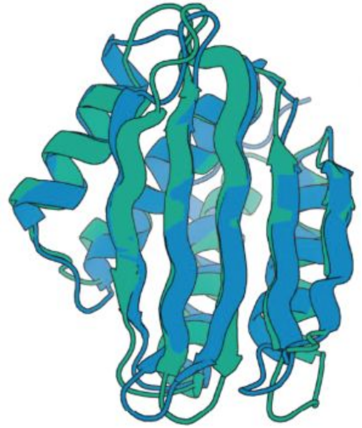
Robust deep learning-based protein sequence design using ProteinMPNN

Brendan Wang

October 12, 2023



Roadmap

- **Motivation**: why protein design?
- **Background**: what exists in the field?
- **Methods**: what is ProteinMPNN and how did the authors build it?
- **Evaluation and results**: how did the model perform?
- **Conclusions**: what's next?

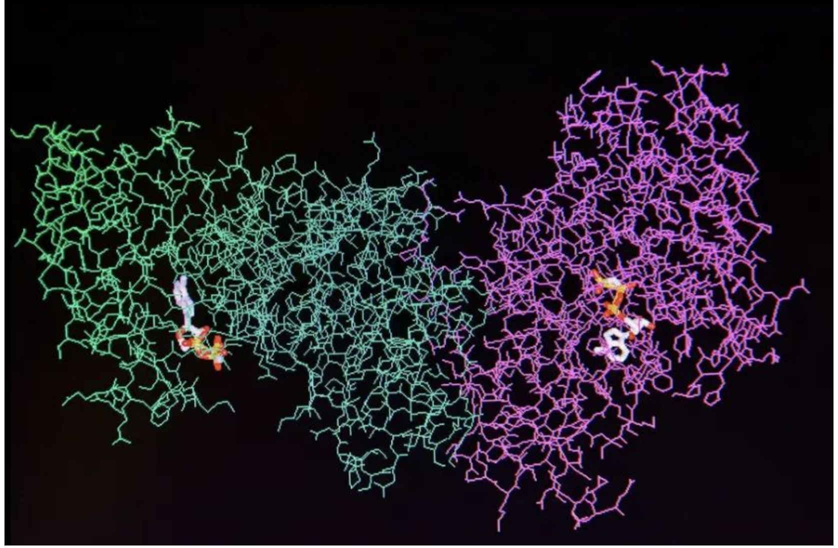
Motivation

AI-designed protein shells could make vaccines more effective

Protein shells designed using AI can work as carriers for immunity-inducing molecules, generating more antibodies in mice than some competing vaccine approaches

By Karmela Padavic-Callaghan

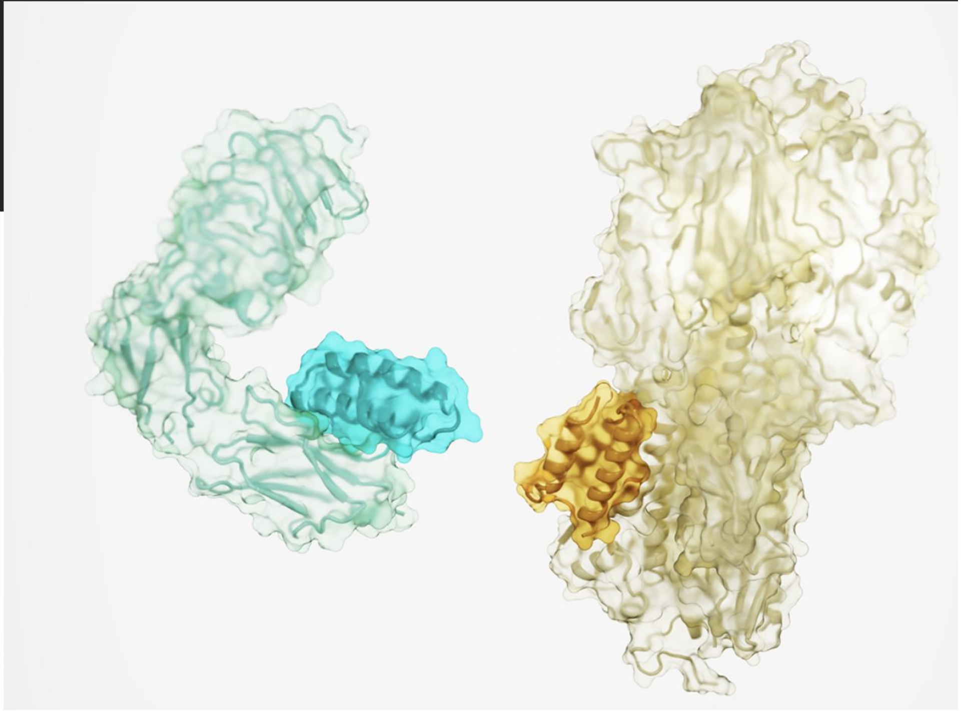
20 April 2023



Software-designed miniproteins could create new class of drugs

Small versions of antibodies bind to virtually any target protein

24 MAR 2022 • 1:35 PM ET • BY ROBERT F. SERVICE



“***...de novo* protein design** was nominated as one of the **top 10 annual breakthroughs** by Science in 2016.”

What is protein design?

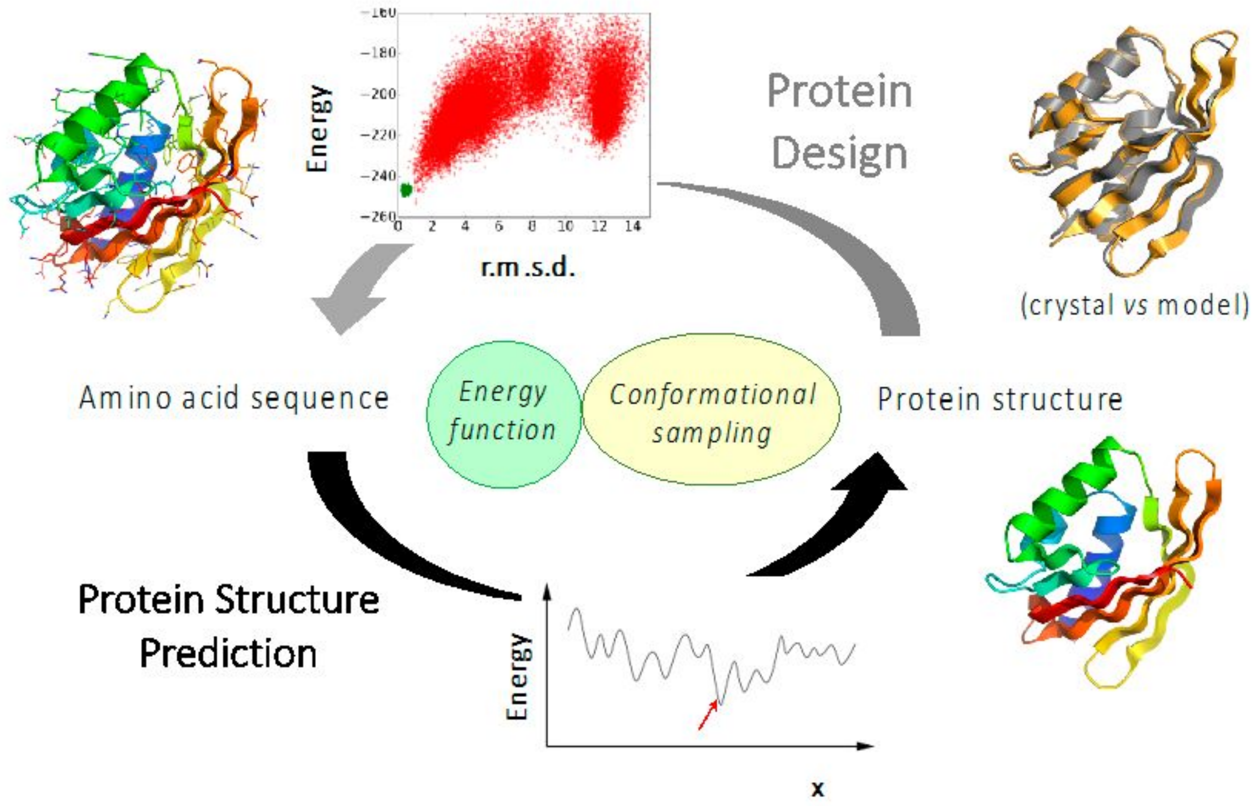
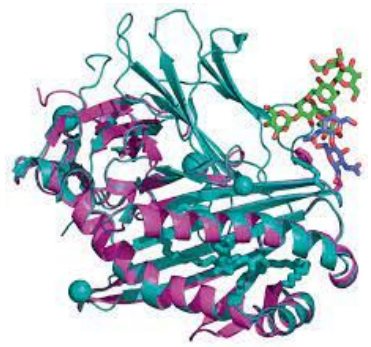


Image: https://www.ibmb.csic.es/en/department-of-structural-and-molecular-biology/protein-design-and-modeling/

Task is to find amino acid sequence that results in a desired protein structure that is stable and functional

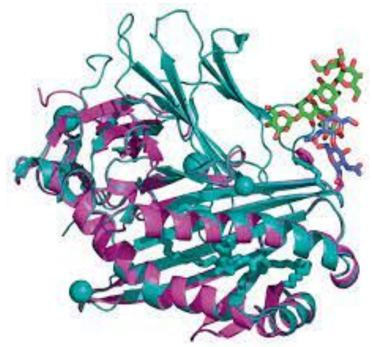


Inverse Folding →

Sequence

?

Experimental Validation →



Stable? Binds efficiently? Soluble?

Applications of Protein Design

- **Pharmaceuticals**: drug design, vaccine development
- **Biotechnology**: bioremediation, biocatalysis
- **Biosensors**: synthetic circuits, metabolic engineering
- **Material Science**: nanomaterials, biopolymers

“Native” proteins selected through millions of years of evolution are not likely to support these needs so want to **design** proteins (often de-novo) tailored to such needs.

Background

Current existing approaches for protein design

Physics-based: energy optimization

- Slower two-step process with need for customization

Deep-learning-based: pattern recognition

- Lacks physical transparency and extensive experimental design

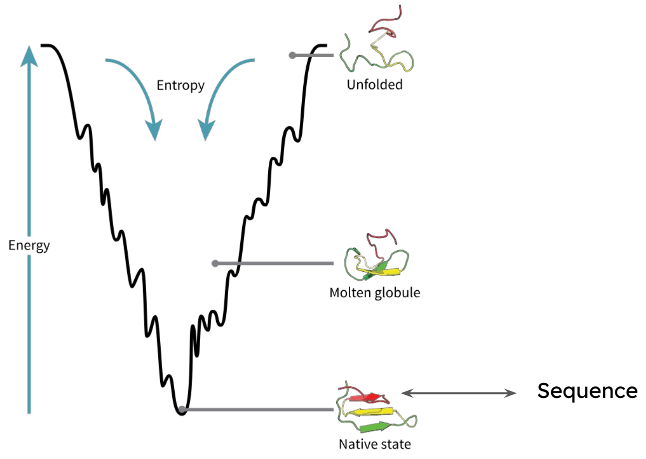


Image: https://upload.wikimedia.org/wikipedia/commons/9/91/Folding_funnel_schematic.svg

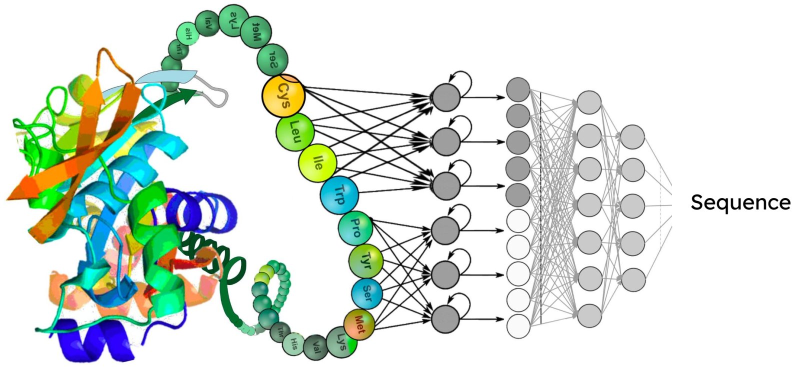
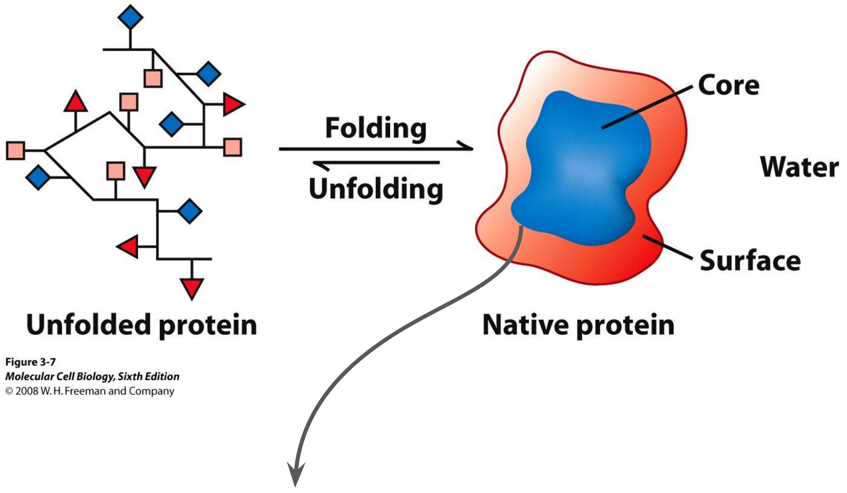


Image: https://scx2.b-cdn.net/gfx/news/hires/2022/study-evaluates-deep-l.jpg

Rosetta is one current state-of-the-art method

- Physics based method that scans sequence space and evaluate energy of chosen sequence
- Requires side-chain packing calculations and expert customization



Hydrophobic amino acids restricted to surface to stabilize undesired multimeric states. How much restriction to place at boundary?

Image: https://www.allometric.com/tom/courses/protected/MCB6/ch03/3-07.jpg

How can we **efficiently** use deep learning to predict amino acid sequences based on protein structures in a **robust**, **self-sufficient**, **accurate** way?

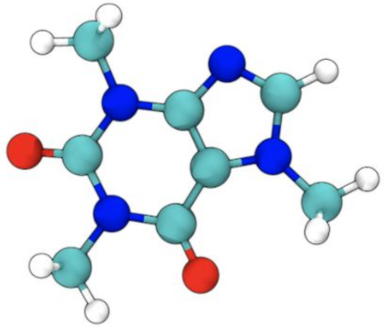
Method

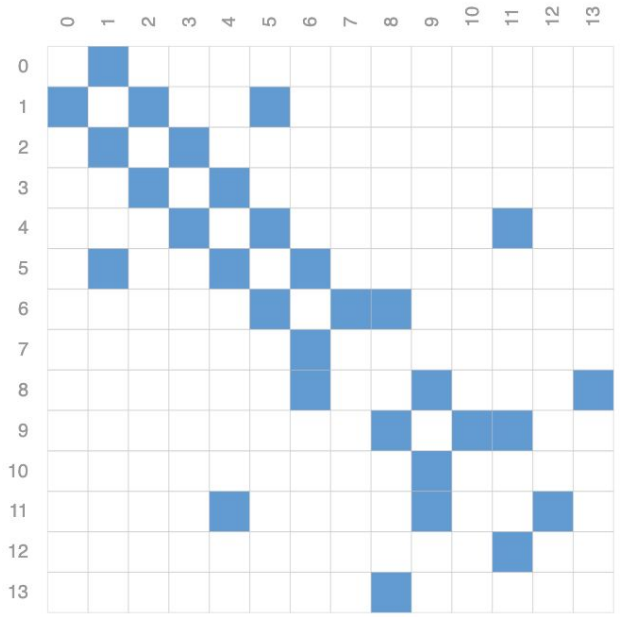
(Protein **Message Passing Neural Network** or ProteinMPNN)

$\downarrow$

Special type of Graph Neural Network

Graph Neural Networks operate on graphical data





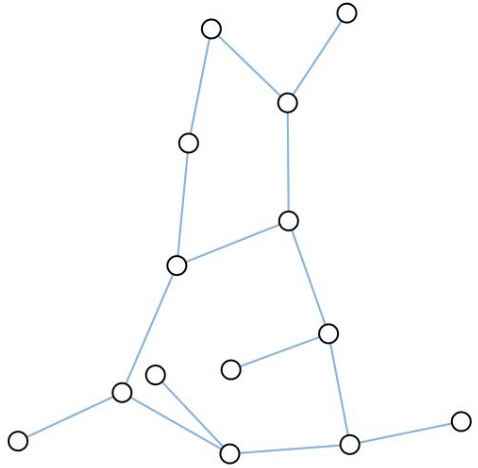


Image: https://distill.pub/2021/gnn-intro/

Message Passing Neural Networks

- Message Passing Paradigm: update node representations based on aggregation from nearby nodes. Three main steps:

  ① Initialization: set each node embedding to seed

  $$\vec{h}_v^{(0)} = \vec{x}_v \quad \forall \vec{v} \in V$$

  ② Aggregation: aggregate information of node and its neighbors from prev. time step

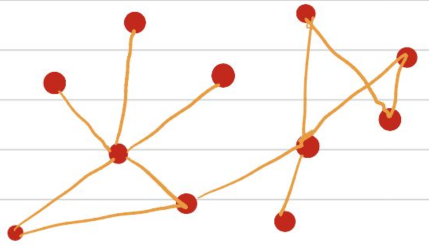
  $$\vec{m}_v^{(t)} = f_{Agg}^{(t)}\left(\vec{h}_v^{(t-1)}, \{\vec{h}_u^{(t-1)} : \vec{u} \in N(v)\}\right), \quad 1 \leq t \leq T$$

  Ex: $\vec{m}_v^{(t)} = \frac{1}{|N(v)+1|} \sum_{u \in N(v)} \vec{h}_u^{t}$

  ③ Transformation: set new node embedding as fn of old node emb. and aggregated info

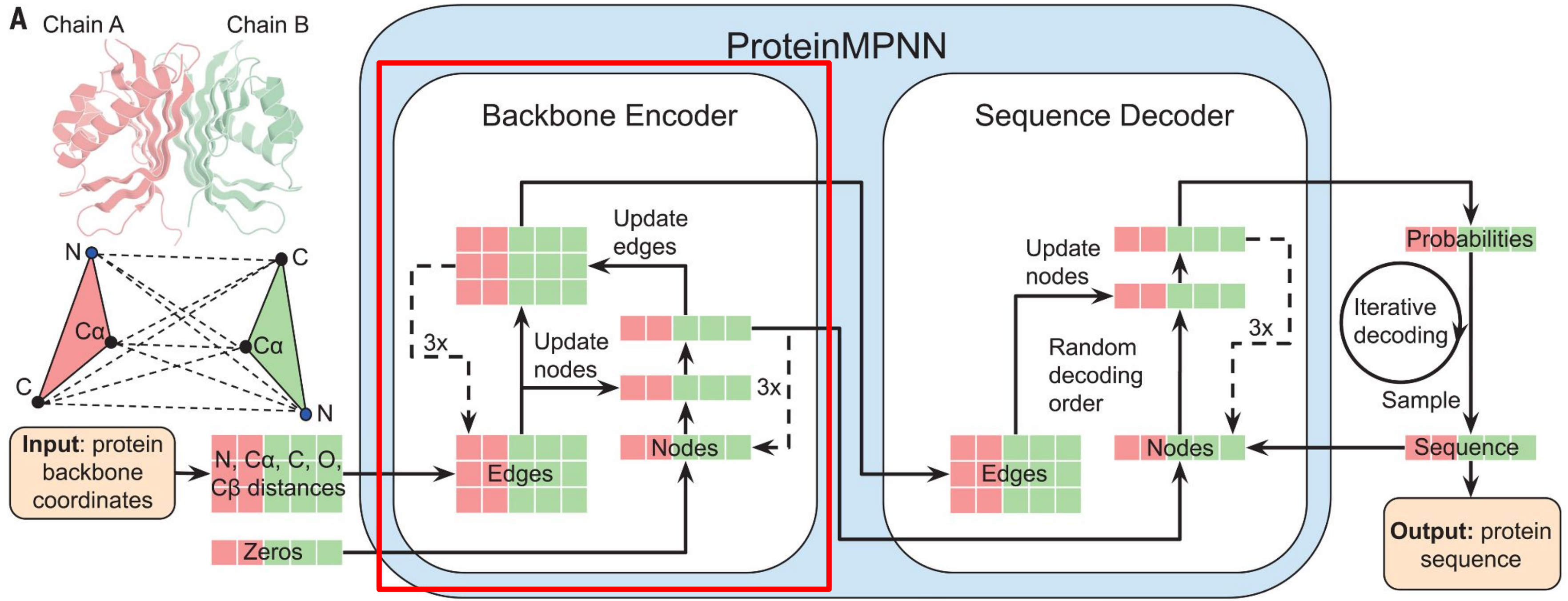
  $$\vec{h}_v^{(t)} = f_{update}\left(\vec{h}_v^{(t)}, \vec{m}_v^{(t)}\right)$$

  Ex: $\vec{h}_v^{(t)} = \sigma\left(W^{(t)} m_v^{(t)}\right)$



- Graph $G = (V, E, X, W)$
  - → V: vertices
  - → E: edges
  - → X: node attributes
  - → W: edge attributes

ProteinMPNN operates on structures represented as graphs



Pseudocode for the encoder layer (V - node features, E - edge features):

```
def encoder_layer_forward(V, E):
    M_ij = MLP[V_i, V_j, E_ij]
    dV_i = Sum_j [M_ij]
    V_i = LayerNorm[V_i + Dropout(dV_i)]
    dV_i = FeedForward[V_i]
    V_i = LayerNorm[V_i + Dropout(dV_i)]
    dE_ij = MLP[V_i, V_j, E_ij]
    E_ij = LayerNorm[E_ij + Dropout(dE_ij)]
    return V, E
```

Get intermediate representation or "message" based on information of neighbors and edges for node i

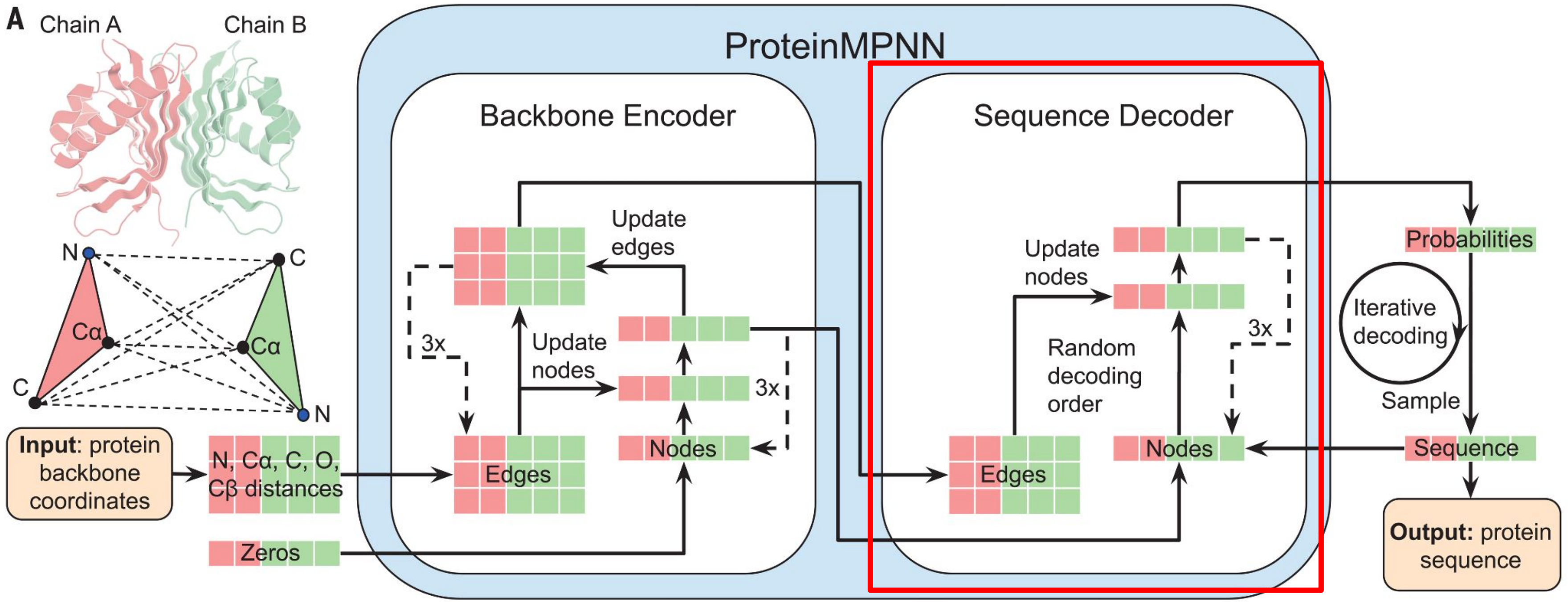
Sum messages across all neighbors

Updates node representations

Updates edges representations based on new node representations

Encoder layer is repeated 3 times – propogate messages **3 neighborhoods** away

ProteinMPNN operates on structures represented as graphs



Pseudocode for the decoder layer (V - node features, E - edge features, S - sequence features, mask - autoregressive mask):

Use information about previous time step to predict at current step

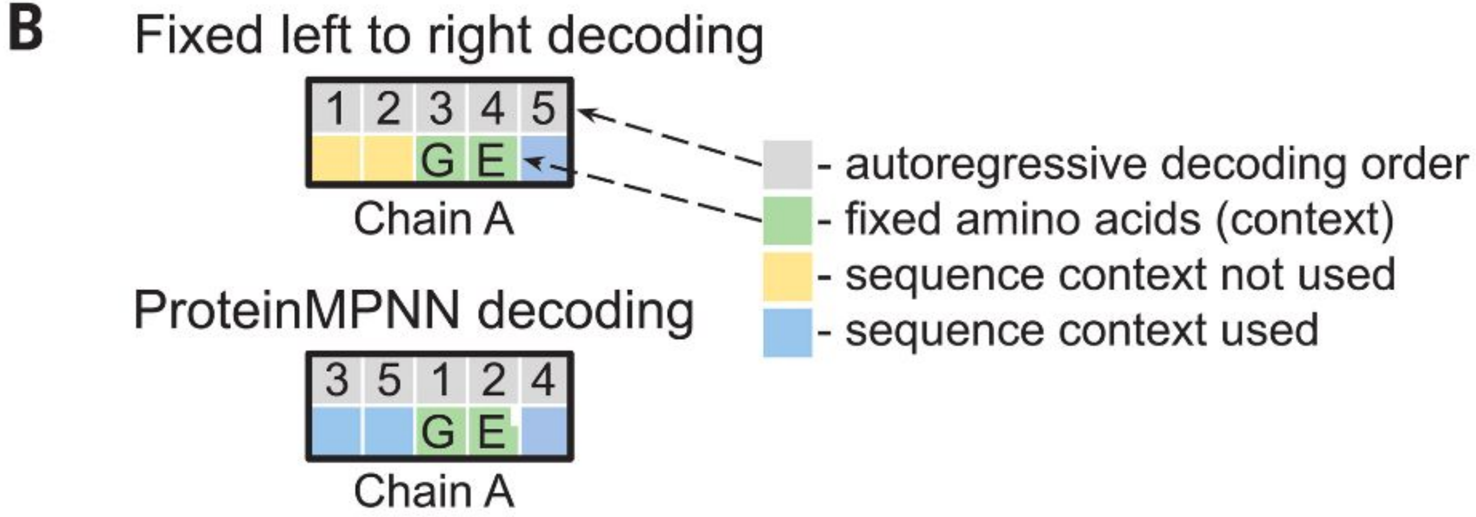
```
def decoder_layer_forward(V, E, S, mask):
    E_ij = Concat[E_ij, S_j] * mask_ij + Concat[E_ij, 0.0*S_j]  * (1-mask_ij)
    M_ij = MLP[V_i, V_j, E_ij]
    dV_i = Sum_j [M_ij]
    V_i = LayerNorm[V_i + Dropout(dV_i)]
    dV_i = FeedForward[V_i]
    V_i = LayerNorm[V_i + Dropout(dV_i)]
    return V
```

Add edge to sequence features together

Use encoded neighbor embeddings to update current embeddings

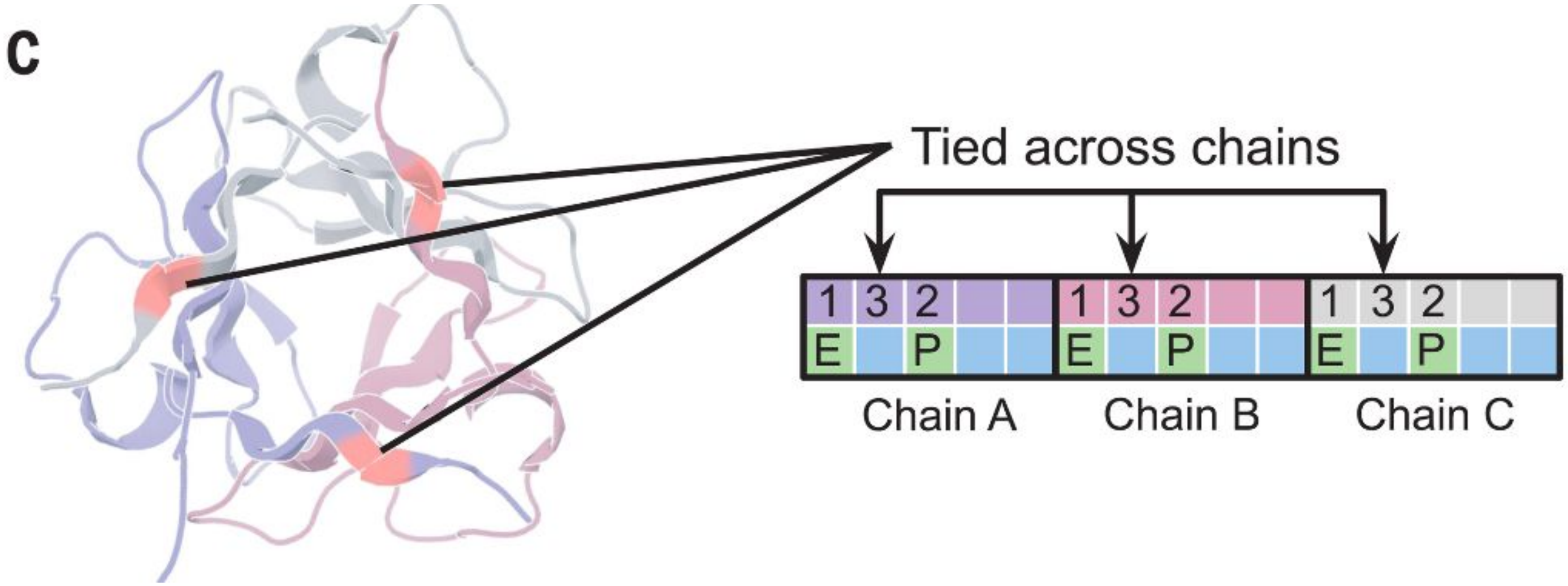
Decoder layer is repeated 3 times – get messages **3 neighborhoods** away

ProteinMPNN uses random decoding order



B
Fixed left to right decoding
1 2 3 4 5
G E
Chain A
ProteinMPNN decoding
3 5 1 2 4
G E
Chain A
- autoregressive decoding order
- fixed amino acids (context)
- sequence context not used
- sequence context used

ProteinMPNN uses positional coupling for multichain predictions



How was ProteinMPNN trained?

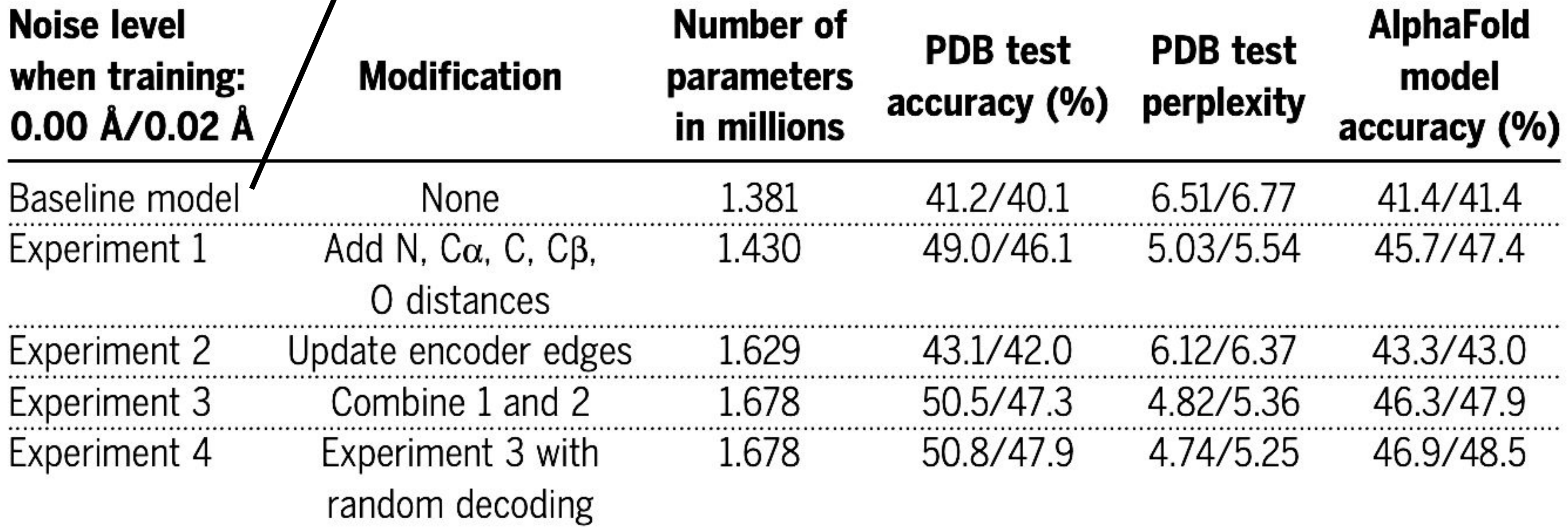
- Protein assemblies in PDB (X-ray or cryoEM)
- Random train, validation, test split (23358/1464/1529)
  - Different chains from one protein must be in same group
- Training Epoch
  1. Pick query sequence from training set
  2. For the query, pick a conformation
- Loss: masked negative log likelihood
- Evaluation: accuracy, perplexity, run time

Results

(In-silico and experimental validation of ProteinMPNN)

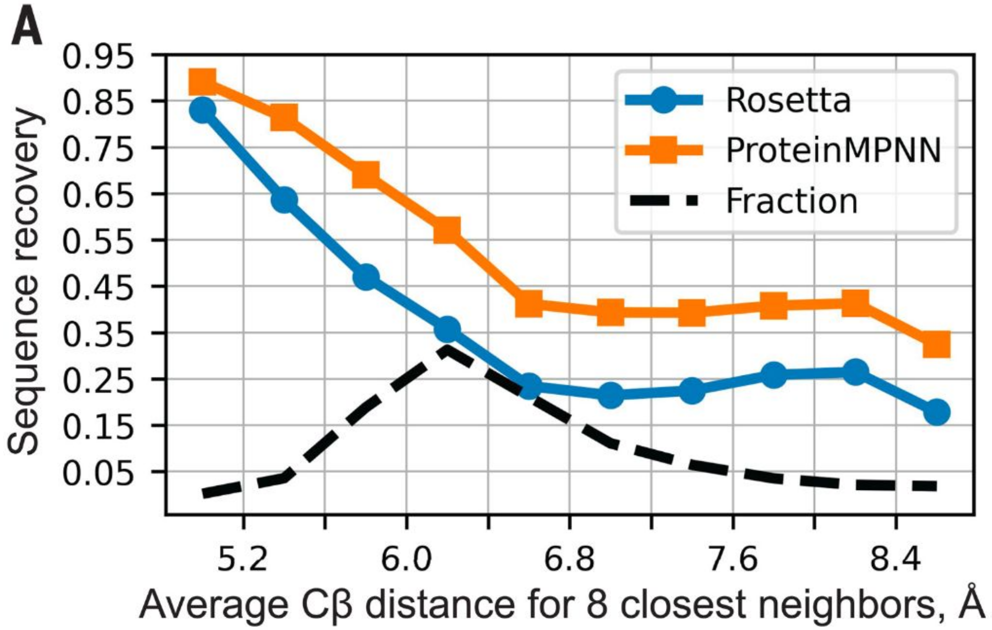
Adding atomic distances as additional input features boosted performance

Ingraham et. al



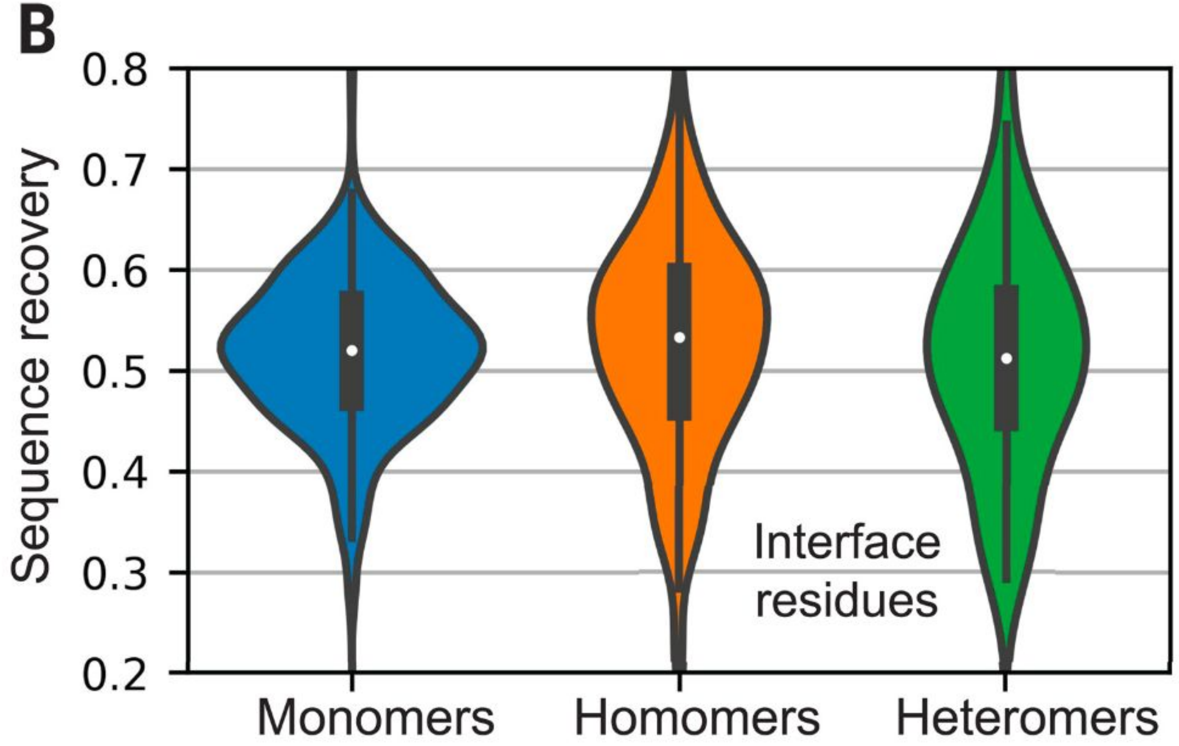
Noise level when training: 0.00 Å/0.02 Å	Modification	Number of parameters in millions	PDB test accuracy (%)	PDB test perplexity	AlphaFold model accuracy (%)
Baseline model	None	1.381	41.2/40.1	6.51/6.77	41.4/41.4
Experiment 1	Add N, Cα, C, Cβ, O distances	1.430	49.0/46.1	5.03/5.54	45.7/47.4
Experiment 2	Update encoder edges	1.629	43.1/42.0	6.12/6.37	43.3/43.0
Experiment 3	Combine 1 and 2	1.678	50.5/47.3	4.82/5.36	46.3/47.9
Experiment 4	Experiment 3 with random decoding	1.678	50.8/47.9	4.74/5.25	46.9/48.5

ProteinMPNN had higher overall native sequence recovery than Rosetta



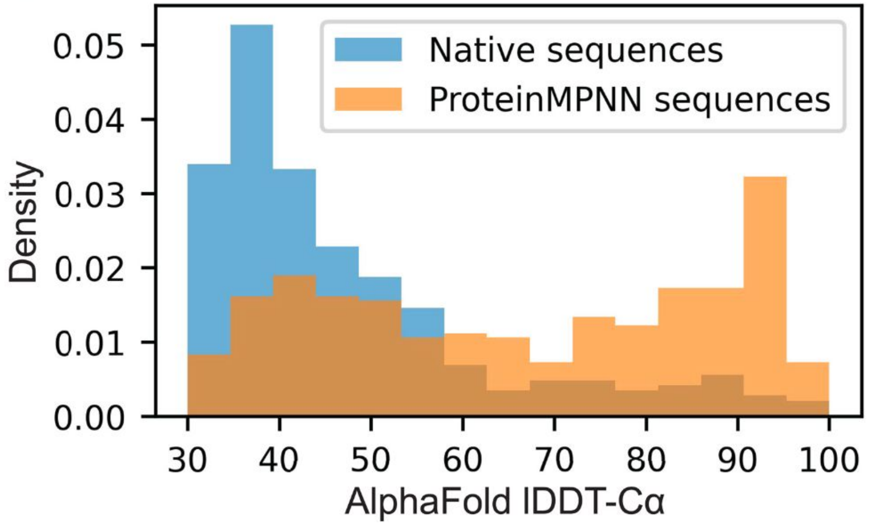
- **Sequence recovery**: 54.29% vs. 32.9%
- **Run time**: 1.2s vs. 258.8s (1 CPU for 100 residues)

ProteinMPNN performs well for different protein categories

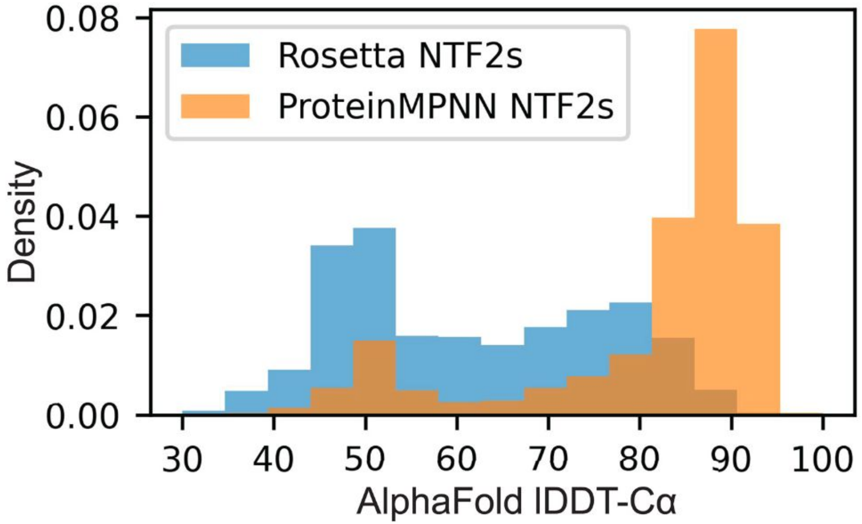


ProteinMPNN sequences predict native structures more effectively than native sequences

E



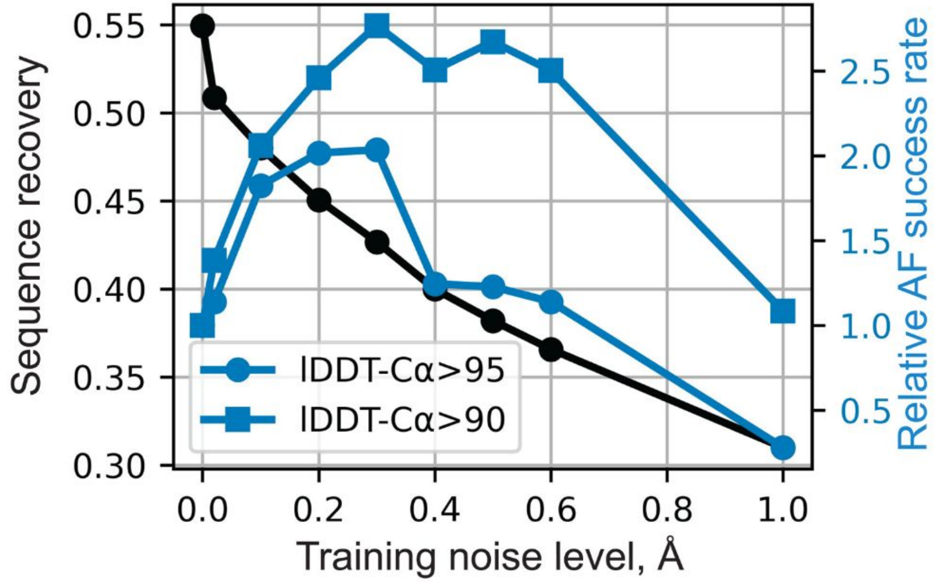
F



- Used AlphaFold to generate structures based native sequence (no MSA) and ProteinMPNN generated sequences
- Used AlphaFold to generate structures based Rosetta and ProteinMPNN sequences

Adding noise helps inference of ProteinMPNN sequences

C

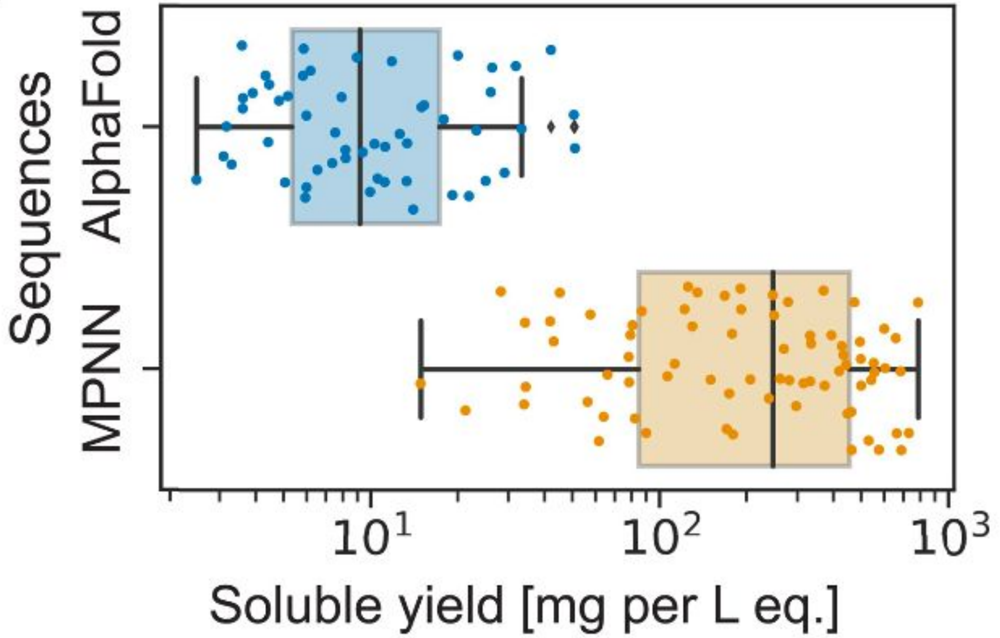


- Noise allows model to focus more on overall topological instead of local features
- More representative of real-world where true structure is not known at atomic resolution (aim is not necessarily to maximize sequence recovery)

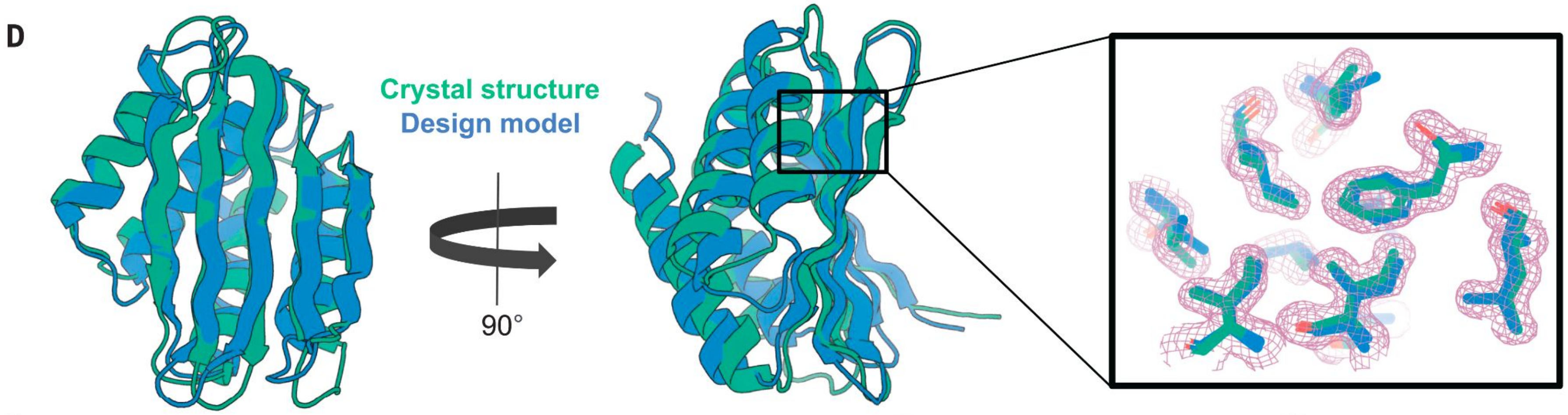
Experimental validation

1. Network hallucination by AlphaFold to produce backbone set
2. Monte Carlo to generate variety of AlphaFold sequences
3. ProteinMPNN to generate sequences
4. Express these proteins in E. coli

A



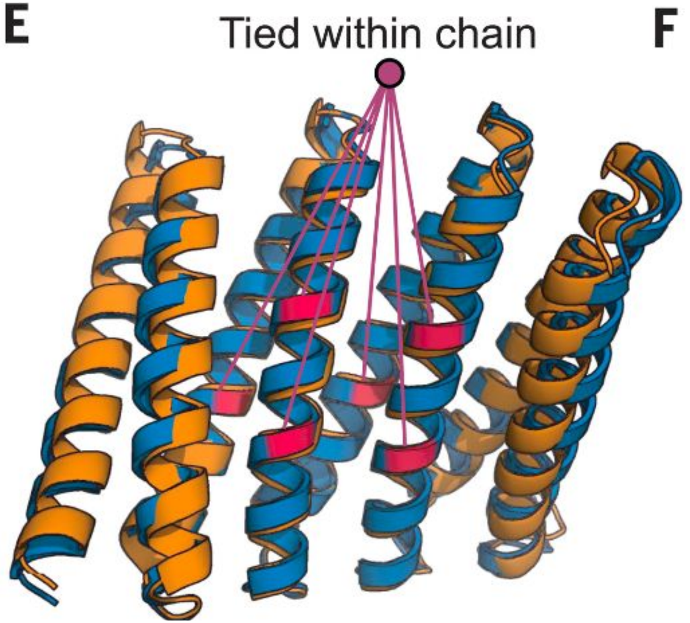
ProteinMPNN can solve crystal structure of proteins with “difficult” folds



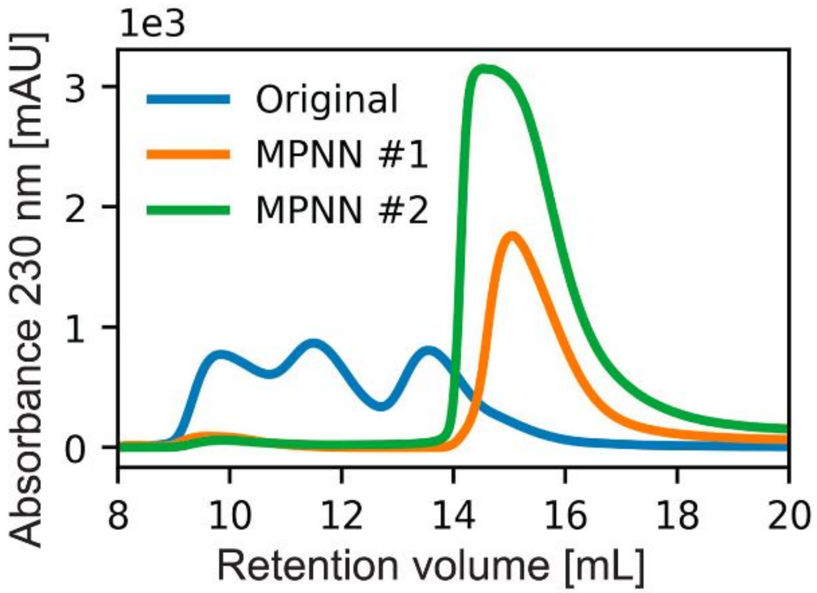
- Design of a difficult protein structure with difficult fold structure (TM-score: 0.56)
- Crystal side chains in green, MPNN in blue
- Very close match suggests MPNN can design accurate sequences robustly

ProteinMPNN can design structural repeats more accurately than Rosetta

E



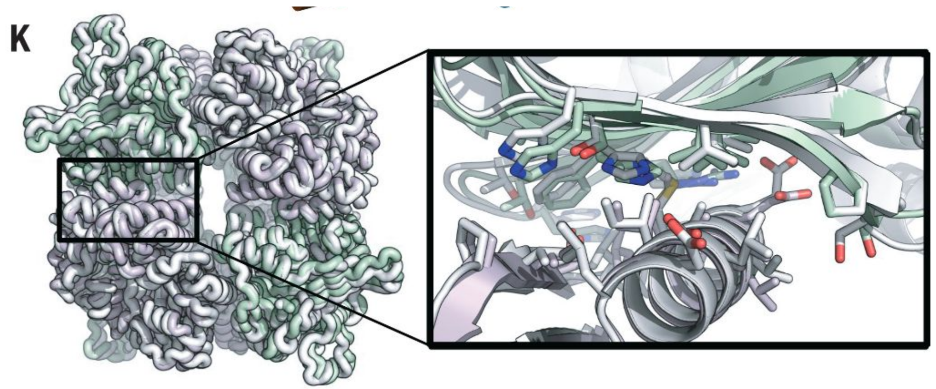
F



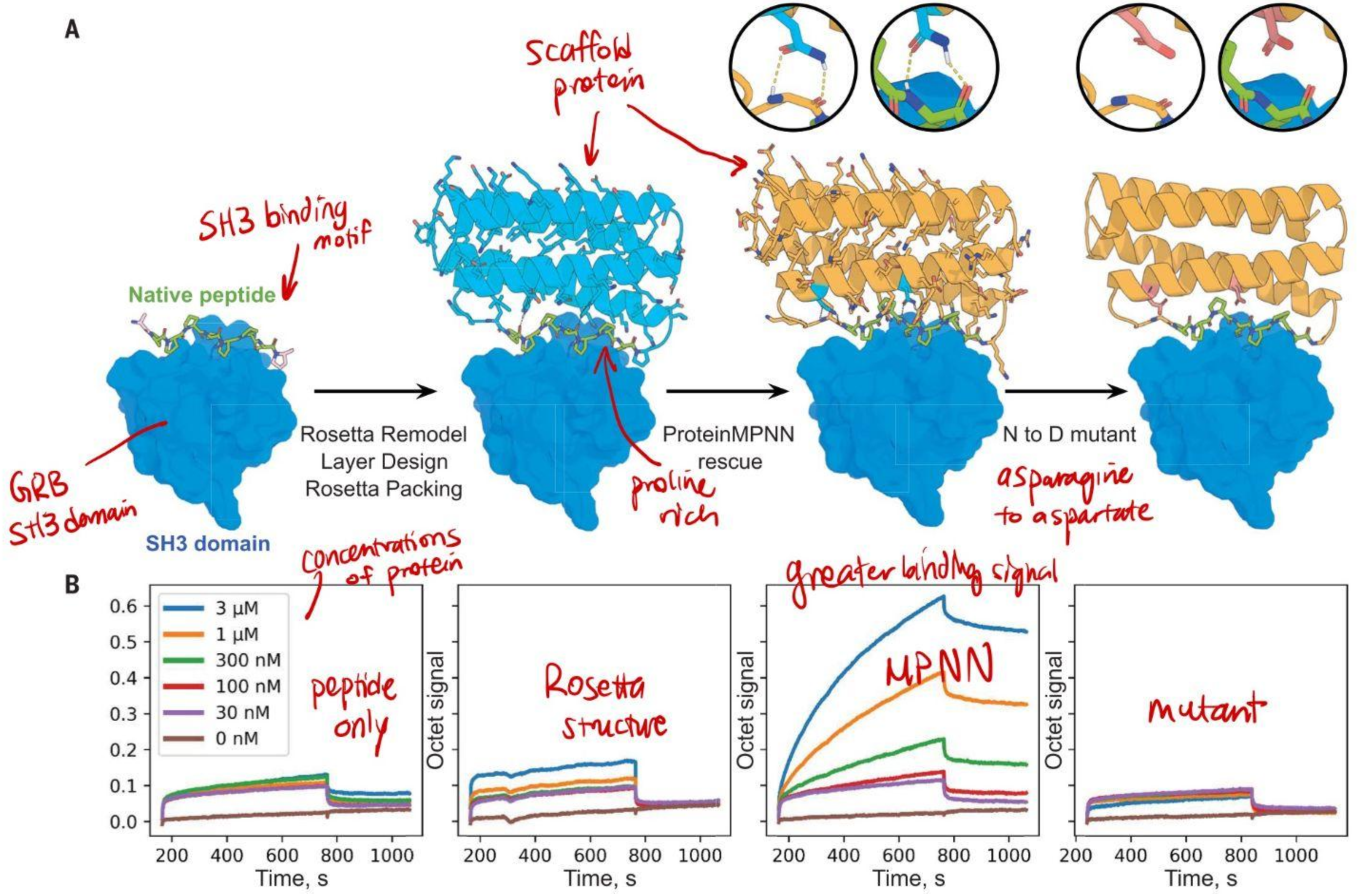
- Orange is backbone design and blue is MPNN
- Rosetta generates protein with many different components, whereas MPNN has one component (peak)

ProteinMPNN can rescue tetrahedral assemblies that Rosetta failed to design

- Designed 76 sequences spanning 27 of tetrahedral nanoparticle backbones
- Express in E-coli: 13 designs formed assemblies including new tetrahedral assemblies failed using Rosetta
- Close match for an example tetrahedral assembly (gray is crystal, green and purple is MPNN)



ProteinMPNN can rescue protein functions that Rosetta failed to design



Conclusion

Summary

- Protein design: finding optimal sequence from structure (inverse folding)
- Limitations of physics-based approaches (e.g., Rosetta)
- ProteinMPNN uses message passing, flexible decoding, tied positions
- High sequence recovery, rescued failed designs and functions, fast
- Robustness and efficiency is promising for protein design

Q&A