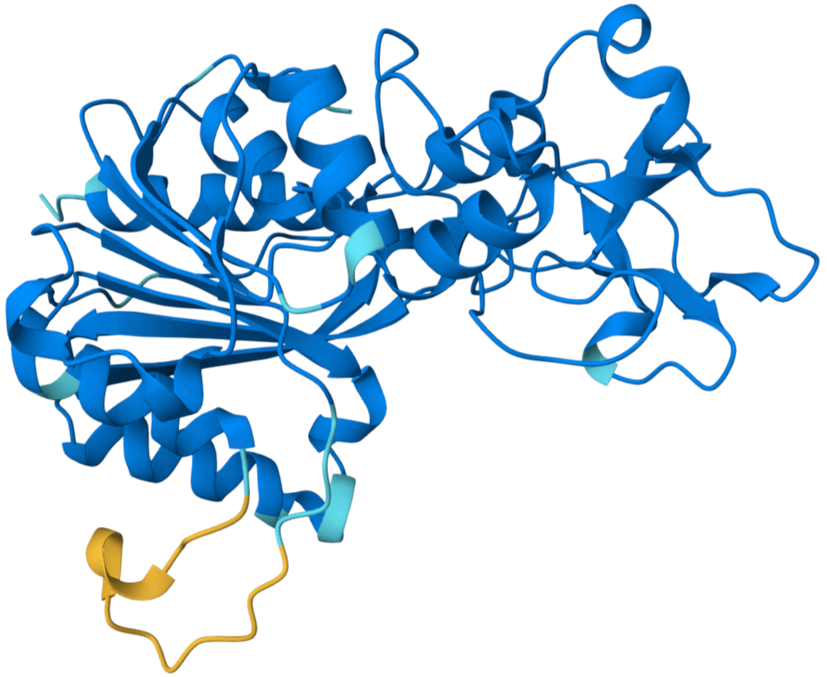




BERT, Devlin et al., NAACL 2019

Masked Autoencoders, He et al., CVPR 2022

“MIEIKDKQLTGLRFIDLFAGLGGFRLALESCGAECVYSNEWDKYAQEVYEMNFG
EKPEGDITQVNEKTIPDHDILCAGFPCQAFSISGKQKGFEDSRGTLFFDIARIV
REKKPKVVFMENVKNFASHDNGNTLEVVKNTMNELDYSFHAKVLNALDYGIPQK
RERIYMICFRNDLNIQNFQFPKPFELNTFVKDLLLPDSEVEHLVIDRKDLVMTN
QEIEQTTPKTVRLGIVGKGGQGERIYSTRGIAITLSAYGGGIFAKTGGYLVNGK
TRKLHPRECARVMGYPDSYKVHPSTSQAYKQFGNSVVINVLQYIAYNIGSSLNF
KPY”



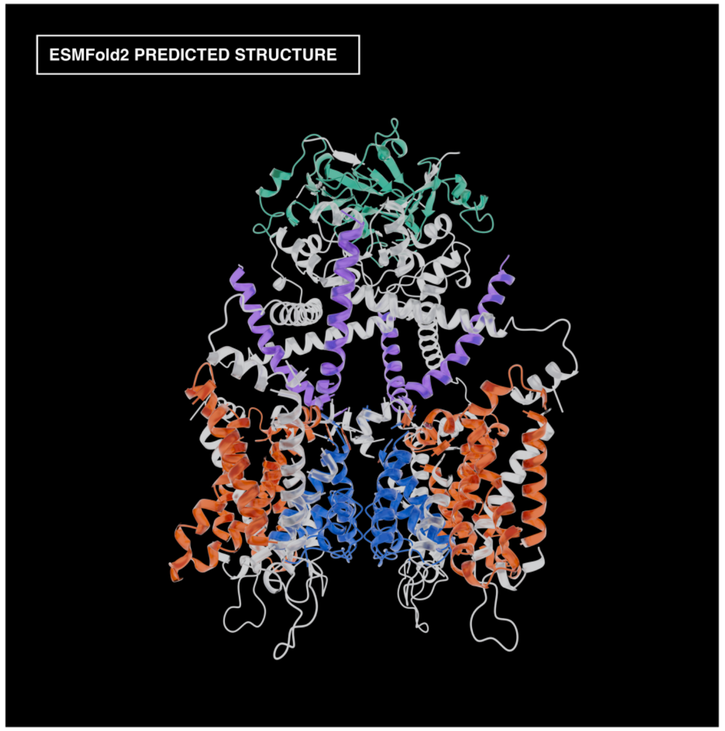
Can we learn a unified protein representation that captures both structure and function from sequence alone?

Overview

PROTEIN SEQUENCE

DALIVLNVSGTRFQTWQDTLE
RYPDTLLGSSERDFFYHPETQ
QYFFDRDPDIFRHILNFYRTG
KLHYPRHECISAYDEELAFFG
LIPEIIGDCCYEEYKDRRREN
AERLQDDADTDTAGESALPTM
TARQRVWRAFENPHTSTMALV
FYYVTGFFIAVSVIANVVETV
PCGSSPGHIKELPCGERYAVA
FFCLDTACVMIFTVEYLLRLA
AAPSRYRFVRSVMSIIDVVAI
LPYYIGLVMTDNEDVSGAFVT
LRVFRVFRIFKFSRHSQGLRI
LGYTLKSCASELGFLLFSLTM
AIIIFATVMFYAEKGSSASKF
TSIPAAFWYTIVTMTTLGYGD
MVPKTIAGKIFGSICSLSGVL
VIALPVPVIVSNFSRIYHQNQ
RADKRRAQKKARLARIRAAK

ESMFold2 PREDICTED STRUCTURE

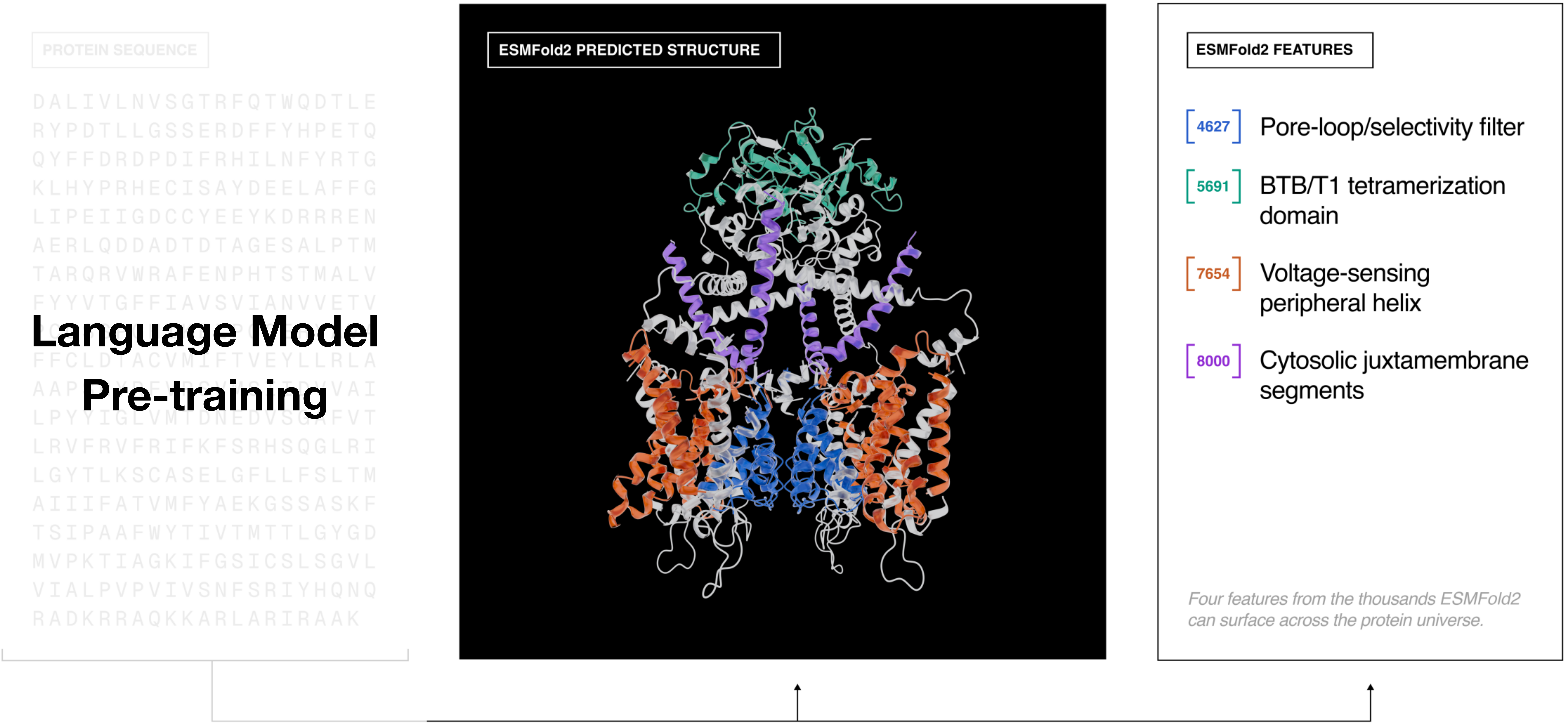


ESMFold2 FEATURES

- [4627] Pore-loop/selectivity filter
- [5691] BTB/T1 tetramerization domain
- [7654] Voltage-sensing peripheral helix
- [8000] Cytosolic juxtamembrane segments

Four features from the thousands ESMFold2 can surface across the protein universe.

Overview


PROTEIN SEQUENCE
DALIVLNVSGTRFQTWQDTLE
RYPDTLLGSSERDFFYHPETQ
QYFFDRDPDIFRHILNFYRTG
KLHYPRHECISAYDEELAFFG
LIPEIIGDCCYEEYKDRRREN
AERLQDDADTDTAGESALPTM
TARQRVWRAFENPHTSTMALV
LRVFRVFRIFKFSRHSQGLRI
LGYTLKSCASELGFLLFSLTM
AIIIFATVMFYAEKGSSASKF
TSIPAAFWYTIVTMTTLGYGD
MVPKTIAGKIFGSICSLSGVL
VIALPVPVIVSNFSRIYHQNQ
RADKRRAQKKARLARIRAAK
Language Model
Pre-training
ESMFold2 PREDICTED STRUCTURE
ESMFold2 FEATURES
4627 Pore-loop/selectivity filter
5691 BTB/T1 tetramerization domain
7654 Voltage-sensing peripheral helix
8000 Cytosolic juxtamembrane segments
Four features from the thousands ESMFold2 can surface across the protein universe.

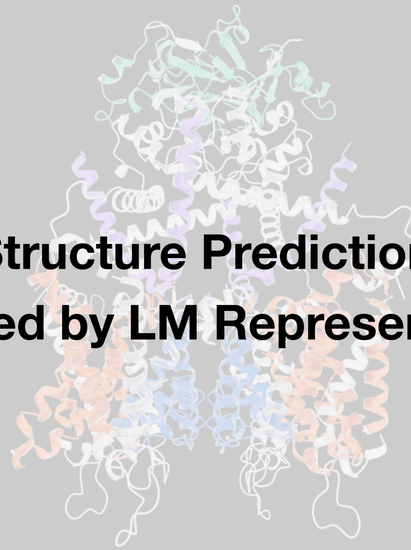
Overview

PROTEIN SEQUENCE

DALIVLNVSGTRFQTWQDTLE
RYPDTLLGSSERDFFYHPETQ
QYFFDRDPDIFRHILNFYRTG
KLHYPRHECISAYDEELAFFG
LIPEIIGDCCYEEYKDRRREN
AERLQDDADTDTAGESALPTM
TARQRVWRAFENPHTSTMALV
FYYVTGFFIAVSVIANVVETV

Language Model Pre-training

ESMFold2 PREDICTED STRUCTURE



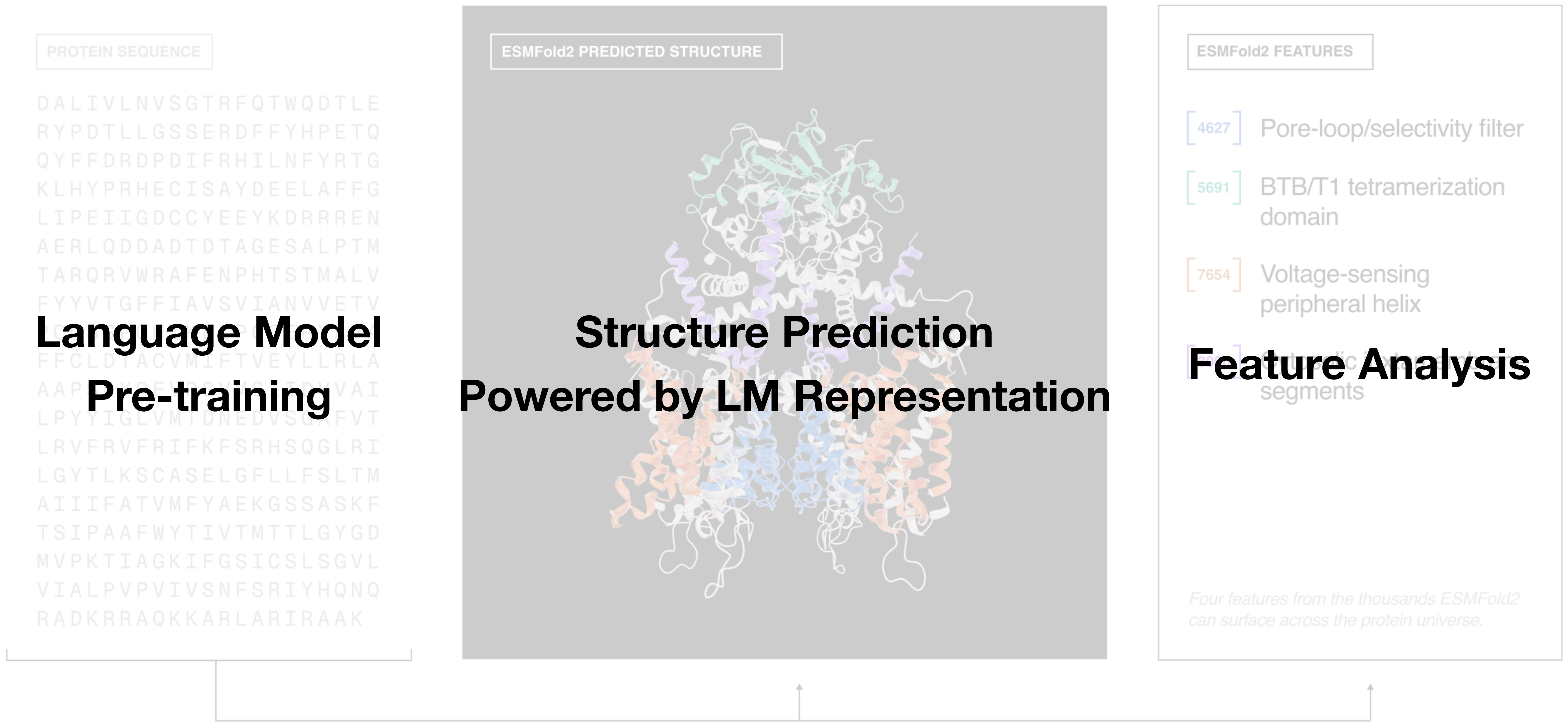
Structure Prediction Powered by LM Representation

ESMFold2 FEATURES

- [4627] Pore-loop/selectivity filter
- [5691] BTB/T1 tetramerization domain
- [7654] Voltage-sensing peripheral helix
- [8000] Cytosolic juxtamembrane segments

Four features from the thousands ESMFold2 can surface across the protein universe.

Overview

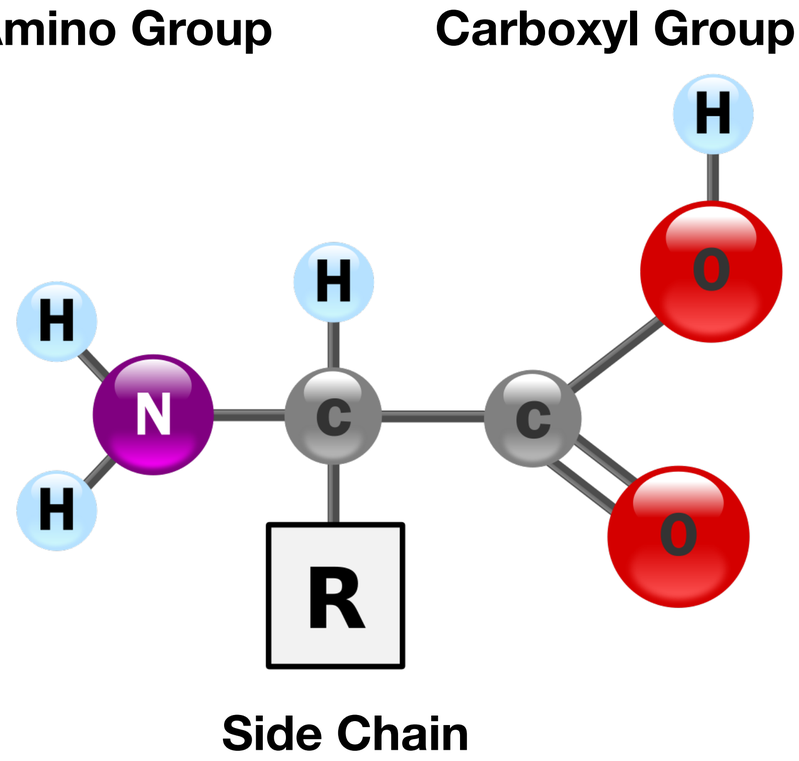


PROTEIN SEQUENCE
Language Model
Pre-training
ESMFold2 PREDICTED STRUCTURE
Structure Prediction
Powered by LM Representation
ESMFold2 FEATURES
4627 Pore-loop/selectivity filter
5691 BTB/T1 tetramerization domain
7654 Voltage-sensing peripheral helix
Feature Analysis
Four features from the thousands ESMFold2 can surface across the protein universe.

Preliminaries

What is a Protein?

A protein consists of **amino acids (AAs)**, each of which contains a unique side chain that determines its chemical properties and functional role.



An Amino Acid Molecule

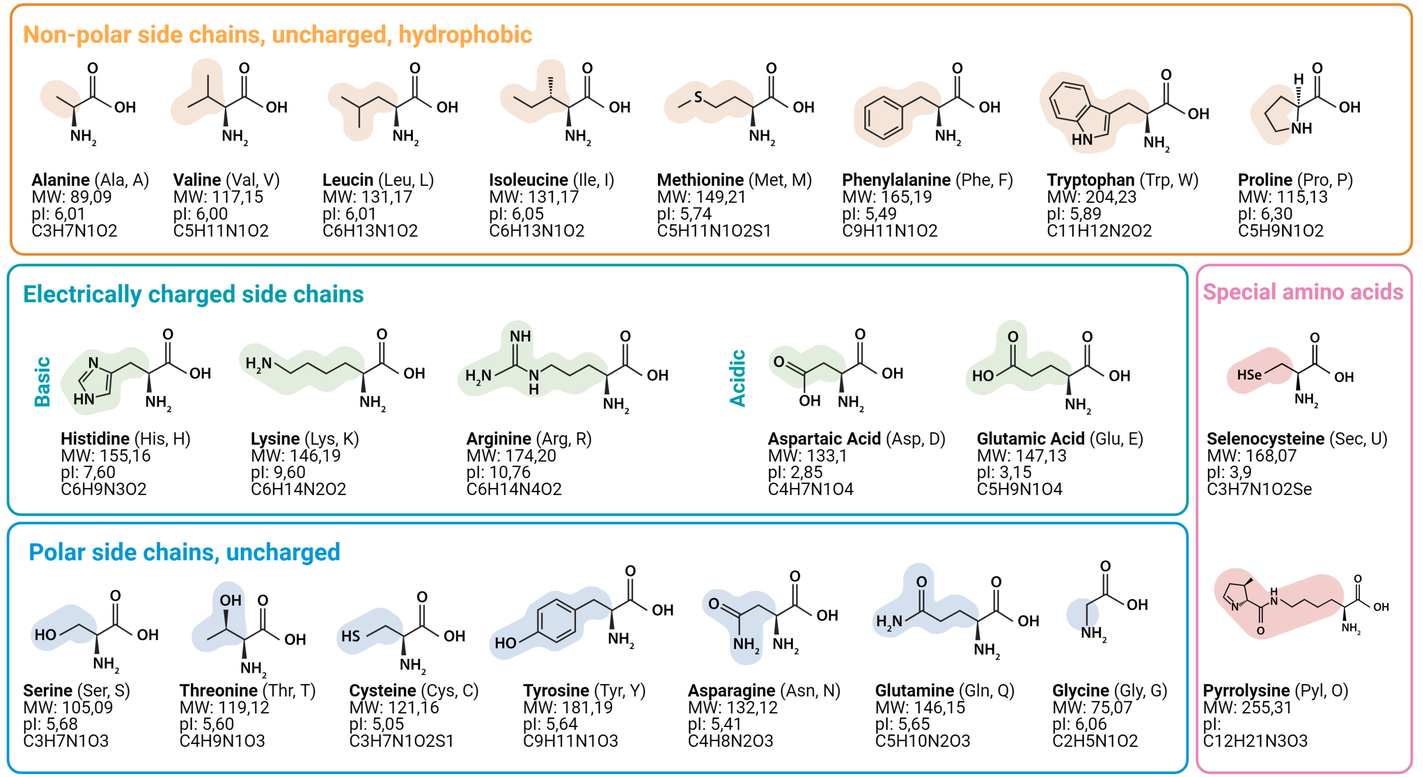
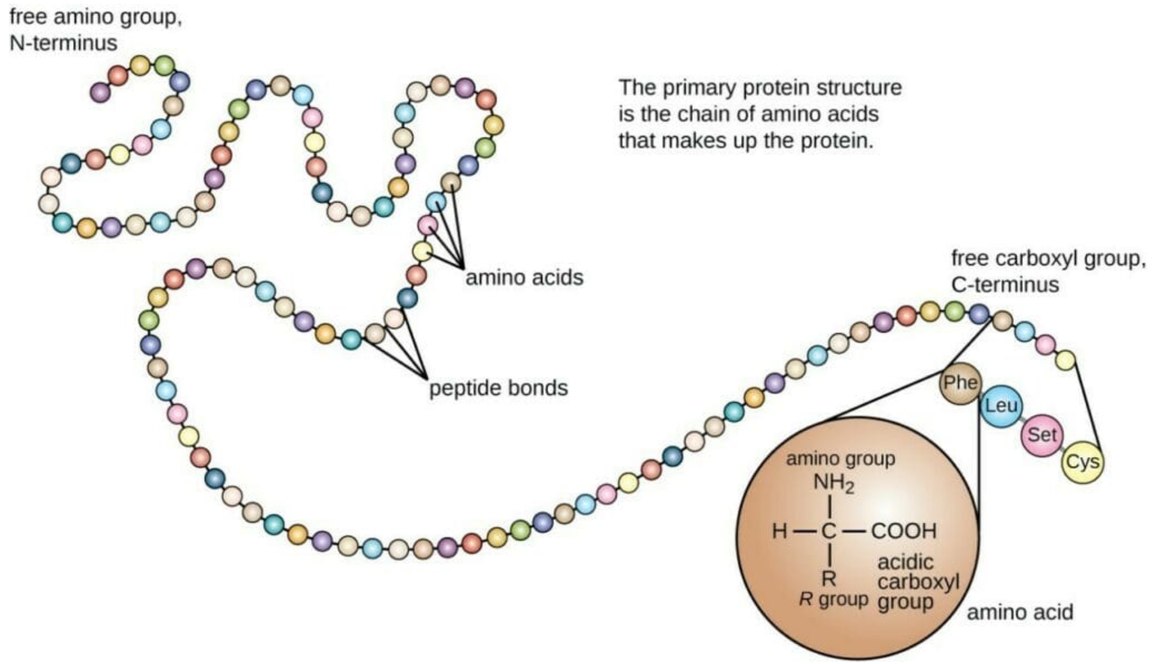


Table of Side Chains of 22 Amino Acids

Preliminaries

What is a Protein?

A protein begins its life as **a chain of AAs** linked by peptide bonds, formed between the carboxyl group of one AA and the amino group of the next.

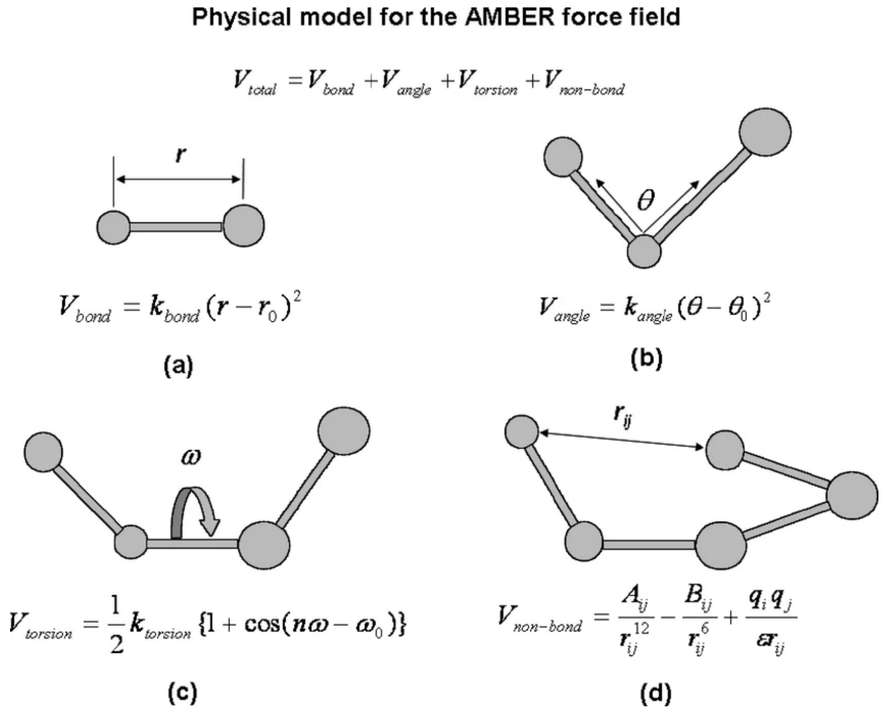


How to Achieve Protein Sequencing?

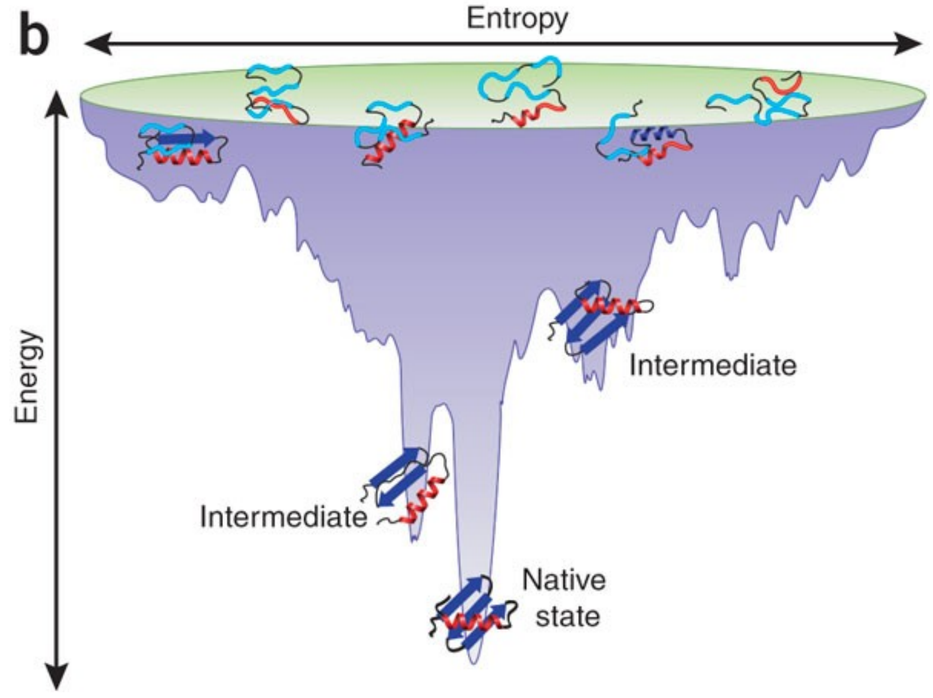
Preliminaries

What is a Protein?

The atoms within AAs constantly interact with one another, driving the protein to fold into a stable **three-dimensional structure** that minimizes its free energy.



Energy Functions of AMBER Force Field

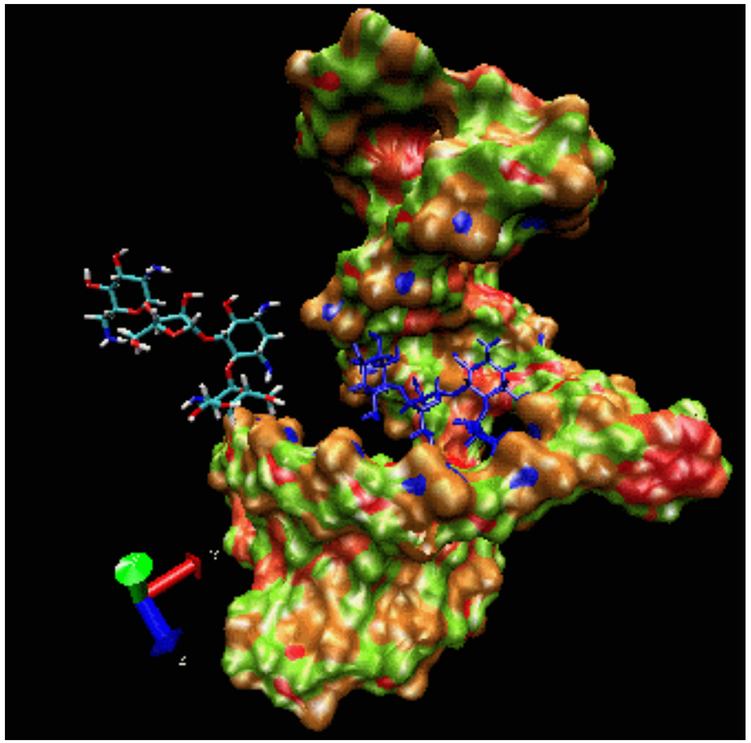


Free Energy Landscape of a Protein

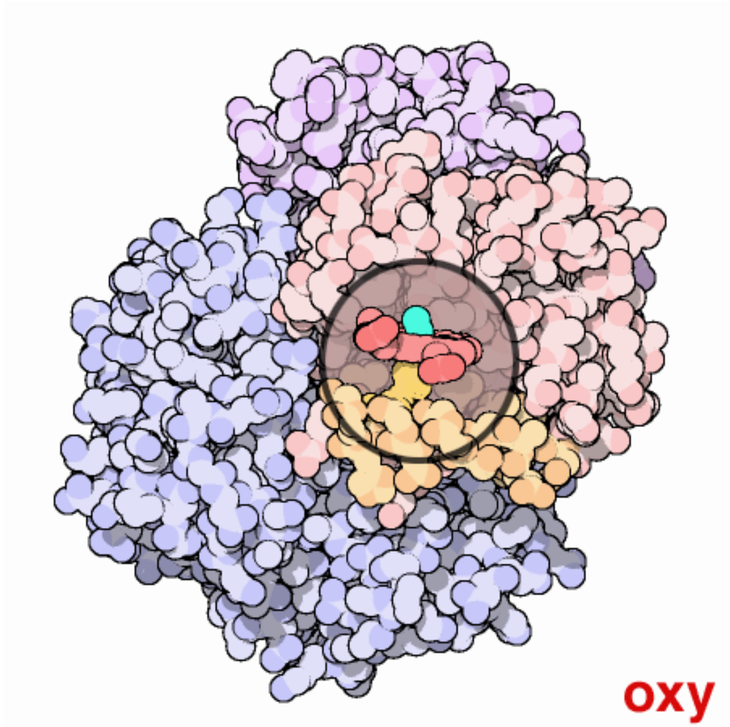
Preliminaries

What is a Protein?

The three-dimensional (tertiary) structure of a protein plays a key role in determining its function.



Flexible docking of a ligand to a protein



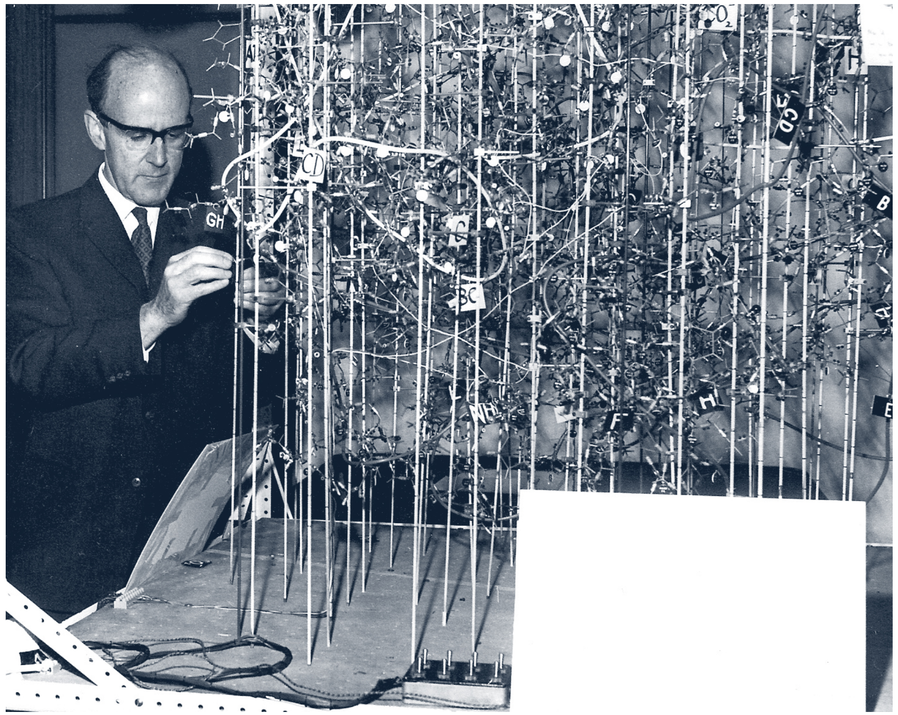
Hemoglobin conformational changes upon oxygen binding and release

Related Work

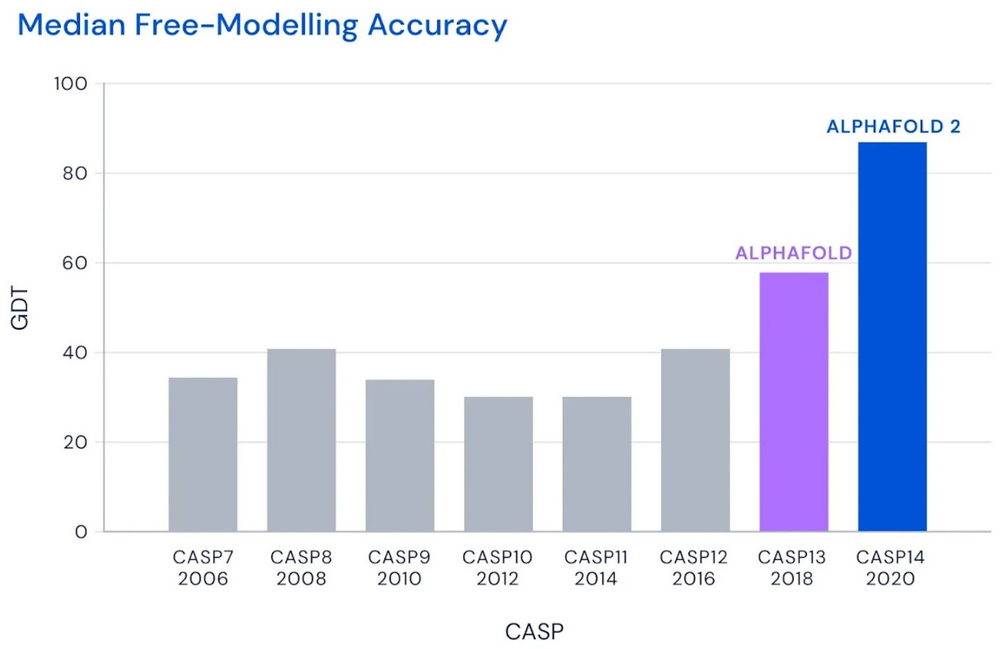
Protein Structure Prediction

Nature folds an AA chain into a functional 3D structure within milliseconds.

Yet predicting that structure remained an unsolved problem until AlphaFold.



Max F. Perutz (Nobel Laureate, Chem. 1962) with his first high-resolution model of haemoglobin.

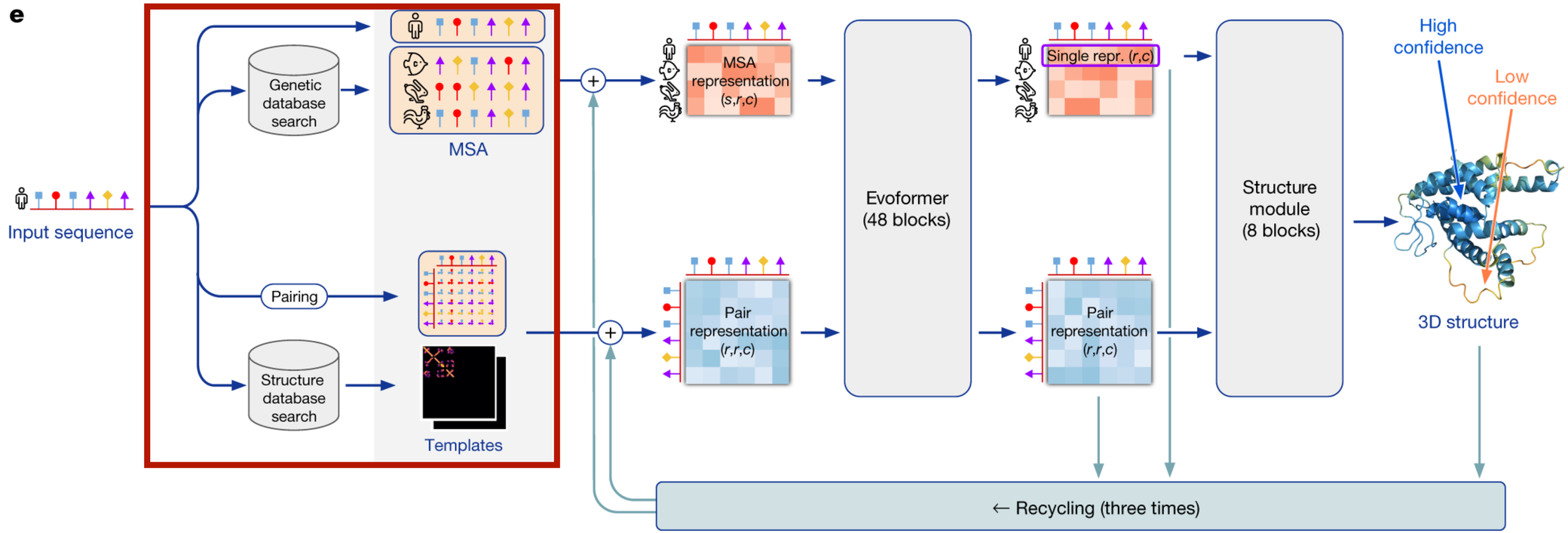


Median of prediction accuracies from the best performing teams in CASP (2006—2020)

Related Work

Protein Structure Prediction

AlphaFold and AlphaFold2 were designed to predict the structure of individual proteins using **evolutionary signals** derived from sequence and structural data.

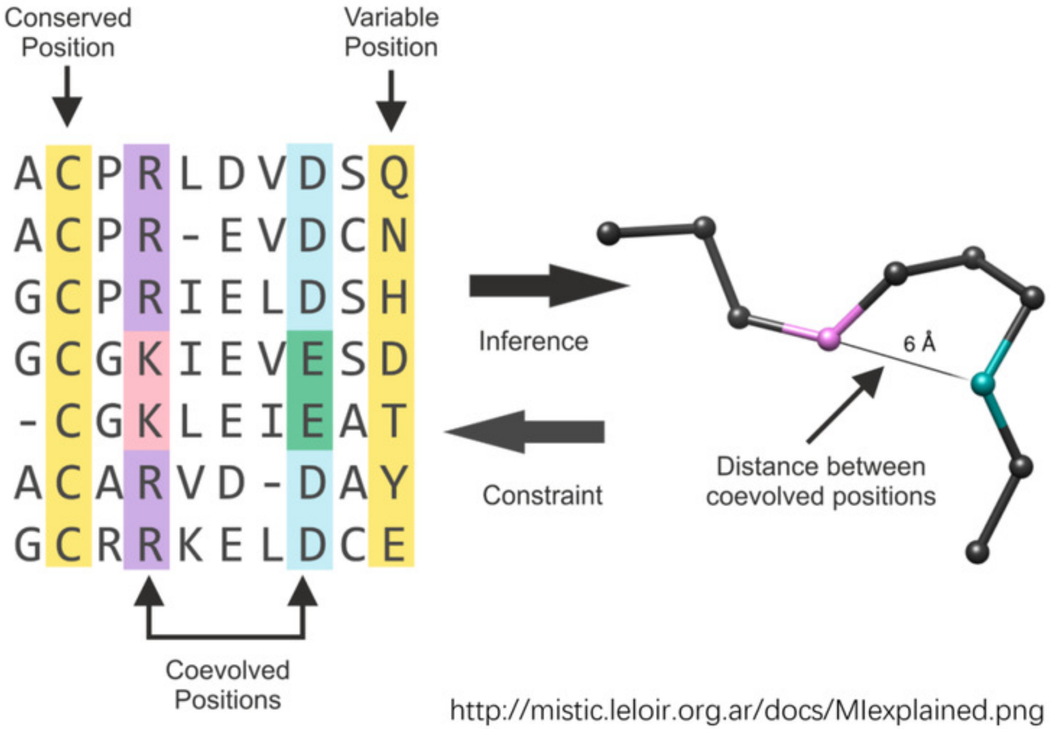


AlphaFold 2 Network Architecture

Related Work

Protein Structure Prediction

AlphaFold relies on **Multiple Sequence Alignment (MSA)**, a technique that identifies evolutionarily related sequences and captures co-evolutionary signals.



http://mistic.leloir.org.ar/docs/MIexplained.png

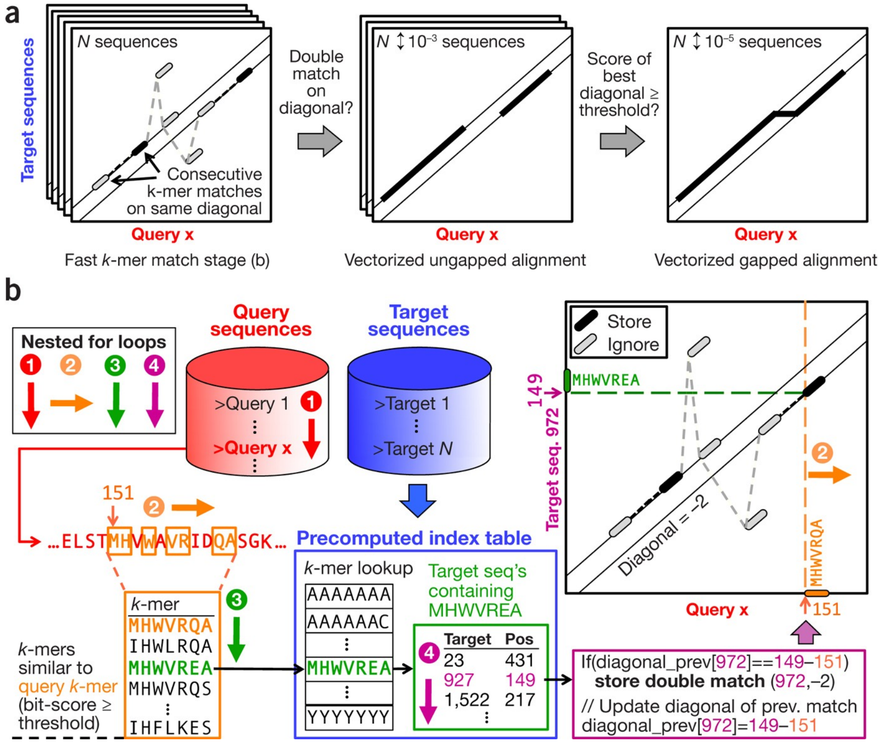
Conserved patterns may imply functional sites.

Coevolved patterns help maintain protein structure.

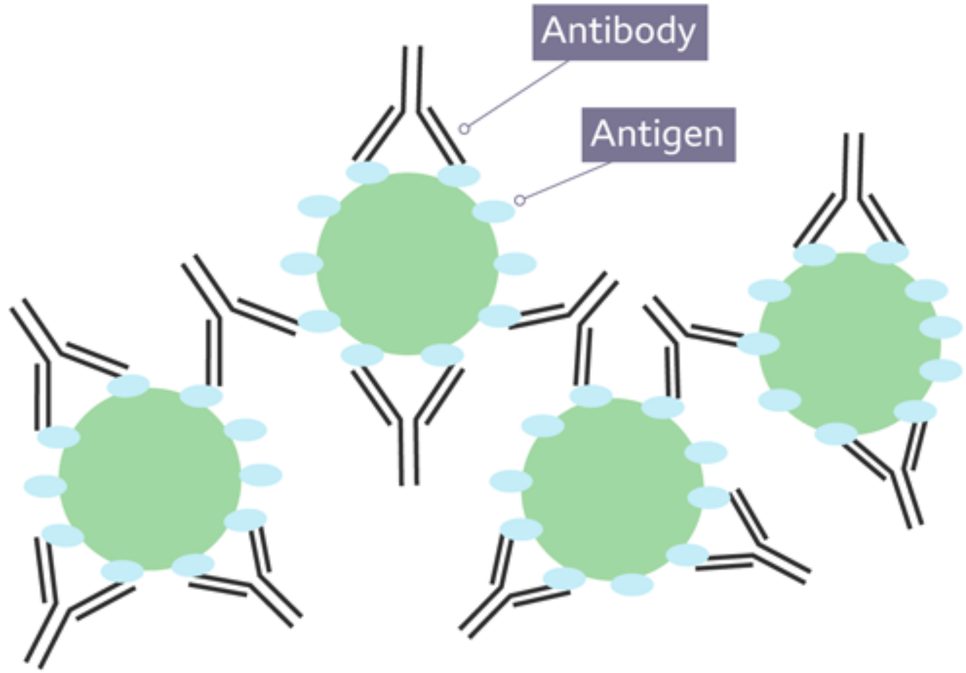
Related Work

Protein Structure Prediction

Despite their success, MSA-based approaches have several limitations.



High Computational Cost

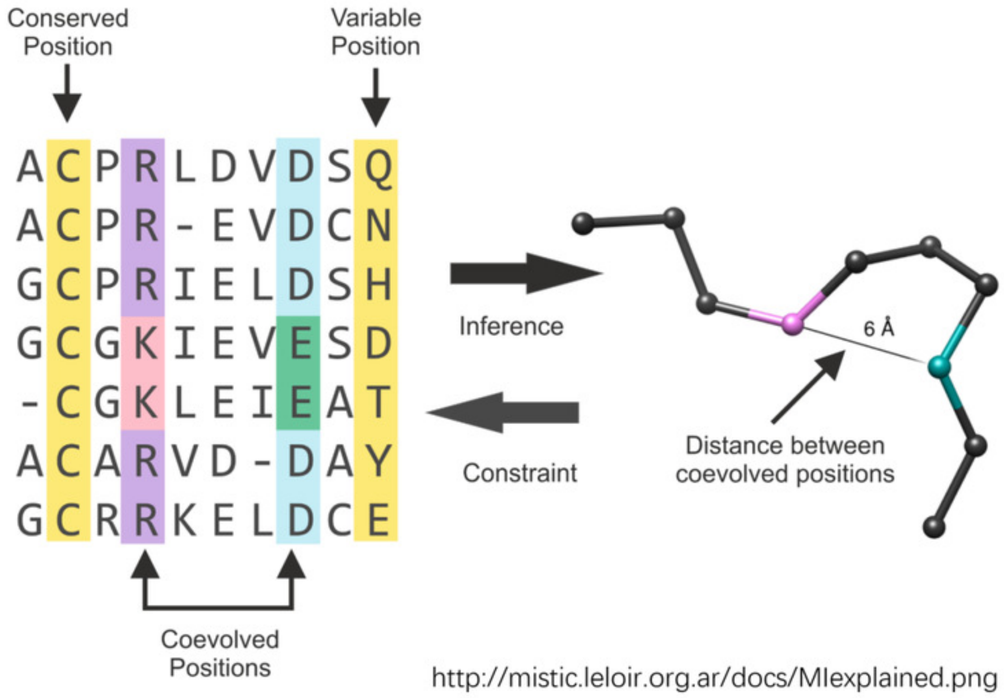


Reduced Performance without Co-evolutionary Signals

Related Work

Protein Structure Prediction

MSA can be viewed as a form of **explicit evolutionary feature engineering**, similar to hand-crafted descriptors used in early computer vision.



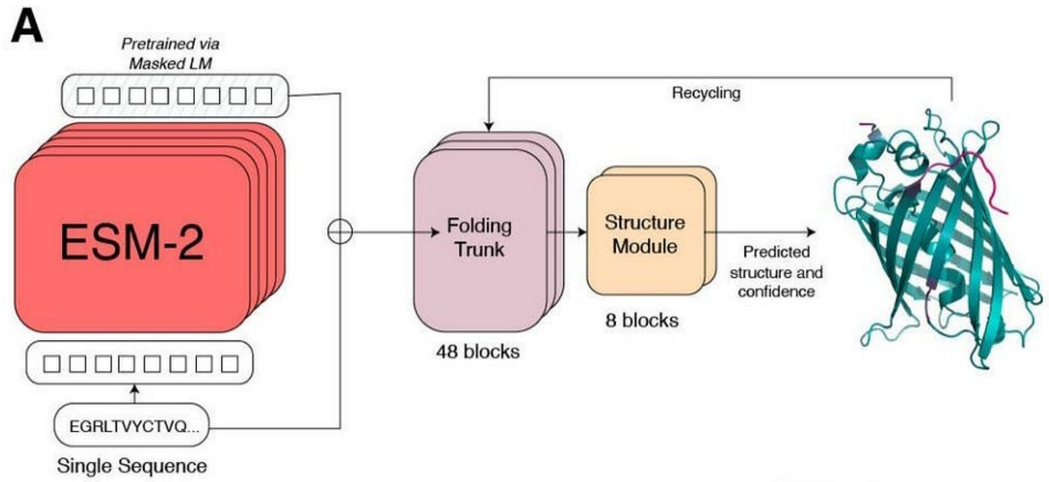
SIFT, Lowe, IJCV 2004

Related Work

Protein Language Models (PLMs)

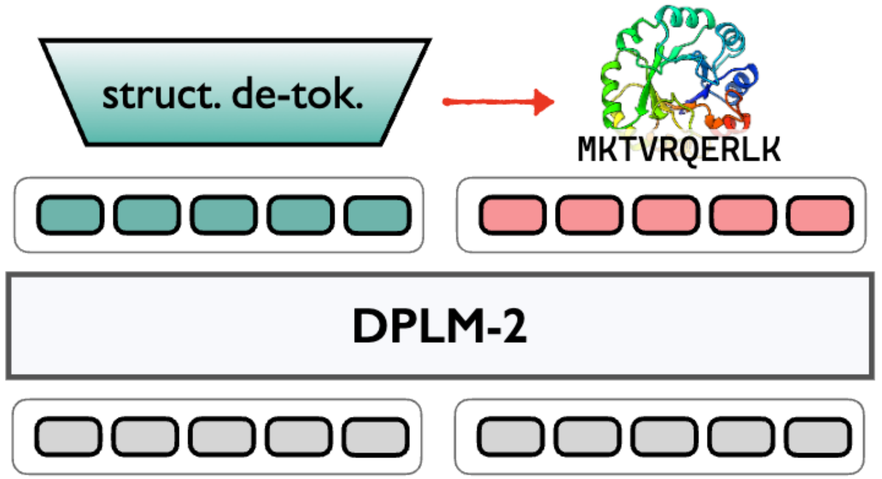
Like LLMs acquire knowledge from large training corpora, protein language models (PLMs) **learn useful representations through self-supervised learning**.

Structure Prediction



ESM-2, Lin et al., Science 2023

Protein Generation

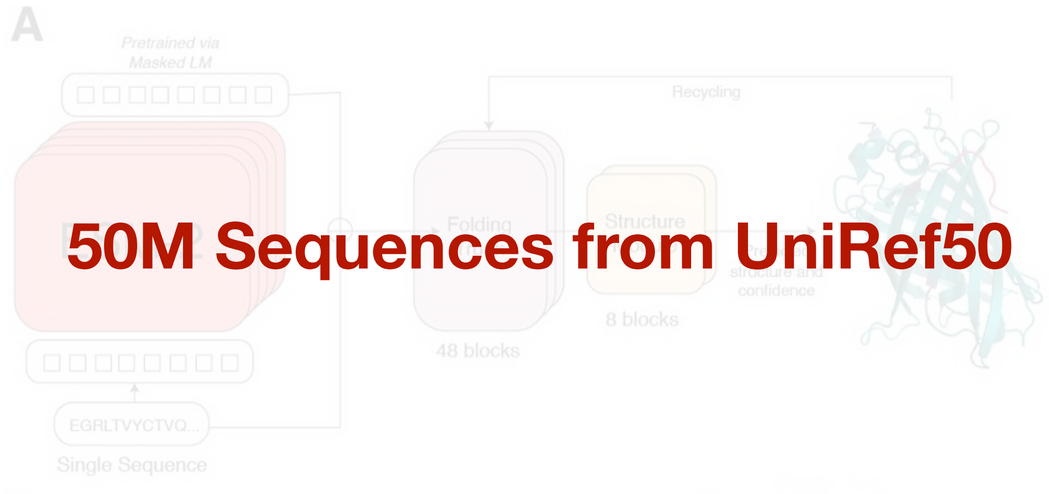


DPLM-2, Wang et al., ICLR 2024

Related Work

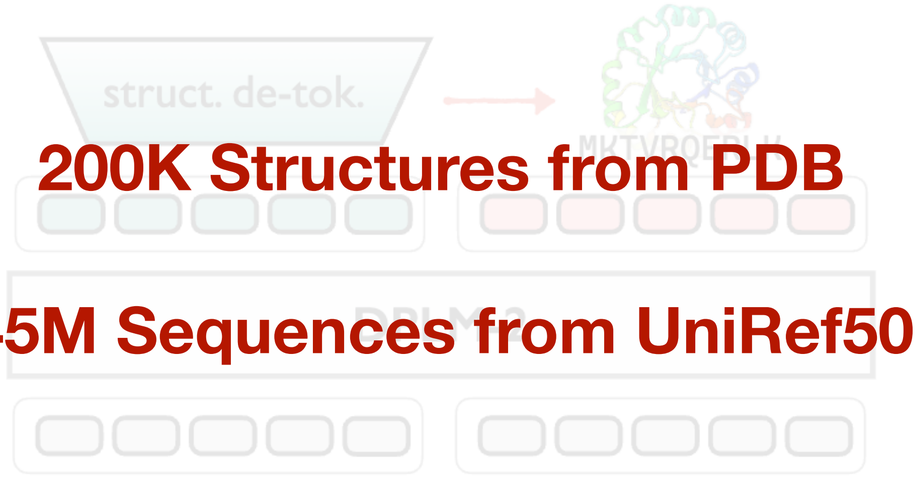
Protein Language Models (PLMs)

Like LLMs acquire knowledge from large training corpora, protein language models (PLMs) **learn useful representations through self-supervised learning**.



50M Sequences from UniRef50

ESM-2, Lin et al., Science 2023



200K Structures from PDB

45M Sequences from UniRef50

DPLM-2, Wang et al., ICLR 2024

Have we truly observed scaling laws in PLMs?

ESM Cambrian (ESMC)

Learning Protein Representations via Masked Language Modeling

ESM Cambrian (ESMC) is a family of transformer models trained on protein sequences by optimizing the masked language modeling objective:

$$\mathcal{L} = \mathbb{E}_{x,M}[- \sum_{i \in M} \log p(x_i \,|\, x_{\setminus M})],$$

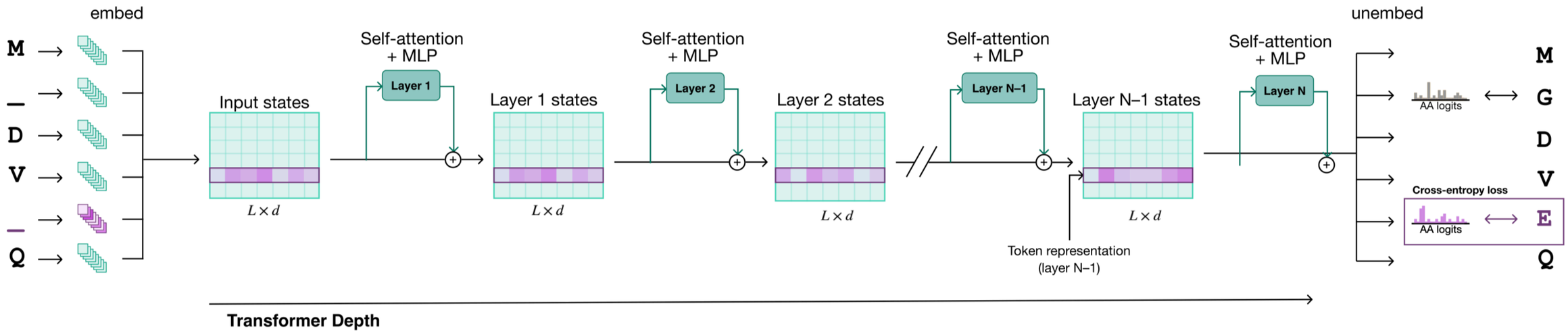
letting the model to predict amino acid types x_i at randomly masked positions $i \in M$ using the context $x_{\setminus M}$.

ESM Cambrian (ESMC)

Learning Protein Representations via Masked Language Modeling

ESMC family comes with three variants: **300M**, **600M**, and **6B**.

Each model consists of 16, 24, and 80 transformer layers, respectively.



ESMC Network Architecture

ESM Cambrian (ESMC)

Learning Protein Representations via Masked Language Modeling

All models were trained on a large-scale dataset composed of three sources:

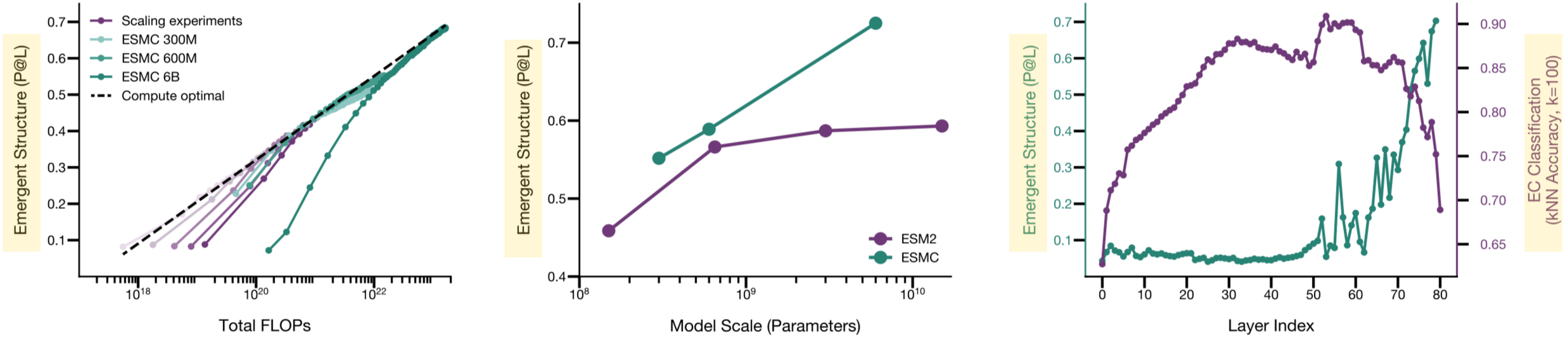
1. **UniRef (~156M Sequences)**
2. **MGnify (~621M Sequences)**
3. **Joint Genome Institute (JGI) Metagenome Database (2B Sequences)**

In total, the pretraining dataset contains **2.8 billion sequences**, approximately **56× larger** than the dataset used to train ESM-2.

ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

With the increased dataset size, ESMC exhibits log-linear scaling laws with respect to (1) training FLOPs and (2) model size.



Scaling Laws and Layer-wise Emergent Properties

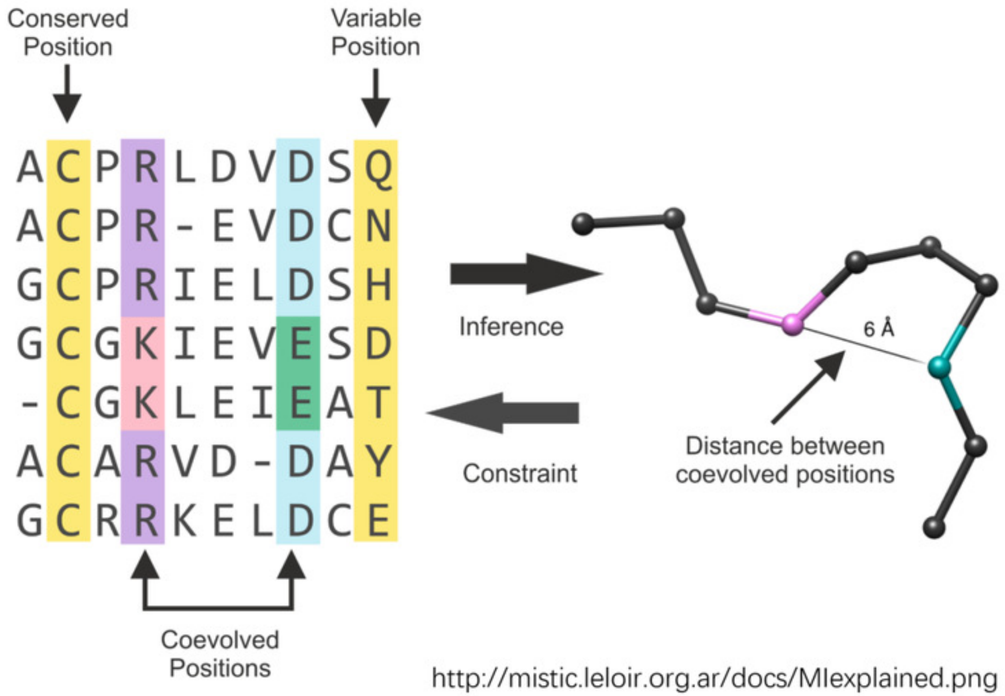
Structural and functional features emerge during pretraining!

ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

Multiple Sequence Alignment (MSA) can identify co-evolving, yet separated residue pairs that are important for maintaining protein structure.

→ **Do ESMC representations encode similar information?**



http://mistic.leloir.org.ar/docs/MIexplained.png

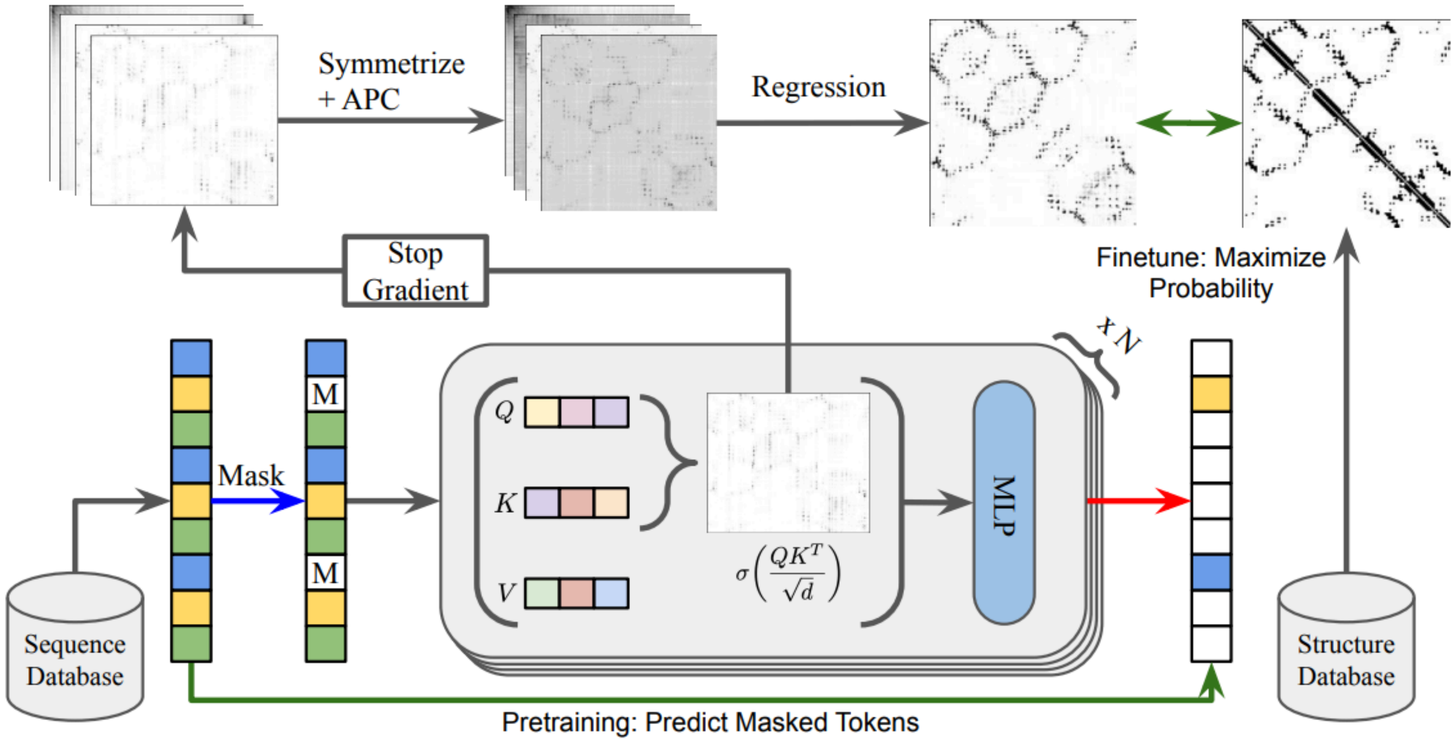
Conserved sequence patterns can reveal functional sites.

Evolutionarily conserved patterns help maintain protein structure.

ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

After pretraining, a logistic regression probe is trained to predict long-range residue contacts (|i – j| ≥ 24).



20

Training Examples

20,775

Test Examples

Transformer protein language models are unsupervised structure learners, Rao et al., ICLR 2021

ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

Representation quality is assessed by how accurately a logistic regression probe predicts long-range contacts.

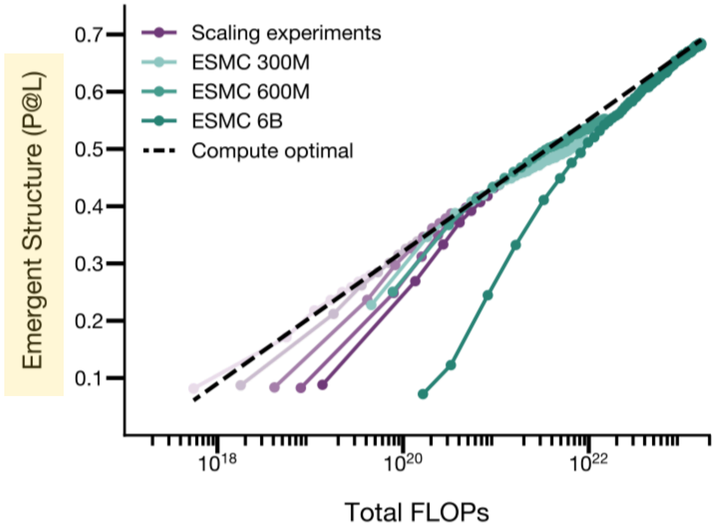
For each protein of length L, we

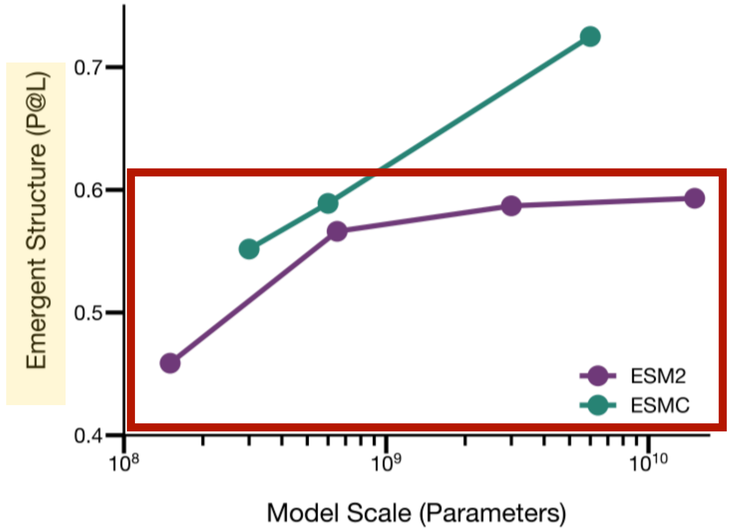
1. Extract ESMC attention maps computed from the sequence;
2. Predict contact probabilities using the trained logistic regression model;
3. Compute the precision of the L most confident predictions as *P@L*.

ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

With the fixed dataset size (2.8B), the precision (*P@L*) increases with the total training FLOPs and parameter counts.



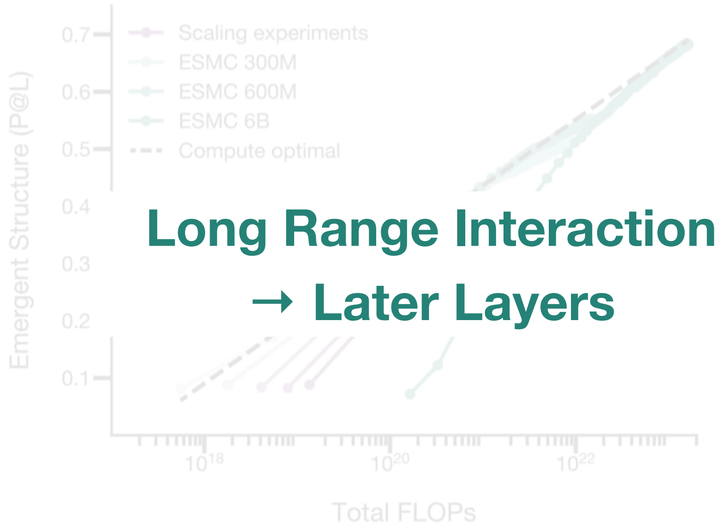


ESM2 performance plateaus due to the limited size of its pretraining dataset (~50M sequences from UniRef50).

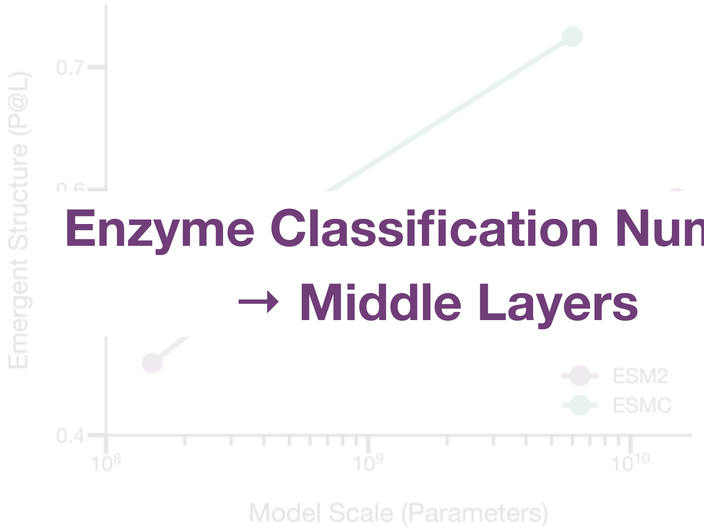
ESM Cambrian (ESMC)

Scaling Law and Emergence of Properties

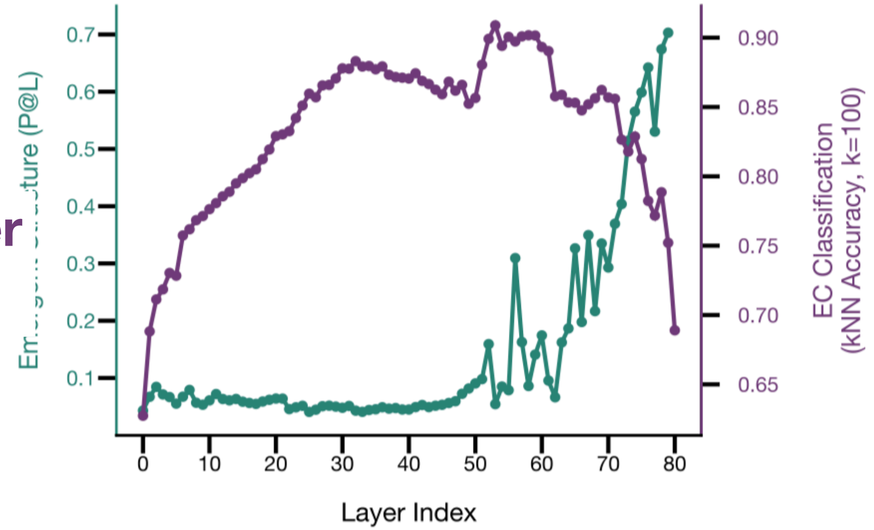
Furthermore, different layers in ESMC-6B (80 in total) encode distinct structural and functional information.



Long Range Interaction
→ Later Layers



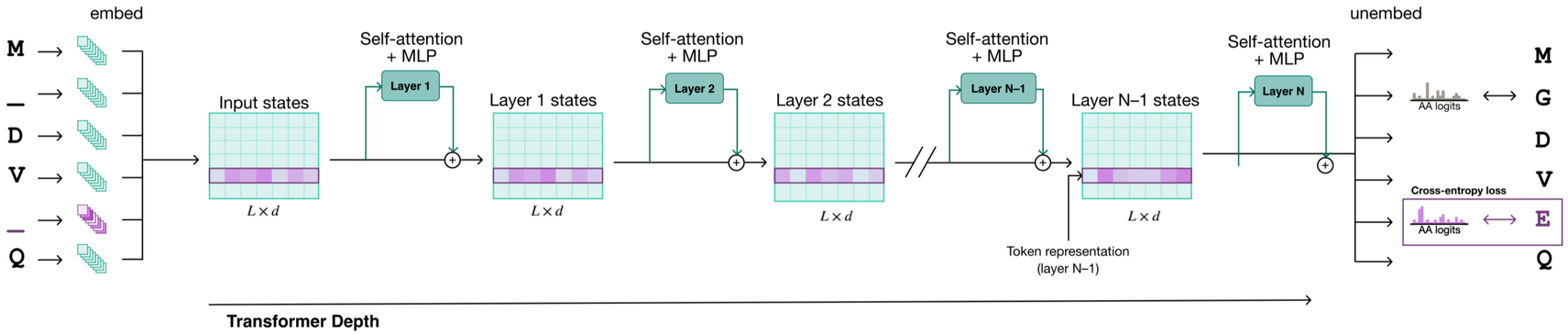
Enzyme Classification Number
→ Middle Layers



ESM Cambrian (ESMC)

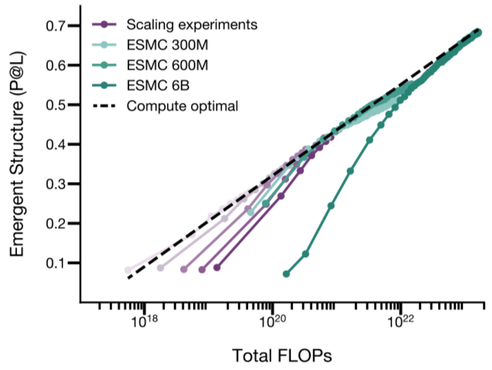
Scaling Law and Emergence of Properties

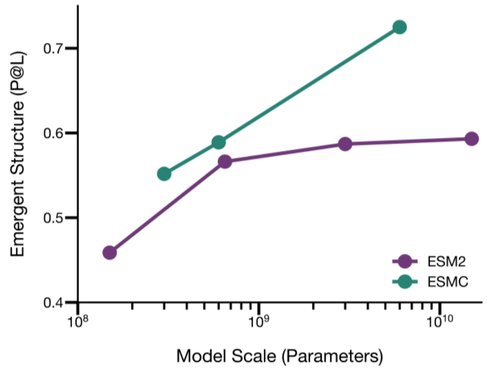
Key Takeaway: Masked language modeling on protein sequences yields rich representations for predicting protein structure and function.

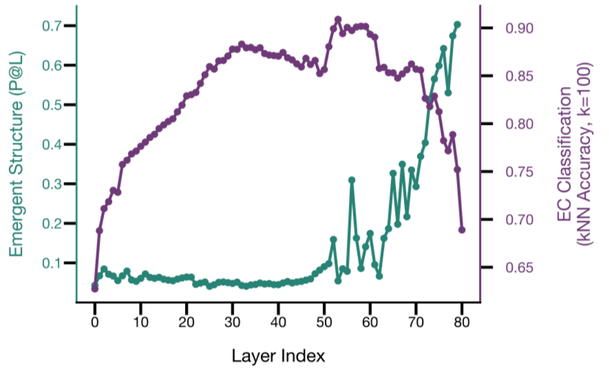


Large-Scale Pretraining

Emergent Properties!



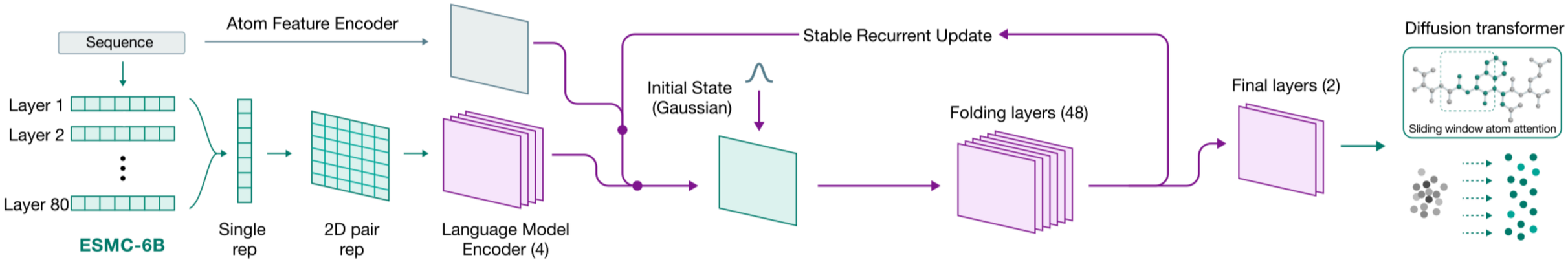




ESMFold 2

A New SotA for Protein Complex Structure Prediction

ESMFold 2 is a protein folding model parameterized as a **looped transformer**, taking **ESMC-6B representations** as primary inputs (not MSA).

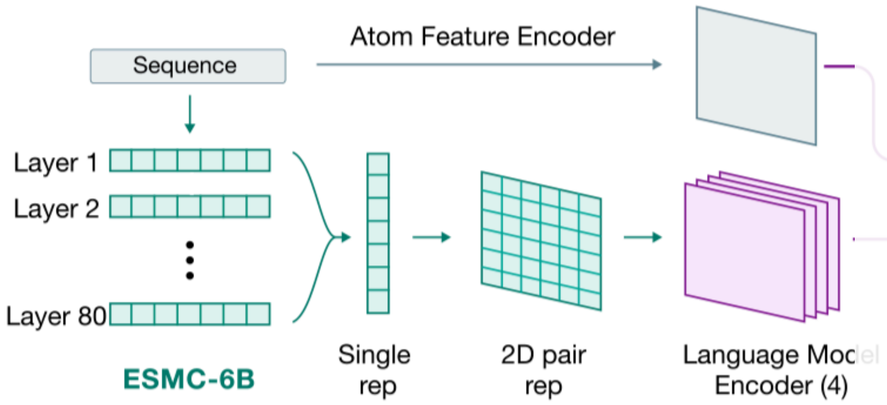


ESMFold 2 Network Architecture

ESMFold 2

A New SotA for Protein Complex Structure Prediction

The forward pass of ESMFold 2 begins by embedding the input sequence into ESMC-6B's embedding space.



Algorithm 2 ESMC Representation Integration.

Require: $\{\mathbf{x}_c\}$
Ensure: $\mathbf{z}_{\text{lm}}$

for each chain c **do**
  $\mathbf{h}_c \leftarrow \text{ESMC}([\text{BOS}] \,\|\, \mathbf{x}_c \,\|\, [\text{EOS}])$
end for
$\mathbf{h} \leftarrow \text{CropAndConcat}(\{\mathbf{h}_c\})$
$\mathbf{h}_{\text{proj}} \leftarrow \text{Linear}(\text{LayerNorm}(\mathbf{h}),\, d_z)$
$\boldsymbol{\alpha} \leftarrow \text{Softmax}(\mathbf{w}_z)$
$\mathbf{h}_{\text{combined}} \leftarrow \sum_k \alpha_k \cdot \mathbf{h}_{\text{proj}}[:, :, k, :]$
$\mathbf{z}_{\text{lm}} \leftarrow \text{MLP}(\text{OuterSum}(\mathbf{h}_{\text{combined}}))$
$\mathbf{z}_{\text{lm}} \leftarrow \text{PairFoldingLayers}_{\text{lm}}(\text{Linear}(\mathbf{z}_{\text{lm}}))$
return $\mathbf{z}_{\text{lm}}$

Layer-wise Features: $[B, L_{\text{crop}}, N_{\text{layers}} + 1, D_{\text{in}}]$

Projected LwF: $[B, L_{\text{crop}}, N_{\text{layers}} + 1, d_z]$

Learnable Layer Weights: $[N_{\text{layers}} + 1]$

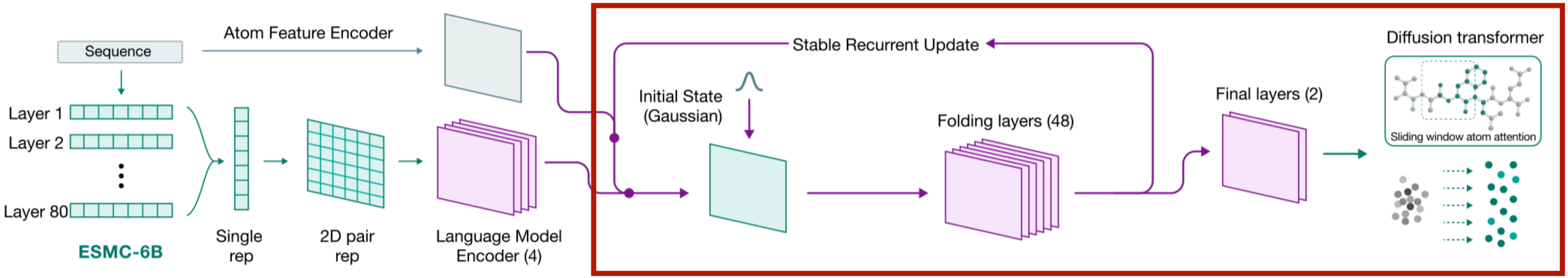
Single Representation: $[B, L_{\text{crop}}, d_z]$

Pair Representation: $[B, L_{\text{crop}}, L_{\text{crop}}, d_z]$

ESMFold 2

A New SotA for Protein Complex Structure Prediction

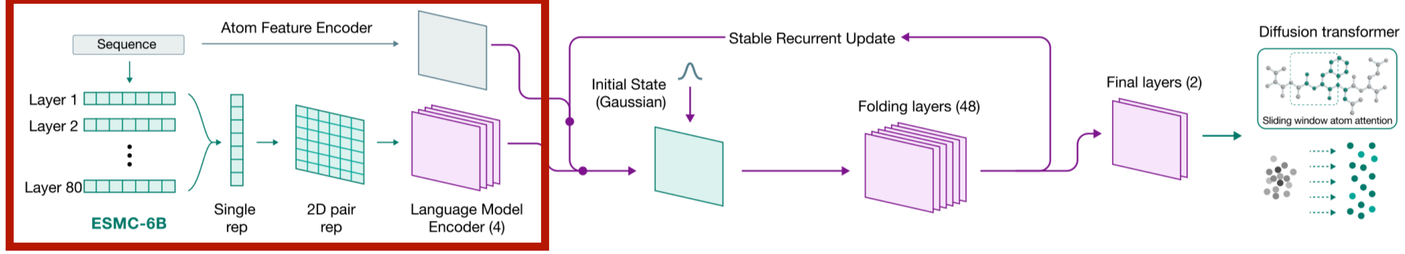
Combined with other embeddings (*e.g.*, atom features), the sequence embeddings are processed by a looped transformer, followed by a DiT.



ESMFold 2 Network Architecture

ESMFold 2

A New SotA for Protein Complex Structure Prediction



Algorithm 1 ESMFold2 Forward Pass.

Require: $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}}, \mathbf{f}_{\text{msa}}, \mathbf{f}_{\text{bond}}, \{\mathbf{x}_c\}, T$
Ensure: $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

$\mathbf{f}_{\text{inputs}} \leftarrow \text{InputEmbedding}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}})$ ▷A.2.3
$\mathbf{f}_{\text{rel\_pos}} \leftarrow \text{RelPosEnc}$
$\mathbf{z}_{\text{feat}} \leftarrow \text{OuterSum}(\mathbf{f}_{\text{inputs}}) + \mathbf{f}_{\text{rel\_pos}} + \mathbf{f}_{\text{bond}}$ ▷Initial pair embedding
$\mathbf{z}_{\text{lm}} \leftarrow \text{ESMCRepresentation}(\{\mathbf{x}_c\})$ ▷Algorithm 2
$\mathbf{z}_0 \sim \text{trunc\_norm}(0, \frac{2}{5d_{\text{pair}}})$ ▷Initial recurrent state, $\pm 3\sigma$ truncation
for $t = 0$ to $T - 1$ **do**
  $\mathbf{u}_t \leftarrow \mathbf{z}_{\text{feat}}$
  if MSA enabled **then**
    $\hat{\mathbf{f}}_{\text{msa}} \leftarrow \text{MSASubsample}(\mathbf{f}_{\text{msa}})$ ▷Resample MSA rows per loop iteration; A.2.4
    $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{MSAEncoder}(\mathbf{e}_t, \mathbf{f}_{\text{inputs}}, \hat{\mathbf{f}}_{\text{msa}})$ ▷A.2.4
  end if
  $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{LMEncoder}(\text{Dropout}(\mathbf{z}_{\text{lm}}, p{=}0.25))$ ▷Per-loop dropout
  $\mathbf{z}_{t+1} \leftarrow \text{PairFoldingLayers}(\bar{\mathbf{A}} \odot \mathbf{z}_t + \bar{\mathbf{B}}\, \text{LayerNorm}(\mathbf{u}_t))$ ▷Recurrent step; A.2.5
end for
$\mathbf{z} \leftarrow \text{PairFoldingLayers}_{[2]}(\text{Linear}(\mathbf{z}_T))$ ▷2 layers; Refine iteration output
$\mathbf{z}_{\text{disto}} \leftarrow \text{DistogramHead}(\mathbf{z} + \mathbf{z}^\top)$
$\mathbf{x}_{\text{pred}} \leftarrow \text{TruncatedDiffusionSampling}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{f}_{\text{rel\_pos}})$ ▷Algorithm 4
$\mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}} \leftarrow \text{ConfidenceHead}(\mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{x}_{\text{pred}}, \mathbf{f}_{\text{rel\_pos}}, \mathbf{f}_{\text{bond}})$ ▷Algorithm 10
return $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

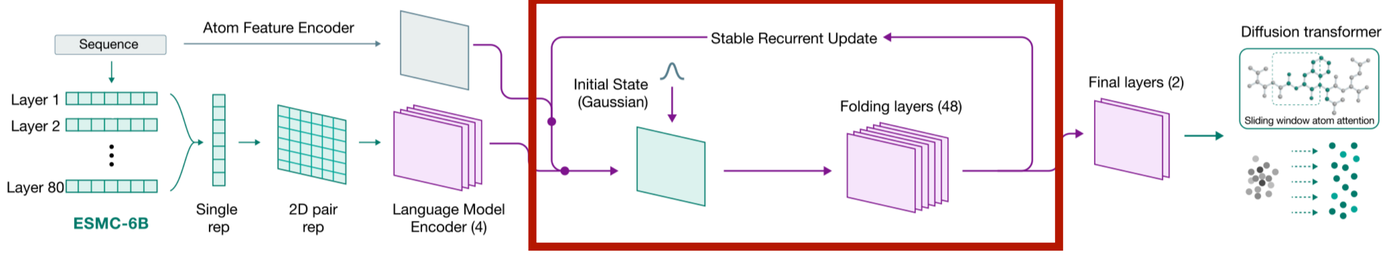
Algorithm 2 ESMC Representation Integration.

Require: $\{\mathbf{x}_c\}$
Ensure: $\mathbf{z}_{\text{lm}}$

for each chain c **do**
  $\mathbf{h}_c \leftarrow \text{ESMC}([\text{BOS}] \,\|\, \mathbf{x}_c \,\|\, [\text{EOS}])$
end for
$\mathbf{h} \leftarrow \text{CropAndConcat}(\{\mathbf{h}_c\})$
$\mathbf{h}_{\text{proj}} \leftarrow \text{Linear}(\text{LayerNorm}(\mathbf{h}),\, d_z)$
$\boldsymbol{\alpha} \leftarrow \text{Softmax}(\mathbf{w}_z)$
$\mathbf{h}_{\text{combined}} \leftarrow \sum_k \alpha_k \cdot \mathbf{h}_{\text{proj}}[:, :, k, :]$
$\mathbf{z}_{\text{lm}} \leftarrow \text{MLP}(\text{OuterSum}(\mathbf{h}_{\text{combined}}))$
$\mathbf{z}_{\text{lm}} \leftarrow \text{PairFoldingLayers}_{\text{lm}}(\text{Linear}(\mathbf{z}_{\text{lm}}))$
return $\mathbf{z}_{\text{lm}}$

ESMFold 2

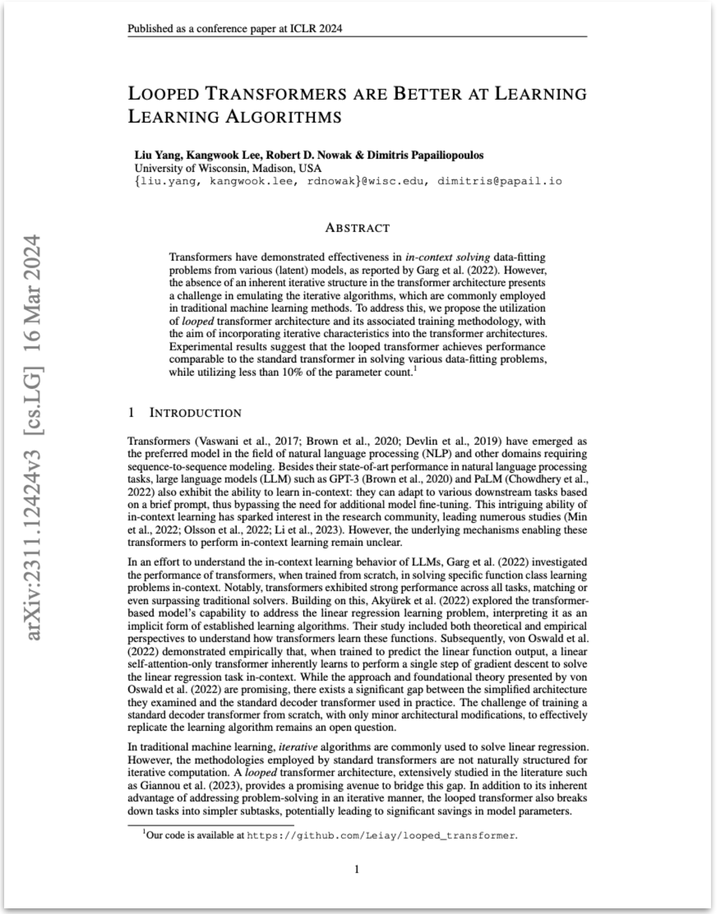
A New SotA for Protein Complex Structure Prediction



Algorithm 1 ESMFold2 Forward Pass.

Require: $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}}, \mathbf{f}_{\text{msa}}, \mathbf{f}_{\text{bond}}, \{\mathbf{x}_c\}, T$
Ensure: $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

$\mathbf{f}_{\text{inputs}} \leftarrow \text{InputEmbedding}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}})$ ▷A.2.3
$\mathbf{f}_{\text{rel\_pos}} \leftarrow \text{RelPosEnc}$
$\mathbf{z}_{\text{feat}} \leftarrow \text{OuterSum}(\mathbf{f}_{\text{inputs}}) + \mathbf{f}_{\text{rel\_pos}} + \mathbf{f}_{\text{bond}}$ ▷Initial pair embedding
$\mathbf{z}_{\text{lm}} \leftarrow \text{ESMCRepresentation}(\{\mathbf{x}_c\})$ ▷Algorithm 2
$\mathbf{z}_0 \sim \text{trunc\_norm}(0, \frac{2}{5d_{\text{pair}}})$ ▷Initial recurrent state, $\pm 3\sigma$ truncation
for $t = 0$ to $T - 1$ **do**
  $\mathbf{u}_t \leftarrow \mathbf{z}_{\text{feat}}$
  if MSA enabled **then**
    $\hat{\mathbf{f}}_{\text{msa}} \leftarrow \text{MSASubsample}(\mathbf{f}_{\text{msa}})$ ▷Resample MSA rows per loop iteration; A.2.4
    $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{MSAEncoder}(\mathbf{e}_t, \mathbf{f}_{\text{inputs}}, \hat{\mathbf{f}}_{\text{msa}})$ ▷A.2.4
  end if
  $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{LMEncoder}(\text{Dropout}(\mathbf{z}_{\text{lm}}, p{=}0.25))$ ▷Per-loop dropout
  $\mathbf{z}_{t+1} \leftarrow \text{PairFoldingLayers}(\bar{\mathbf{A}} \odot \mathbf{z}_t + \bar{\mathbf{B}}\, \text{LayerNorm}(\mathbf{u}_t))$ ▷Recurrent step; A.2.5
end for
$\mathbf{z} \leftarrow \text{PairFoldingLayers}_{[2]}(\text{Linear}(\mathbf{z}_T))$ ▷2 layers; Refine iteration output
$\mathbf{z}_{\text{disto}} \leftarrow \text{DistogramHead}(\mathbf{z} + \mathbf{z}^\top)$
$\mathbf{x}_{\text{pred}} \leftarrow \text{TruncatedDiffusionSampling}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{f}_{\text{rel\_pos}})$ ▷Algorithm 4
$\mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}} \leftarrow \text{ConfidenceHead}(\mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{x}_{\text{pred}}, \mathbf{f}_{\text{rel\_pos}}, \mathbf{f}_{\text{bond}})$ ▷Algorithm 10
return $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$



Published as a conference paper at ICLR 2024

LOOPED TRANSFORMERS ARE BETTER AT LEARNING LEARNING ALGORITHMS

Liu Yang, Kangwook Lee, Robert D. Nowak & Dimitris Papailiopoulos
University of Wisconsin, Madison, USA
{liu.yang, kangwook.lee, rdnowak}@wisc.edu, dimitris@papail.io

ABSTRACT

Transformers have demonstrated effectiveness in *in-context solving* data-fitting problems from various (latent) models, as reported by Garg et al. (2022). However, the absence of an inherent iterative structure in the transformer architecture presents a challenge in emulating the iterative algorithms, which are commonly employed in traditional machine learning methods. To address this, we propose the utilization of *looped* transformer architecture and its associated training methodology, with the aim of incorporating iterative characteristics into the transformer architectures. Experimental results suggest that the looped transformer achieves performance comparable to the standard transformer in solving various data-fitting problems, while utilizing less than 10% of the parameter count.[1]

1 INTRODUCTION

Transformers (Vaswani et al., 2017; Brown et al., 2020; Devlin et al., 2019) have emerged as the preferred model in the field of natural language processing (NLP) and other domains requiring sequence-to-sequence modeling. Besides their state-of-art performance in natural language processing tasks, large language models (LLM) such as GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2022) also exhibit the ability to learn in-context: they can adapt to various downstream tasks based on a brief prompt, thus bypassing the need for additional model fine-tuning. This intriguing ability of in-context learning has sparked interest in the research community, leading numerous studies (Min et al., 2022; Olsson et al., 2022; Li et al., 2023). However, the underlying mechanisms enabling these transformers to perform in-context learning remain unclear.

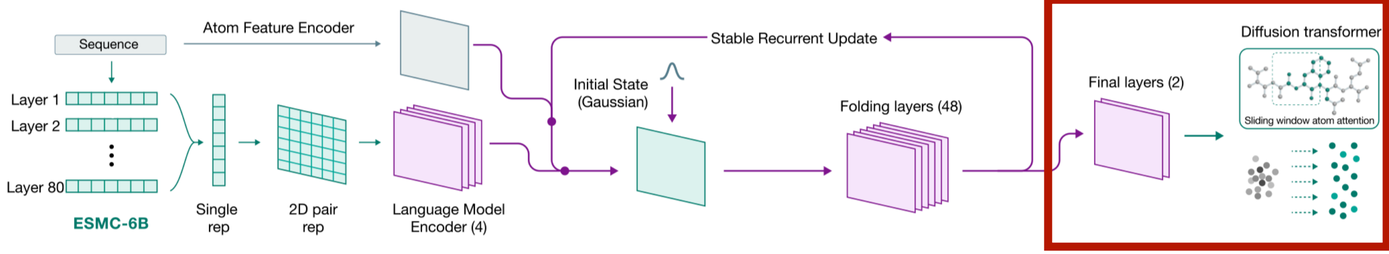
In an effort to understand the in-context learning behavior of LLMs, Garg et al. (2022) investigated the performance of transformers, when trained from scratch, in solving specific function class learning problems in-context. Notably, transformers exhibited strong performance across all tasks, matching or even surpassing traditional solvers. Building on this, Akyürek et al. (2022) explored the transformer-based model's capability to address the linear regression learning problem, interpreting it as an implicit form of established learning algorithms. Their study included both theoretical and empirical perspectives to understand how transformers learn these functions. Subsequently, von Oswald et al. (2022) demonstrated empirically that, when trained to predict the linear function output, a linear self-attention-only transformer inherently learns to perform a single step of gradient descent to solve the linear regression task in-context. While the approach and foundational theory presented by von Oswald et al. (2022) are promising, there exists a significant gap between the simplified architecture they examined and the standard decoder transformer used in practice. The challenge of training a standard decoder transformer from scratch, with only minor architectural modifications, to effectively replicate the learning algorithm remains an open question.

In traditional machine learning, *iterative* algorithms are commonly used to solve linear regression. However, the methodologies employed by standard transformers are not naturally structured for iterative computation. A *looped* transformer architecture, extensively studied in the literature such as Giannou et al. (2023), provides a promising avenue to bridge this gap. In addition to its inherent advantage of addressing problem-solving in an iterative manner, the looped transformer also breaks down tasks into simpler subtasks, potentially leading to significant savings in model parameters.

[1]Our code is available at https://github.com/Leiay/looped_transformer.

ESMFold 2

A New SotA for Protein Complex Structure Prediction



Algorithm 1 ESMFold2 Forward Pass.

Require: $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}}, \mathbf{f}_{\text{msa}}, \mathbf{f}_{\text{bond}}, \{\mathbf{x}_c\}, T$
Ensure: $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

$\mathbf{f}_{\text{inputs}} \leftarrow \text{InputEmbedding}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{seq}})$ ▷A.2.3
$\mathbf{f}_{\text{rel\_pos}} \leftarrow \text{RelPosEnc}$
$\mathbf{z}_{\text{feat}} \leftarrow \text{OuterSum}(\mathbf{f}_{\text{inputs}}) + \mathbf{f}_{\text{rel\_pos}} + \mathbf{f}_{\text{bond}}$ ▷Initial pair embedding
$\mathbf{z}_{\text{lm}} \leftarrow \text{ESMCRepresentation}(\{\mathbf{x}_c\})$ ▷Algorithm 2
$\mathbf{z}_0 \sim \text{trunc\_norm}(0, \frac{2}{5d_{\text{pair}}})$ ▷Initial recurrent state, $\pm 3\sigma$ truncation
for $t = 0$ to $T - 1$ **do**
  $\mathbf{u}_t \leftarrow \mathbf{z}_{\text{feat}}$
  if MSA enabled **then**
    $\hat{\mathbf{f}}_{\text{msa}} \leftarrow \text{MSASubsample}(\mathbf{f}_{\text{msa}})$ ▷Resample MSA rows per loop iteration; A.2.4
    $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{MSAEncoder}(\mathbf{e}_t, \mathbf{f}_{\text{inputs}}, \hat{\mathbf{f}}_{\text{msa}})$ ▷A.2.4
  end if
  $\mathbf{u}_t \leftarrow \mathbf{u}_t + \text{LMEncoder}(\text{Dropout}(\mathbf{z}_{\text{lm}}, p{=}0.25))$ ▷Per-loop dropout
  $\mathbf{z}_{t+1} \leftarrow \text{PairFoldingLayers}(\bar{\mathbf{A}} \odot \mathbf{z}_t + \bar{\mathbf{B}}\, \text{LayerNorm}(\mathbf{u}_t))$ ▷Recurrent step; A.2.5
end for
$\mathbf{z} \leftarrow \text{PairFoldingLayers}_{[2]}(\text{Linear}(\mathbf{z}_T))$ ▷2 layers; Refine iteration output
$\mathbf{z}_{\text{disto}} \leftarrow \text{DistogramHead}(\mathbf{z} + \mathbf{z}^\top)$
$\mathbf{x}_{\text{pred}} \leftarrow \text{TruncatedDiffusionSampling}(\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{f}_{\text{rel\_pos}})$ ▷Algorithm 4
$\mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}} \leftarrow \text{ConfidenceHead}(\mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{x}_{\text{pred}}, \mathbf{f}_{\text{rel\_pos}}, \mathbf{f}_{\text{bond}})$ ▷Algorithm 10
return $\mathbf{x}_{\text{pred}}, \mathbf{z}_{\text{disto}}, \mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

Algorithm 4 Truncated Diffusion Sampling.

Require: $\mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{inputs}}, \mathbf{z}_{\text{ij}}, M, \{\sigma_k\}_{k=0}^{N}$ with $\sigma_k \le \sigma_{\max}, \sigma_N = 0$
  Parameters: $\sigma_{\text{data}}, \gamma_0, \gamma_{\min}, \lambda, \eta$
Ensure: $\mathbf{x}_{\text{pred}}$

$\mathbf{x} \leftarrow \sigma_0 \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for each consecutive pair (σ_{k-1}, σ_k) in schedule **do**
  $\gamma \leftarrow \gamma_0$ **if** $\sigma_k > \gamma_{\min}$ **else** 0
  $\hat{t} \leftarrow \sigma_{k-1} \cdot (1 + \gamma)$
  $\epsilon_{\text{std}} \leftarrow \lambda \cdot \sqrt{\max(\hat{t}^2 - \sigma_{k-1}^2,\, 0)}$
  $\mathbf{x} \leftarrow \text{CenterRandomAugmentation}(\mathbf{x})$
  $\mathbf{x}_{\text{noisy}} \leftarrow \mathbf{x} + \epsilon_{\text{std}} \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{\text{denoised}} \leftarrow \text{DiffusionModule}(\mathbf{x}_{\text{noisy}}, \hat{t}, \mathbf{f}_{\text{atom}}, \mathbf{f}_{\text{inputs}}, \mathbf{z}_{\text{ij}})$
  $\mathbf{x}_{\text{noisy}} \leftarrow \text{RigidAlign}(\mathbf{x}_{\text{noisy}}, \mathbf{x}_{\text{denoised}})$
  $\mathbf{d} \leftarrow (\mathbf{x}_{\text{noisy}} - \mathbf{x}_{\text{denoised}})/\hat{t}$
  $\mathbf{x} \leftarrow \mathbf{x}_{\text{noisy}} + \eta \cdot (\sigma_k - \hat{t}) \cdot \mathbf{d}$
end for
return $\mathbf{x}_{\text{pred}} \leftarrow \mathbf{x}$

EDM

Algorithm 10 ConfidenceHead.

Require: $\mathbf{f}_{\text{inputs}}, \mathbf{z}, \mathbf{x}_{\text{pred}}, \mathbf{f}_{\text{rel\_pos}}, \mathbf{f}_{\text{bond}}$
Ensure: $\mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

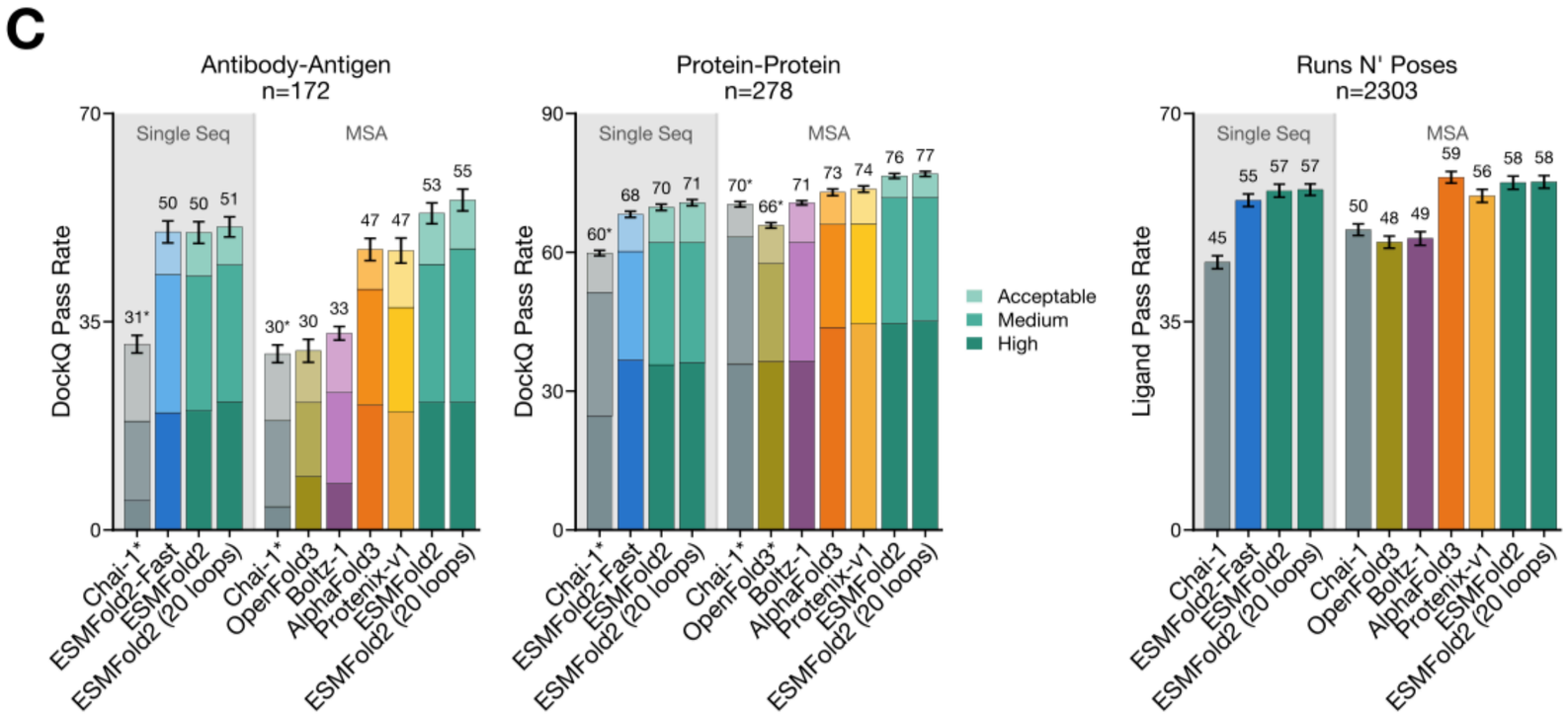
$\mathbf{d}_{\text{pred}} \leftarrow \text{Distogram}(\mathbf{x}_{\text{pred}})$
$\mathbf{z}_c \leftarrow \text{OuterSum}(\mathbf{f}_{\text{inputs}}) + \mathbf{f}_{\text{rel\_pos}} + \mathbf{f}_{\text{bond}} + \text{Embed}(\mathbf{d}_{\text{pred}})$
$\beta \sim \text{Bernoulli}(0.8)$
$\mathbf{z}_c \leftarrow \mathbf{z}_c + \beta \cdot \text{Linear}(\mathbf{z})$
for 4 layers **do**
  $\mathbf{z}_c \leftarrow \text{PairFoldingLayer}(\mathbf{z}_c)$
end for
$\boldsymbol{\alpha}_{ij} \leftarrow \text{Softmax}_j\left(\text{Linear}(\mathbf{z}_c[i,j],\, 1) + m_j\right)$
$\mathbf{s}_c[i] \leftarrow \text{Linear}\left(\sum_j \boldsymbol{\alpha}_{ij}\, \mathbf{z}_c[i,j],\, d_{\text{single}}\right)$
$\mathbf{x}_{\text{plddt}} \leftarrow \text{Linear}(\text{LayerNorm}(\mathbf{s}_c))$
$\mathbf{x}_{\text{resolved}} \leftarrow \text{Linear}(\text{LayerNorm}(\mathbf{s}_c))$
$\mathbf{z}_{\text{norm}} \leftarrow \text{LayerNorm}(\mathbf{z}_c)$
$\mathbf{z}_{\text{pde}} \leftarrow \text{Linear}(\mathbf{z}_{\text{norm}} + \mathbf{z}_{\text{norm}}^\top)$
$\mathbf{z}_{\text{pae}} \leftarrow \text{Linear}(\text{LayerNorm}(\mathbf{z}_c))$
return $\mathbf{x}_{\text{plddt}}, \mathbf{x}_{\text{resolved}}, \mathbf{z}_{\text{pde}}, \mathbf{z}_{\text{pae}}$

Confidence Outputs

ESMFold 2

A New SotA for Protein Complex Structure Prediction

ESMFold 2 outperforms AlphaFold 3 even without MSA, and benefits further from optional MSA inputs.



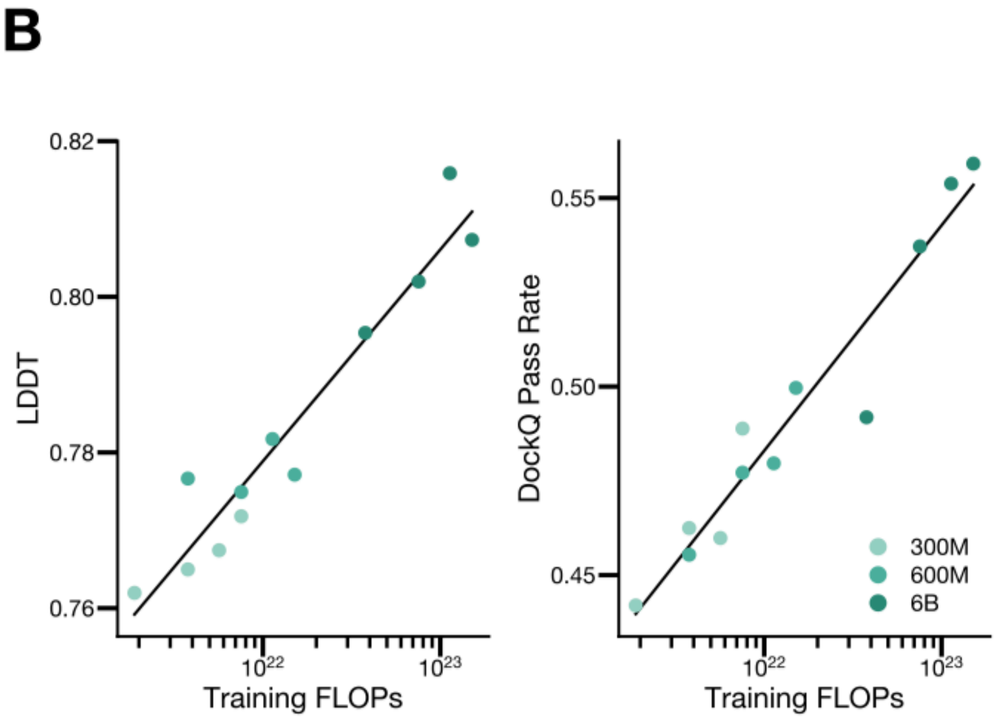
Comparison on FoldBench Antibody-Antigen (Left), Protein-Protein (Middle), and Runs N' Poses Benchmark (Right)

ESMFold 2

A New SotA for Protein Complex Structure Prediction

The scaling law extends to ESMFold 2: structure prediction performance improves with the quality of ESMC representations.

FoldBench Monomer



FoldBench Protein-Protein

Scaling Law of Structure Prediction Accuracy with ESMC Training FLOPs

ESMFold 2

Designing De Novo Minibinder and scFvs

Beyond structure prediction, ESMFold2 can be used to design binders for a given target by searching the sequence space and sampling candidate binders

$$p(x, s) = p(s \mid x, t)p(x)$$

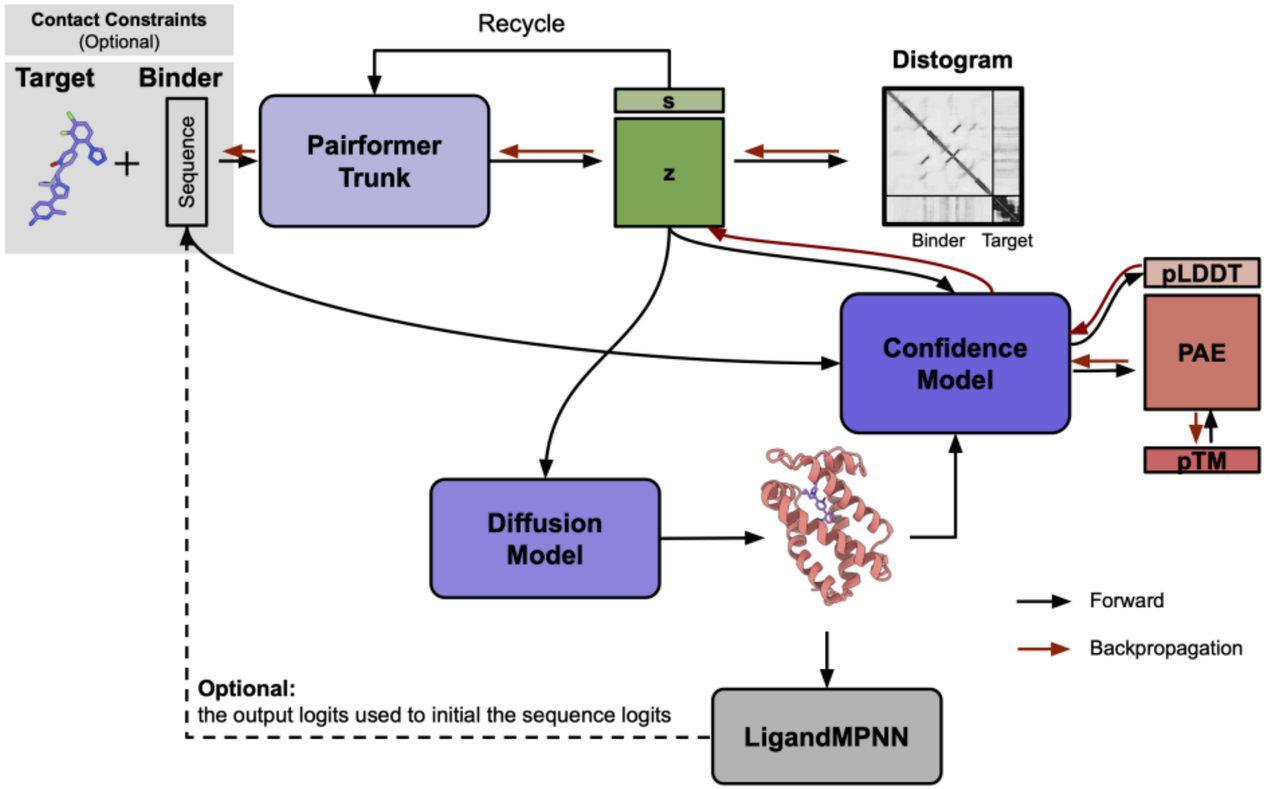
where

- x: the candidate binder sequence (desired output);
- t: the target sequence (input);
- s: the structure of the target-bound complex (auxiliary output).

ESMFold 2

Designing De Novo Minibinder and scFvs

Similar to BoltzDesign1, new protein sequences are designed by iteratively updating amino acid token logits through backpropagation.



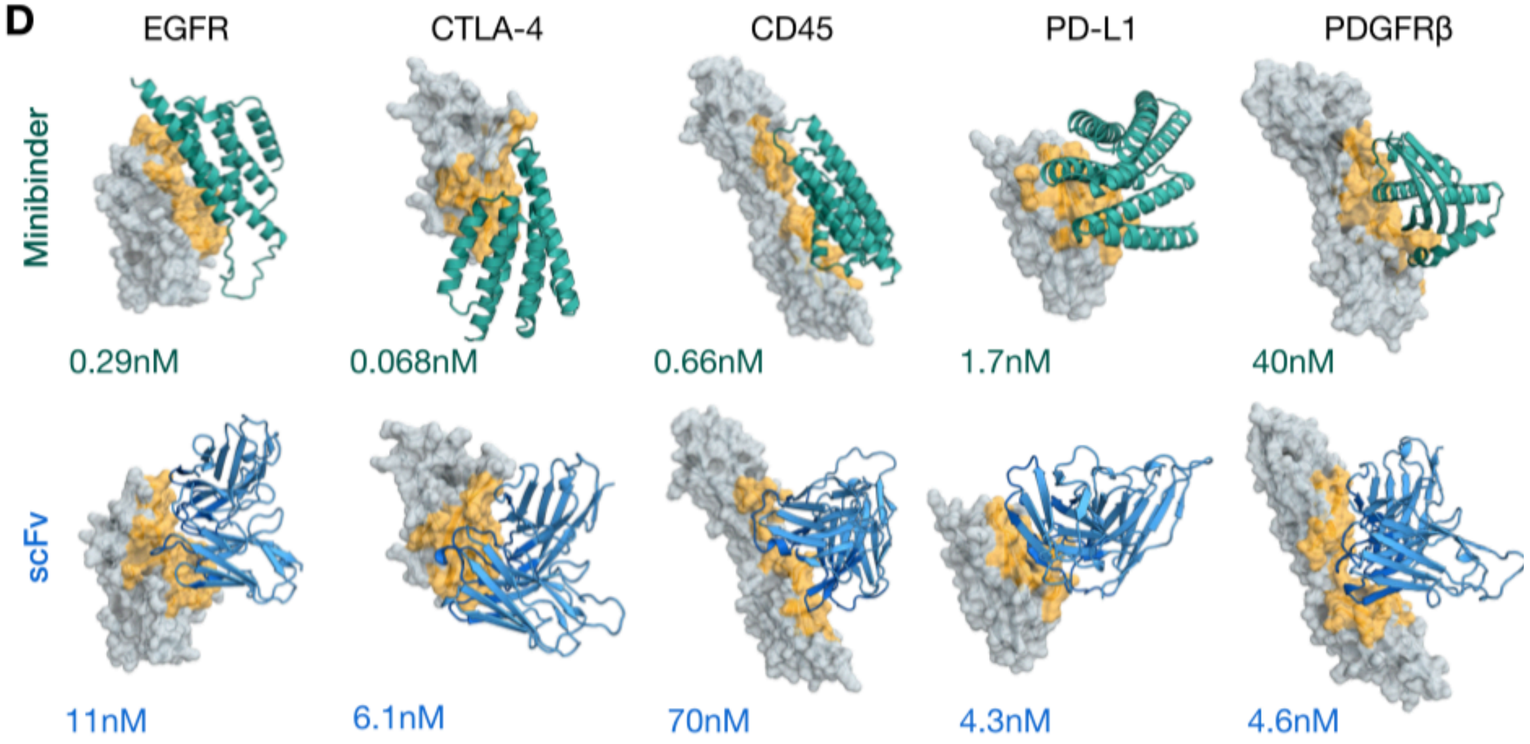
BoltzDesign1 Overview, Cho et al., bioRxiv 2025

ESMFold 2

Designing De Novo Minibinder and scFvs

Across five clinically relevant targets (PDGFRβ , EGFR, PD-L1, CD45, CTLA-4), the gradient-based sequence optimization produced high-affinity binders.

Wet Lab Validated



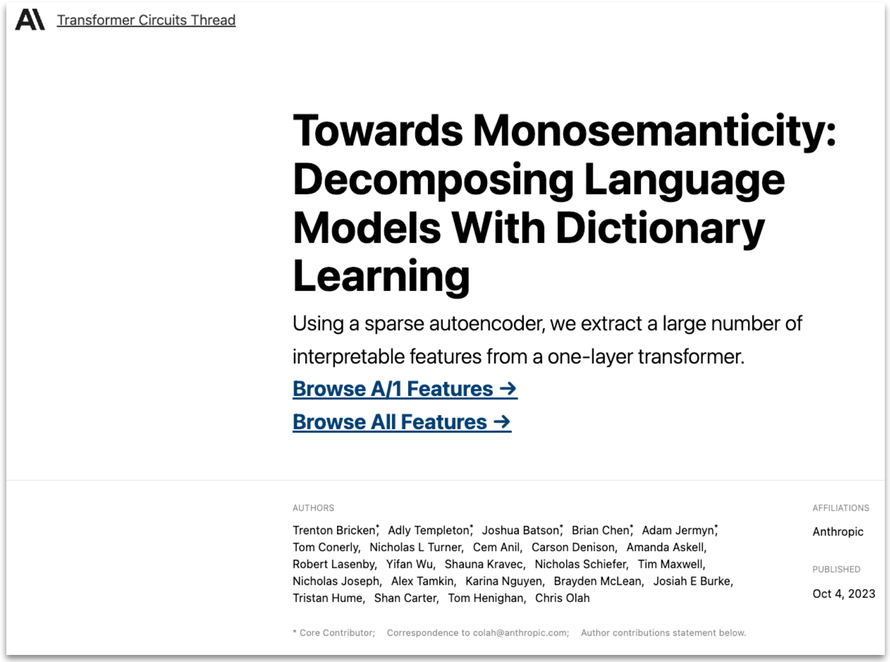
High-affinity binders for each target and modality. Gray: Target proteins, Cartoons: Designed binders.

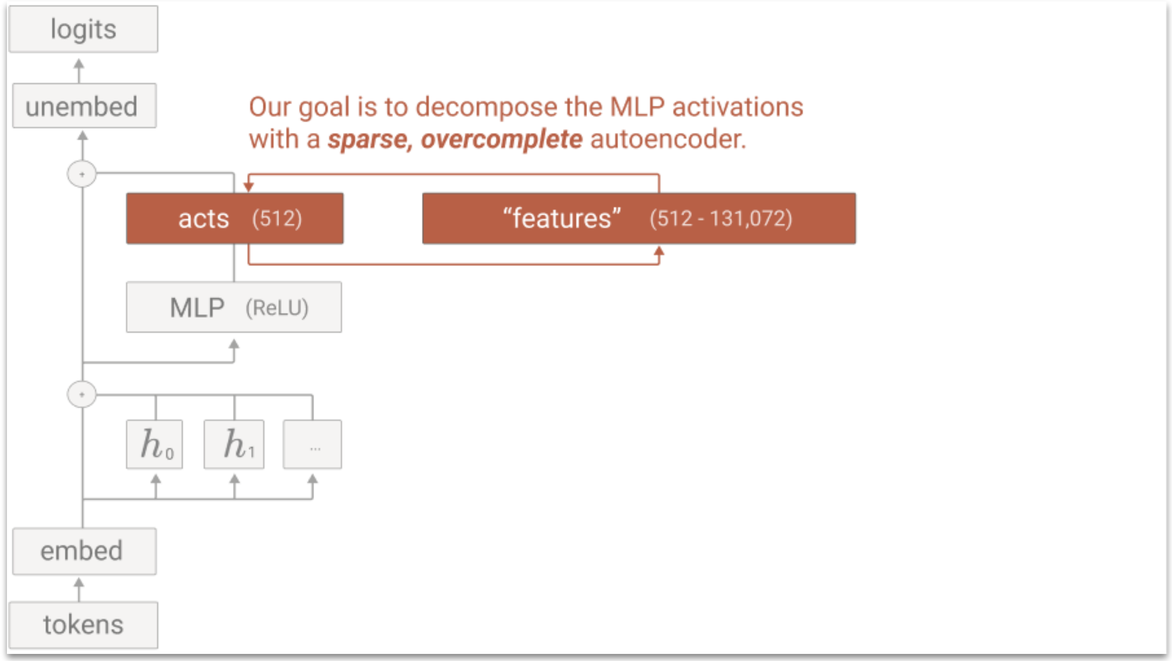
Do ESMC representations encode information useful beyond structure prediction?

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

The ESMC latent space is analyzed using **sparse autoencoders (SAEs)**, a method widely used to identify interpretable features in LLMs.





Towards Monosemanticity, Anthropic, 2023

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

For each token's hidden-state activation $x \in \mathbb{R}^{d_{\text{model}}}$, an SAE is trained to map it to a sparse vector $f \in \mathbb{R}^{d_{\text{SAE}}}$ ($d_{\text{SAE}} > d_{\text{model}}$):

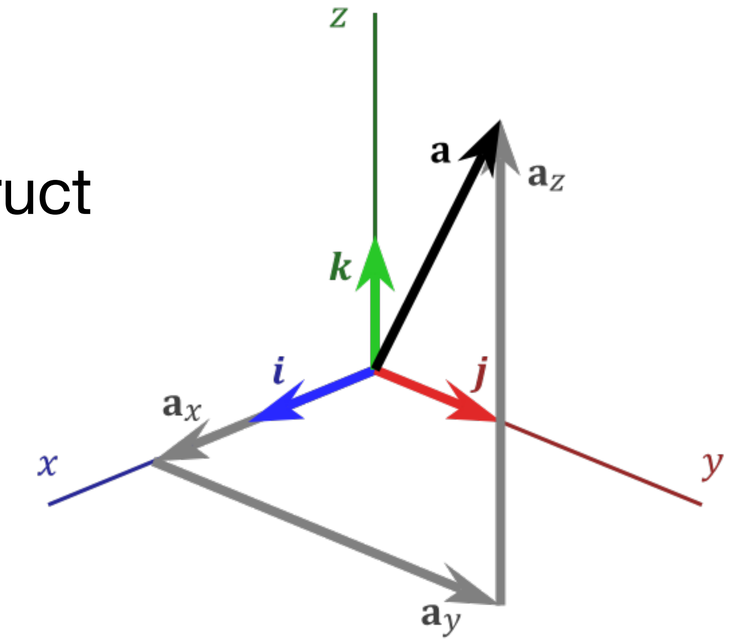
$$f = \text{ReLU}(W_{\text{enc}}^{\top}(x - b)),$$

which can be passed through a decoder layer to reconstruct

$$\hat{x} = W_{\text{dec}}^{\top} f + b.$$

Basis **Coordinates**

To enforce sparsity, we keep top K elements of f and zero-out the rest during training.



Standard Basis, Wikipedia

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

SAEs are trained for every layer of ESMC-{300M, 600M, 6B} models, using 8B tokens drawn from the pretraining dataset. Specifically:

1. A separate SAE is trained for every layer;

2. For each layer, SAEs with different feature dimensions ($2^{13} - 2^{17}$) and sparsity levels (8-128) are trained;

3. The analysis mainly focus on an SAE trained with activations at **layer 60**, with a 2^{14}**-dimensional feature space** and **sparsity level of** 64.

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

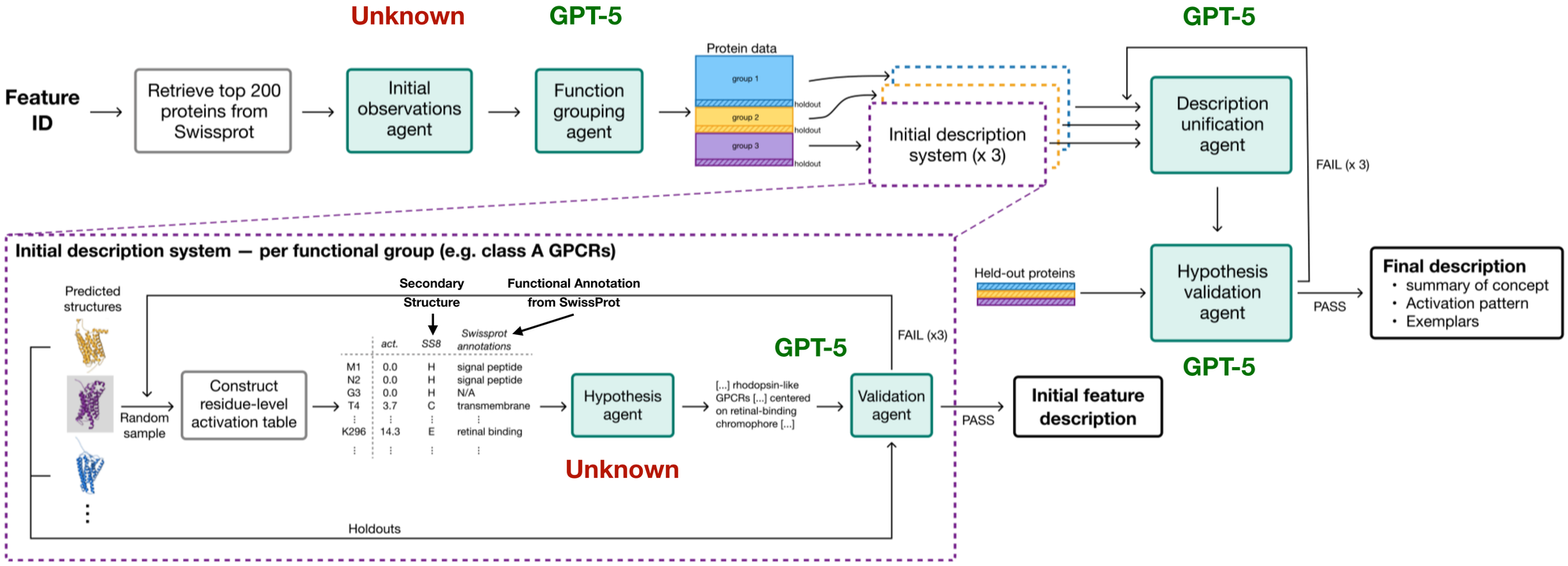
Due to the high dimensionality of the feature space (2^{14}), the team developed an agentic system to annotate features in natural language including:

- **Summary**: The biological concept represented by a feature;
- **Examples**: Proteins in which the feature activates;
- **Activation Pattern**: A description of the activation pattern.

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

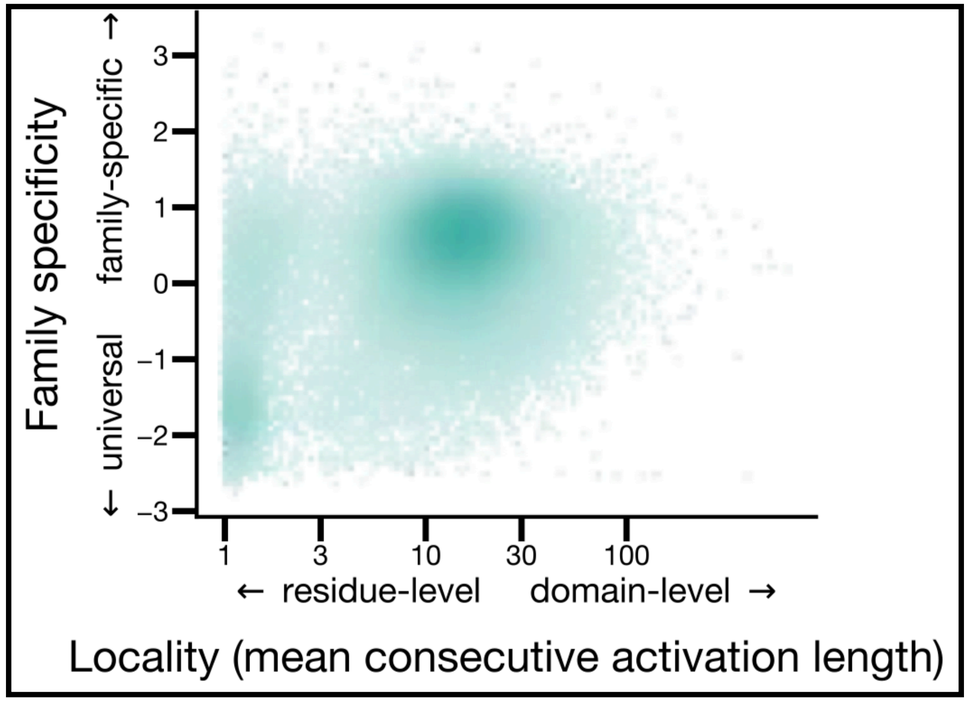
The system iterates over hypothesis-verification cycle with multiple data splits and holdouts, producing a generalizable descriptions as outputs.



Multi-agent system for generating feature descriptions based on top-activating proteins from Swissprot

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)



Local

Global

B

Specificity/ Locality				
Residue identity (n=88)	**F189:** Tryptophans	**F2319:** Threonines	**F9000:** Negatively-charged, prefers Glu	**F12945:** Nonstandard amino acids
Secondary structure (n=1895)	**F5891:** C-term. α-helix loop into β-strand	**F12575:** Helix/ strand capping residues	**F13801:** Generic α-helix, *i–i+4* positions	**F1350:** High-confidence β-strand positions
Tertiary motif (n=1554)	**F5876:** Beta-barrel scaffold residues	**F9577:** Disulfide bonds	**F14430:** Register positions in helical repeats	**F15663:** Helix-turn-helix hairpins
Domain / fold (n=2546)	**F15943:** WD40 repeat β-propellers	**F554:** DNA polymerase clamp domain	**F3264:** CAZyme catalytic domains	**F14236:** Allosteric sensing/ regulatory modules
Disorder / low-complexity (n=686)	**F5474:** Disordered loops between structured regions	**F3220:** Plant regulatory IDRs	**F1215:** Module boundary linkers	**F11442:** Low-complexity IDRs
Biochemical environment (n=1770)	**F16325:** Aromatic-rich regions	**F3026:** Alt. charged/ hydrophobic interface	**F11549:** Acidic region in phosphate metalloenzymes	**F14444:** Hydrophobic assembly/ anchoring
Localization / topology (n=2319)	**F15681:** N-terminal membrane anchors	**F6821:** Juxtamembrane helix transitions	**F1195:** Post-cleavage region of proproteins	**F3890:** Secretome-associated proteins
Functional site / region (n=5382)	**F3616:** Catalytic DNA-binding groove	**F14193:** Proteolytic cleavage junctions	**F8445:** Glycoside hydrolase active sites	**F13526:** Heteroaromatic ligand recognition

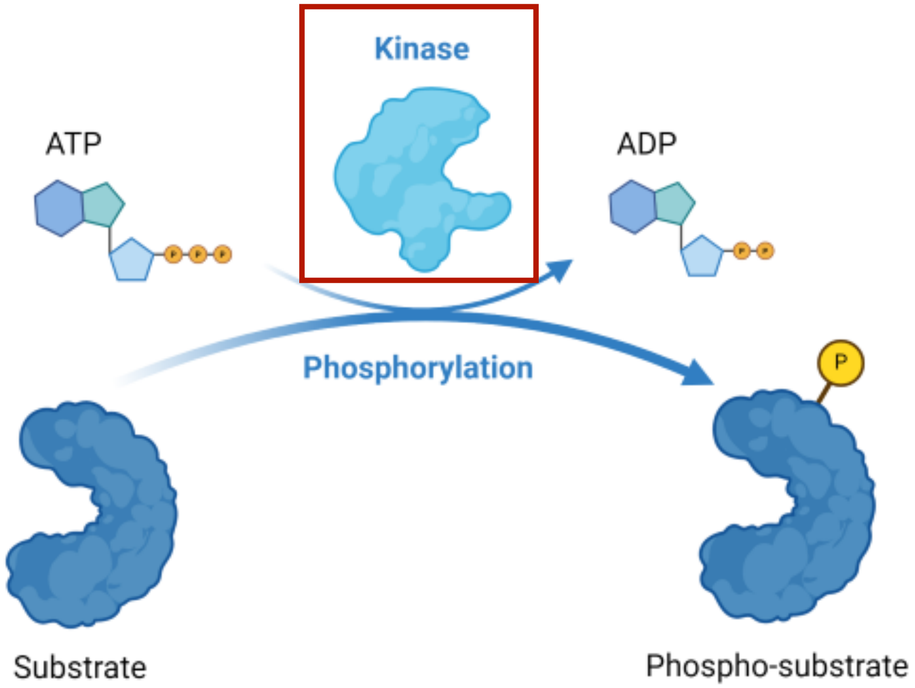
Distribution of features in 2D and examples of biological concepts captured by SAE features

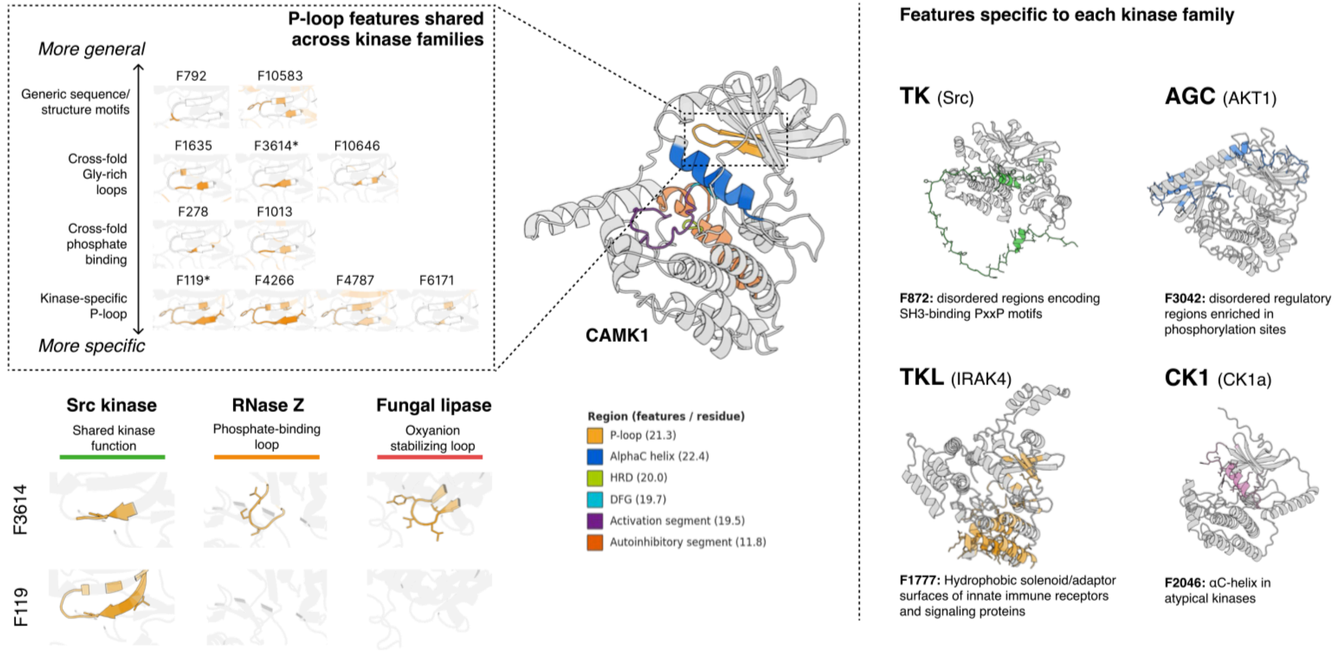
Bonus: A cool video explaining the structure of kinases

Analyzing ESMC Latent Space

Mechanistic Interpretation using Sparse Autoencoders (SAEs)

Complex functional mechanisms are represented through the composition of many features (*e.g.*, glycine-rich loop and activation loops of kinases).





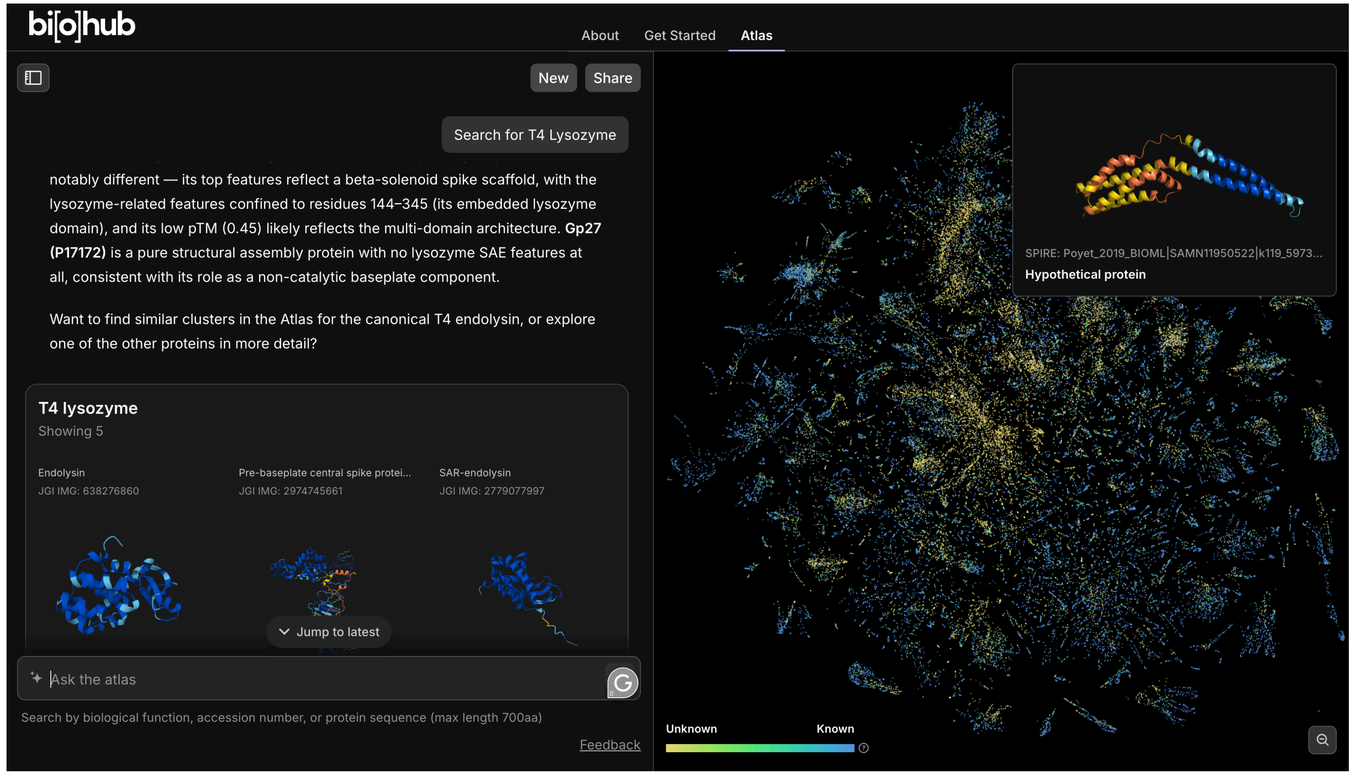
Phosphorylation catalyzed by a kinase

SAE features activated on functionally relevant regions on a kinase CAMK1

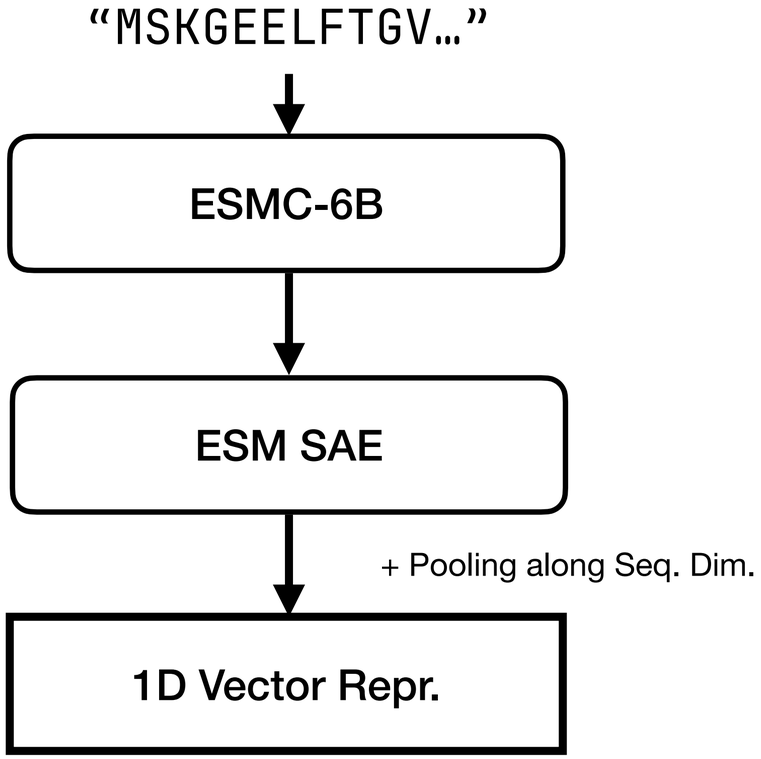
ESM Atlas

A Map of 6.8B Sequences and 1.1B Structures

Using ESMC representations, the team built ESM Atlas, which maps the structure and function of 6.8B sequences and 1.1B structures.



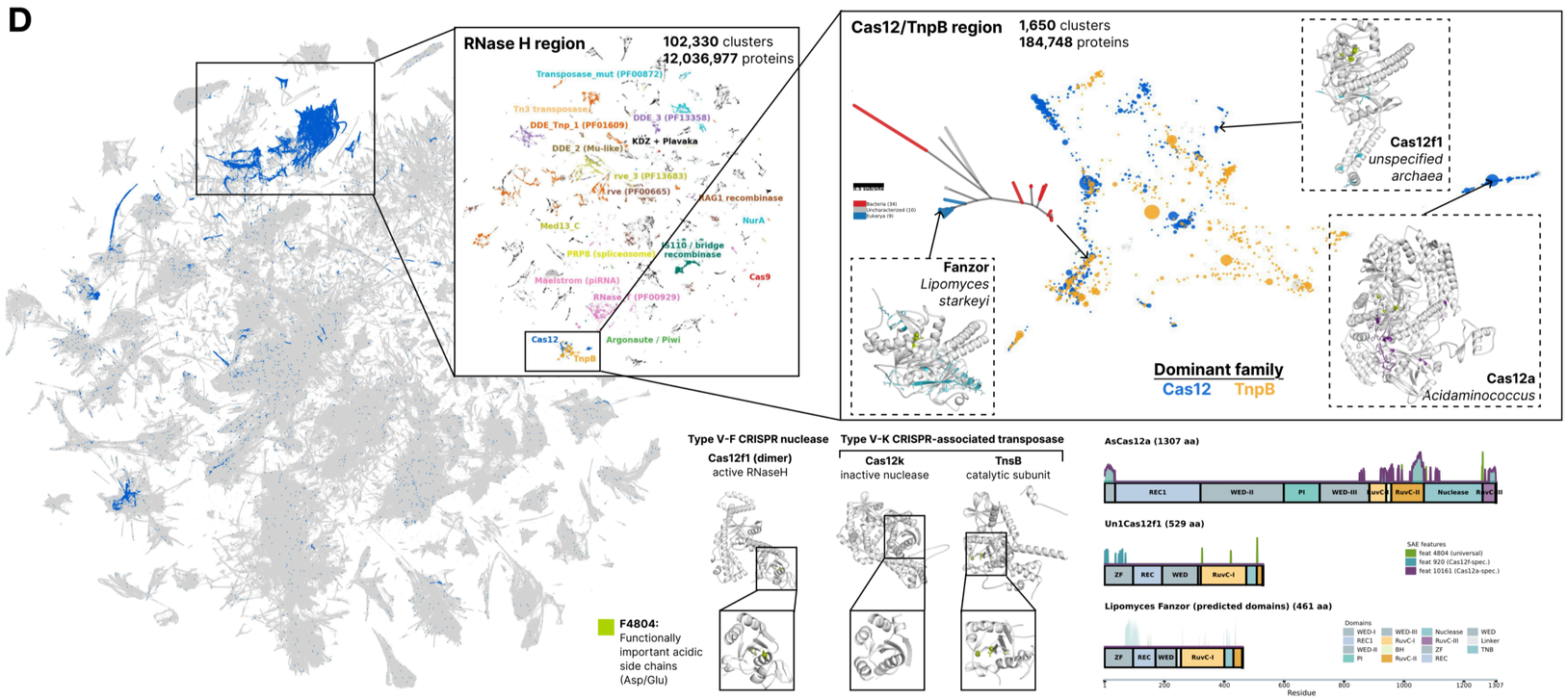
ESM Atlas Web Interface



ESM Atlas

A Map of 6.8B Sequences and 1.1B Structures

SAE features enable protein clustering and retrieval based on shared functional and structural concepts, beyond sequence or structure similarity alone.



UMAP projection of ESMC embeddings encoded from 7.7M sequences representing clusters with 50> members

Conclusion

To summarize, this paper presents

1. **ESMC**, a family of protein language models trained on 2.8B sequences, exhibiting emergent representations of protein structure and function.

Conclusion

To summarize, this paper presents

1. **ESMC**, a family of protein language models trained on 2.8B sequences, exhibiting emergent representations of protein structure and function.

2. **ESMFold2**, a structure prediction model built on ESMC representations, achieving state-of-the-art performance without requiring MSAs.

Conclusion

To summarize, this paper presents

1. **ESMC**, a family of protein language models trained on 2.8B sequences, exhibiting emergent representations of protein structure and function.

2. **ESMFold2**, a structure prediction model built on ESMC representations, achieving state-of-the-art performance without requiring MSAs.

3. **Mechanistic analyses** showing that ESMC representations capture biologically meaningful concepts across multiple layers.

Conclusion

To summarize, this paper presents

1. **ESMC**, a family of protein language models trained on 2.8B sequences, exhibiting emergent representations of protein structure and function.

2. **ESMFold2**, a structure prediction model built on ESMC representations, achieving state-of-the-art performance without requiring MSAs.

3. **Mechanistic analyses** showing that ESMC representations capture biologically meaningful concepts across multiple layers.

4. **ESM Atlas**, a large-scale resource linking 6.8B sequences to 1.1B predicted structures for protein search, retrieval, and clustering.

Thank You

Q&A

Language Modeling Materializes a World Model of Protein Biology

Team Biohub and EvolutionaryScale

Seungwoo Yoo, KAIST Visual AI Group

https://dvelopery0115.github.io/