Learning inverse folding from millions of predicted structures

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, Alexander Rives, ICML 2022

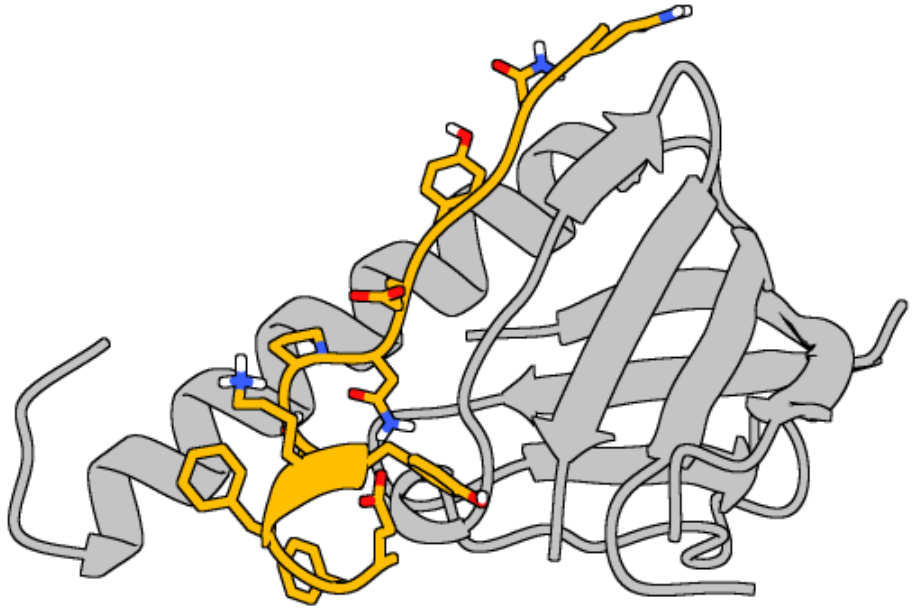
Presented by Joseph Clark

Outline

Motivation
- Why do we care about protein design?
- What is inverse folding?

Methods
- Problem statement
- Model architecture
- Training data

Results
- Fixed backbone sequence design
- Predicting multiple conformations
- Zero-shot binding affinity prediction



Why do we care about protein design?

Proteins are machines that perform nearly all biological functions

Protein design repurposes natures tools for our own needs

New Results

Follow this preprint Previous

De-novo design of a random protein walker

Posted January 09, 2026.

Liza Ulčakar, Hao Shen, Eva Rajh, Tadej Satler, Federico Olivieri, Joseph Watson, Yang Hsia, Justin Decarreau, Eric Lynch, Justin Kollman, David Baker, Ajasja Ljubetič

doi: https://doi.org/10.1101/2025.09.29.677966

This article is a preprint and has not been certified by peer review [what does this mean?].

Download PDF

Print/Save Options

Data/Code

Revision Summary

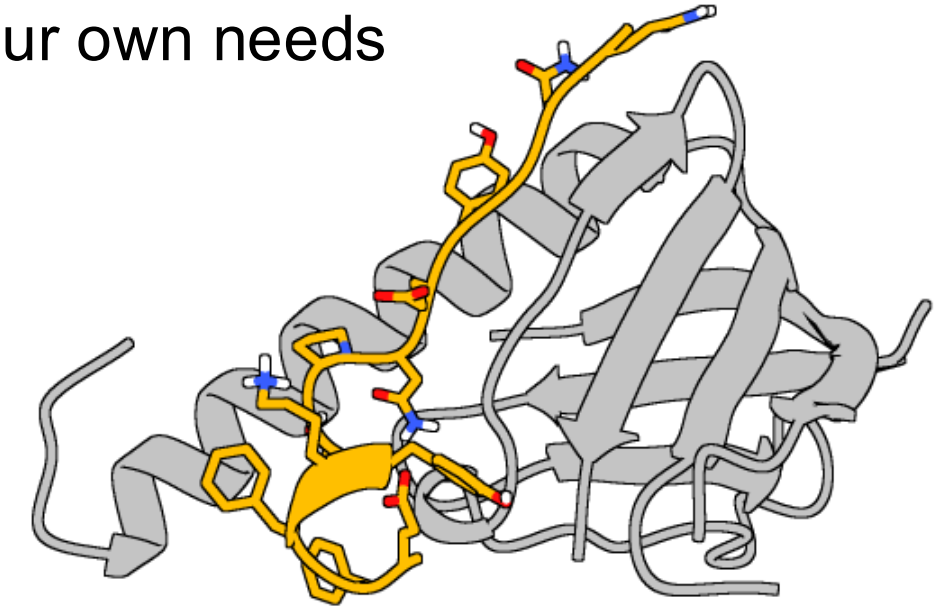
Abstract Full Text Info/History Metrics Preview PDF

Article | Open access | Published: 18 December 2023

De novo design of high-affinity binders of bioactive helical peptides

Susana Vázquez Torres, Philip J. Y. Leung, Preetham Venkatesh, Isaac D. Lutz, Fabian Hink, Huu-Hien Huynh, Jessica Becker, Andy Hsien-Wei Yeh, David Juergens, Nathaniel R. Bennett, Andrew N. Hoofnagle, Eric Huang, Michael J. MacCoss, Marc Expòsit, Gyu Rie Lee, Asim K. Bera, Alex Kang, Joshmyn De La Cruz, Paul M. Levine, Xinting Li, Mila Lamb, Stacey R. Gerben, Analisa Murray, Piper Heine, ... David Baker

+ Show authors



RESEARCH ARTICLE | BIOCHEMISTRY

Targeted degradation of α-Synuclein using an evolved botulinum toxin protease

Philipp Sondermann, Christian S. Diercks, Cynthia Rong, and Peter G. Schultz Authors Info & Affiliations

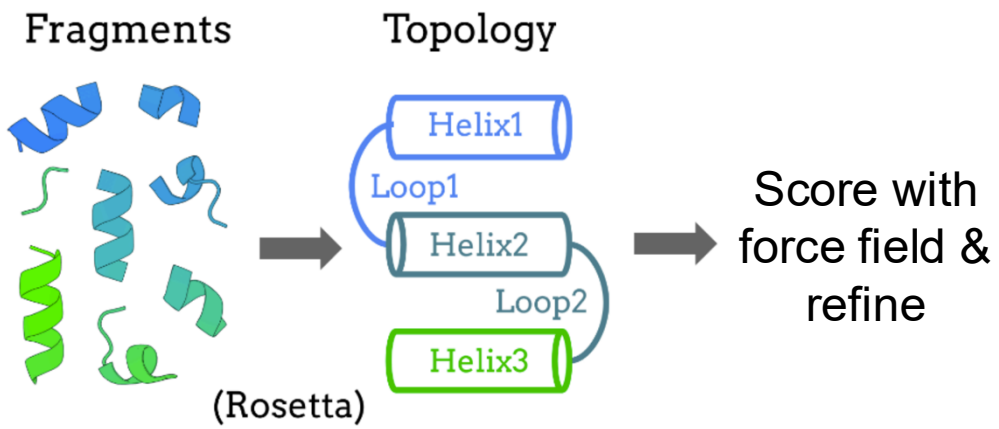
Contributed by Peter G. Schultz; received December 20, 2024; accepted February 24, 2025; reviewed by Peter S. Kim and Kevan M. Shokat

March 24, 2025 | 122 (13) e2426745122 | https://doi.org/10.1073/pnas.2426745122

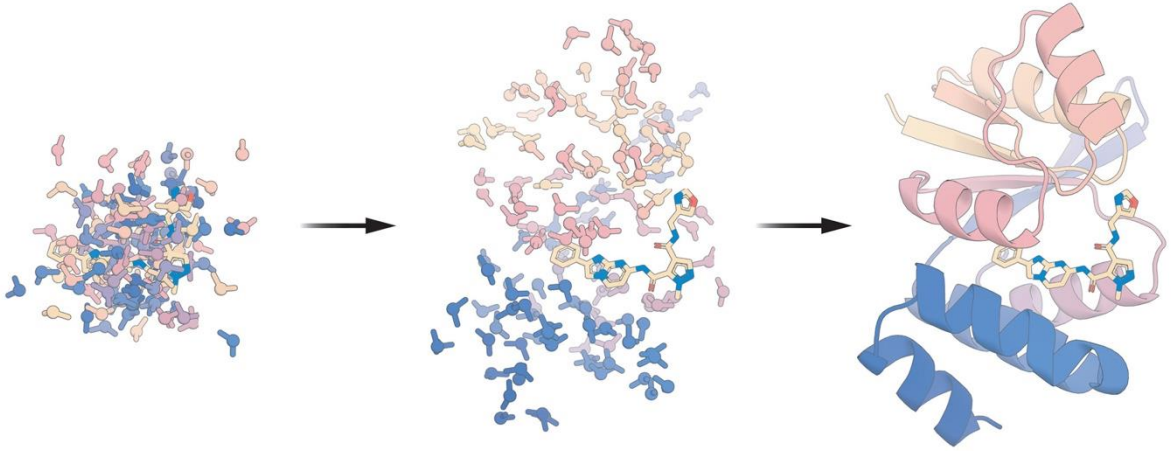
de novo protein design

Goal: create new proteins from scratch rather than modifying existing ones

Traditional: physics-based modeling



Recent: deep generative models



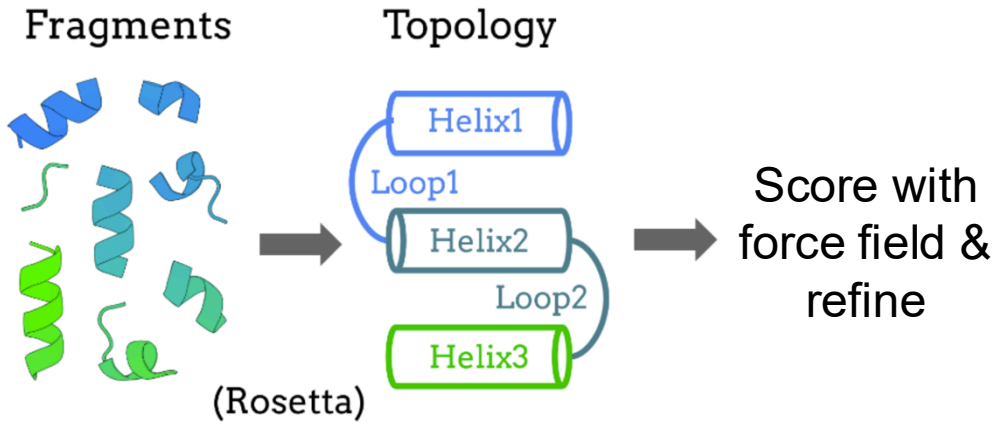
Matsubara et. al., *bioRxiv*, 2025; Krishna et. al., *Science,* 2024

de novo protein design

Goal: create new proteins from scratch rather than modifying existing ones

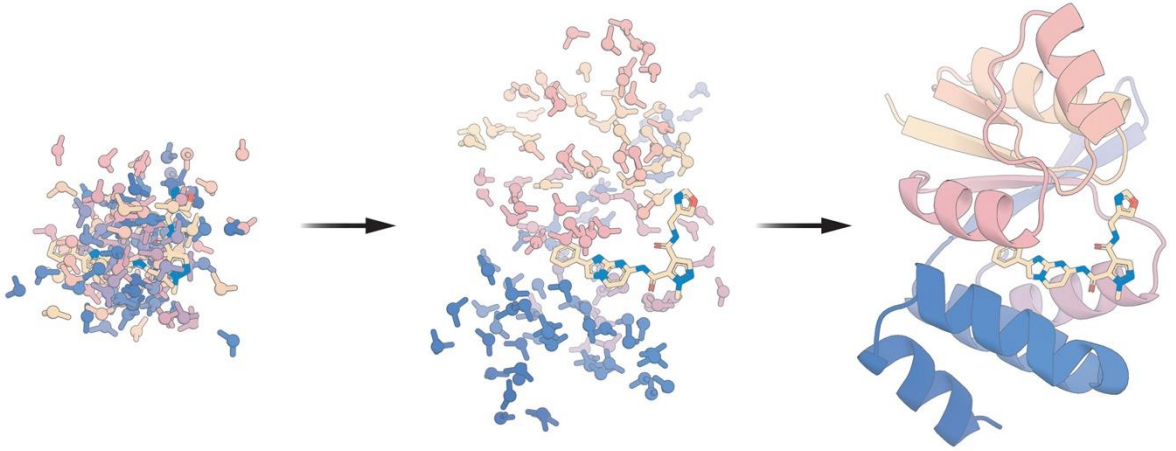
Can predicted structures augment protein design?

Traditional: physics-based modeling



Recent: deep generative models

! Data hungry

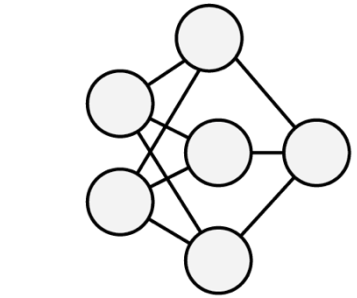


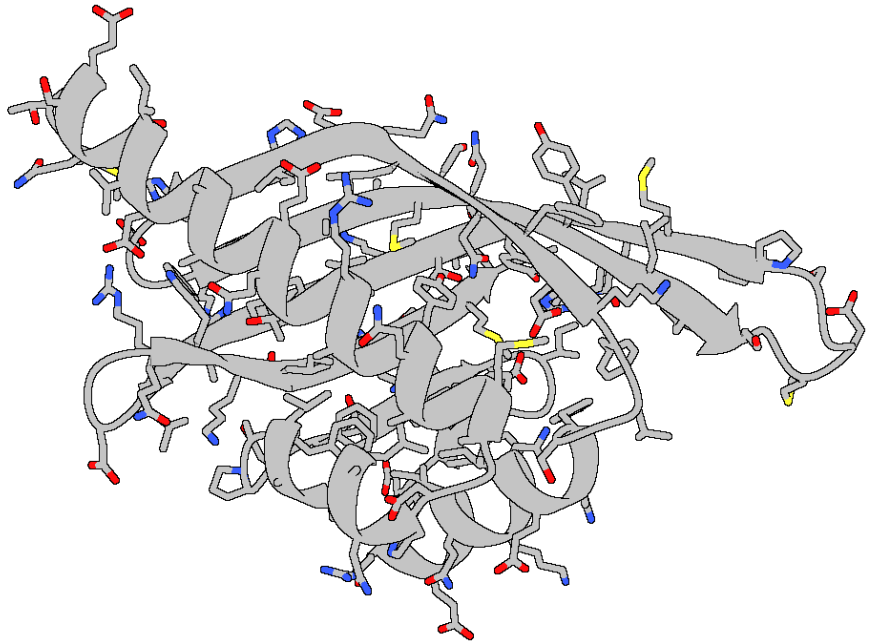
Matsubara et. al., *bioRxiv*, 2025; Krishna et. al., *Science,* 2024

Structure prediction vs inverse folding

Structure prediction: determine the native structure of a protein from sequence

GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGELAQRLLKDIAGSG
KDAILKAYDVIAYLANVCNKE
QNVIDYFGFLADYDLSA…





Protein sequence

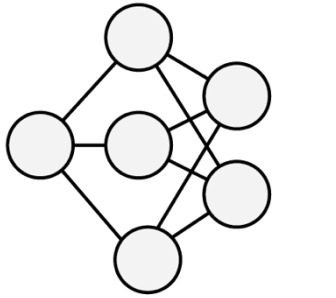
All-atom protein structure

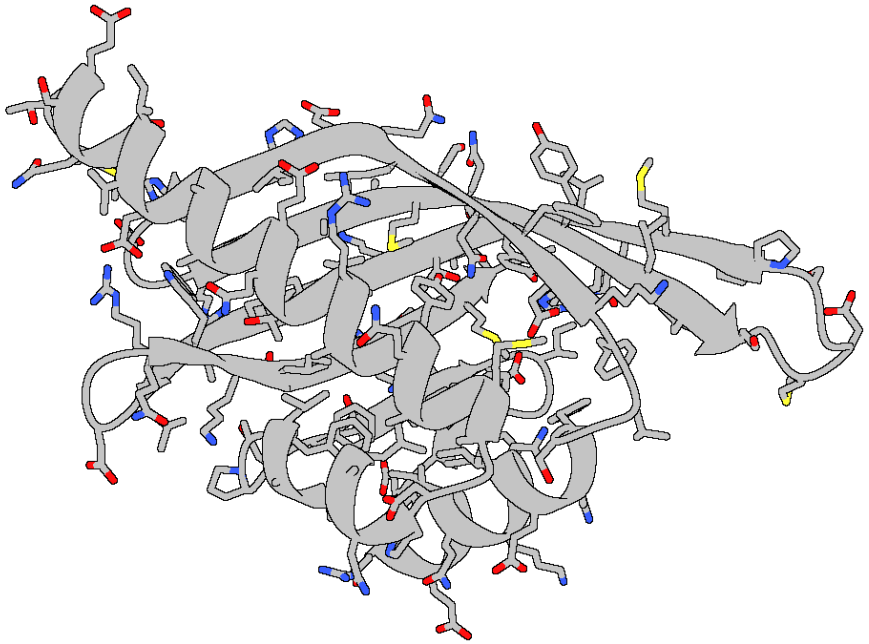
Structure prediction vs inverse folding

Structure prediction: determine the native structure of a protein from sequence

Inverse folding: design a sequence that folds into a desired structure

GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGELAQRLLKDIAGSG
KDAILKAYDVIAYLANVCNKE
QNVIDYFGFLADYDLSA…





Protein sequence

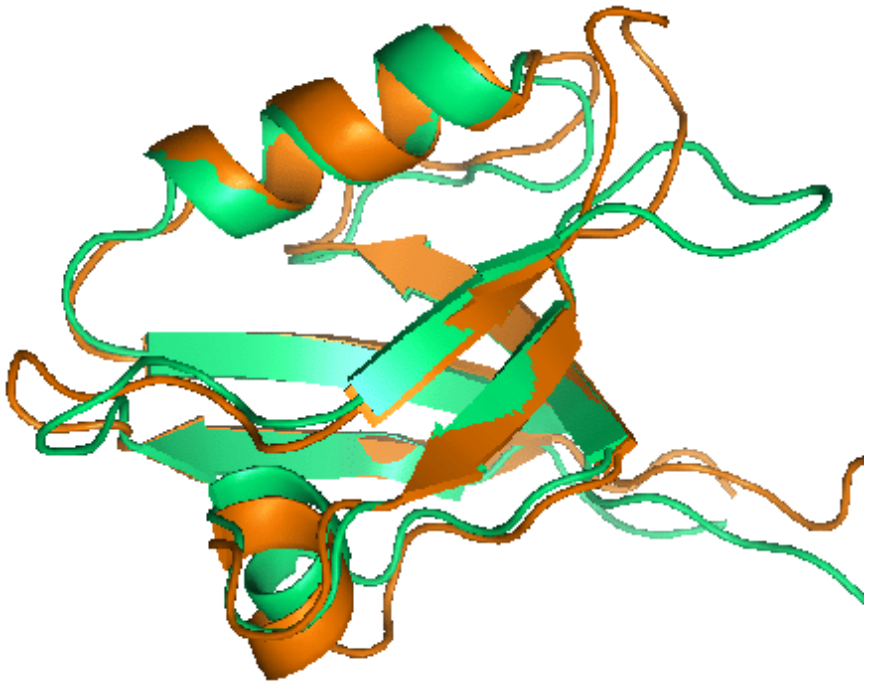
All-atom protein structure

Many sequences share similar structures

Example: protein superfamilies
- High structural similarity
- Low sequence identity

Structure is more conserved than sequence

Why?



```
RMLPRLCCLEKGPNGYGFHLHGEKG---KLGQYIRLVEPGSPAE-KAGLLAGDRLVEVNGENVEKETHQQVVSRIRAALNAVRLLVVDPETDEQL----
!!!!!!!!!!!.!!!!!!!!!.     .!!!!!!!!!!!!!!! !.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!..
GAIIYTVELKRYGGPLGITISGT--EEPFDPIIISSLTKGGLAERTGAIHIGDRILAINSSSLKGKPLSEAIHLLQMAGETVTLKIKKQTDAQ--PASS
```

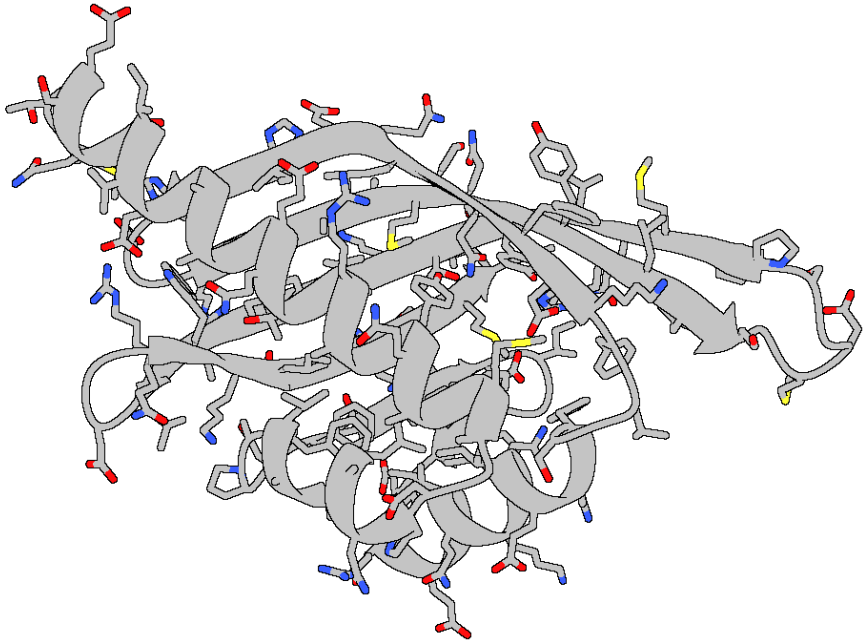
2.03Å RMSD
15% sequence identity

Joo et. al., 2011

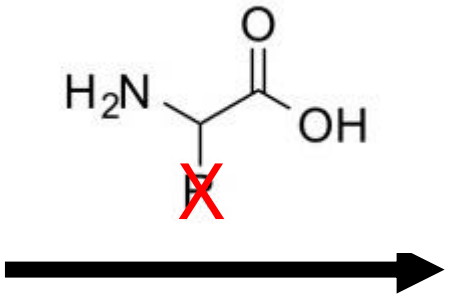
Structure prediction vs inverse folding

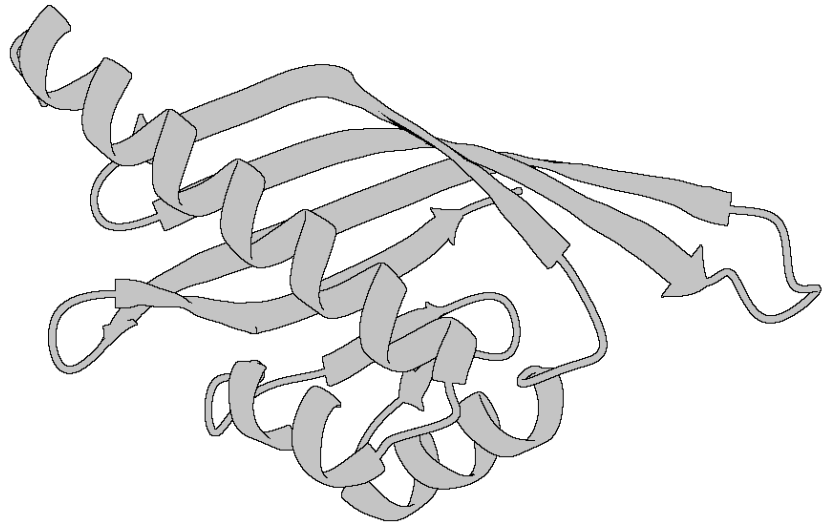
Structure prediction: determine the native structure of a protein from sequence

Inverse folding: design a sequence that folds into a desired structure



All-atom protein structure



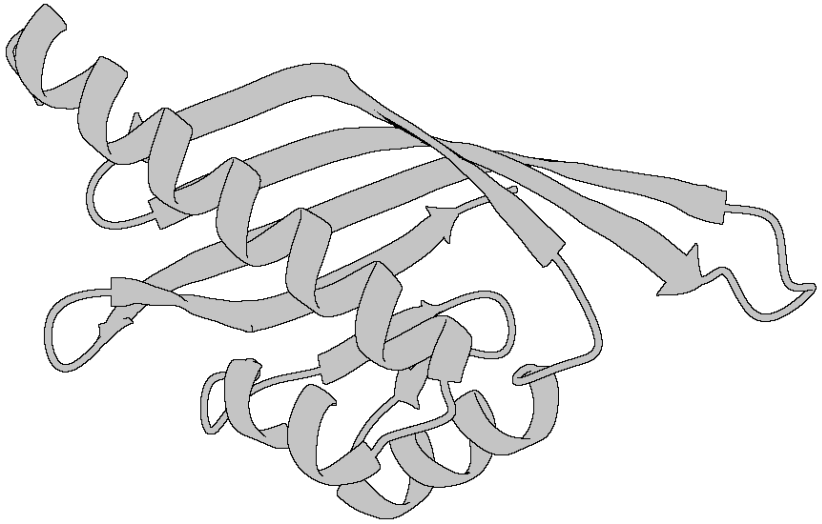


Backbone atoms only

Structure prediction vs inverse folding

Structure prediction: determine the native structure of a protein from sequence

Inverse folding: design a sequence that folds into a desired structure



P(seq | structure)

GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGELAQRLLKDIAGSG
KDAILKAYDVIAYLANVCNKE
QNVIDYFGFLADYDLSA…

Formal statement of inverse folding

Autoregressive sequence generation conditioned on backbone structure:

$$p(Y|X) = \prod_{i=1}^{n} p(y_i|y_{i-1}, \ldots, y_1; X)$$

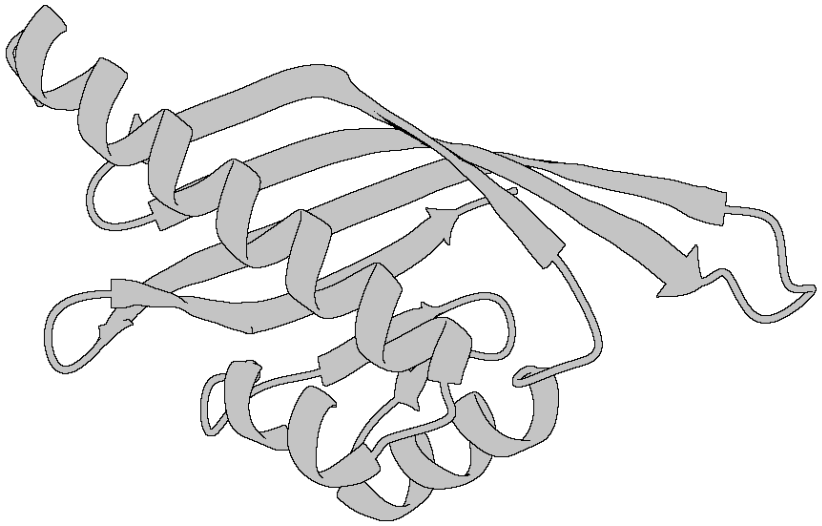
Sequence Backbone structure

Formal statement of inverse folding

Autoregressive sequence generation conditioned on backbone structure:

$$p(Y|X) = \prod_{i=1}^{n} p(y_i|y_{i-1}, \ldots, y_1; X)$$

Sequence Backbone structure



P(seq | structure)

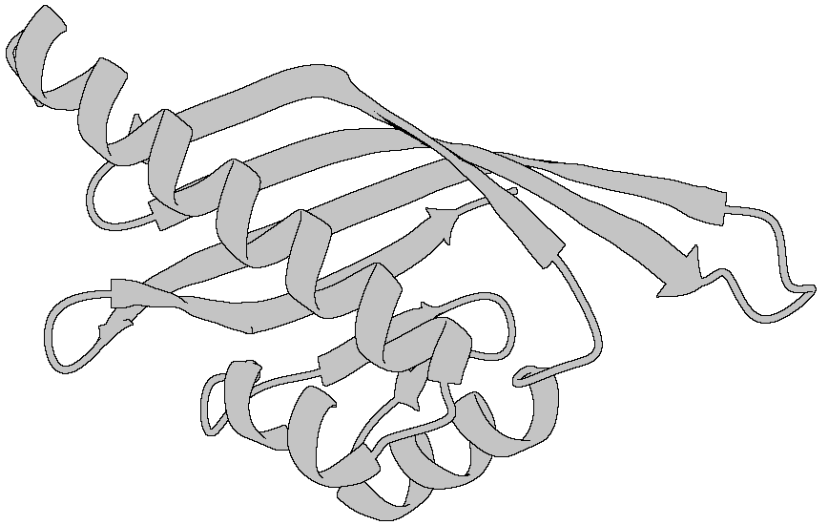
GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGEL**A**_

Formal statement of inverse folding

Autoregressive sequence generation conditioned on backbone structure:

$$p(Y|X) = \prod_{i=1}^{n} p(y_i|y_{i-1}, \ldots, y_1; X)$$

Sequence Backbone structure



P(seq | structure)

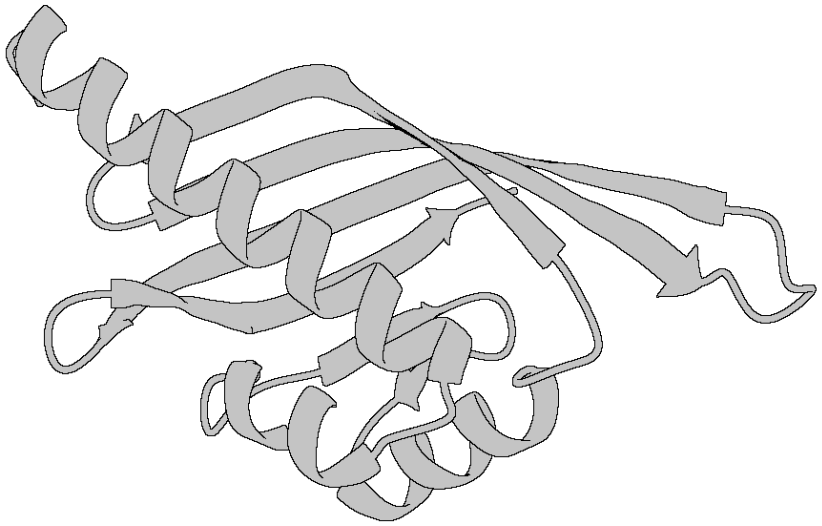
GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGELA**Q**_

Formal statement of inverse folding

Autoregressive sequence generation conditioned on backbone structure:

$$p(Y|X) = \prod_{i=1}^{n} p(y_i|y_{i-1}, \ldots, y_1; X)$$

Sequence Backbone structure



P(seq | structure)

GSELETAMETLINVFHAHSG
KEGDKYKLSKKELKELLQTE
LSQPSGELAQ**R**_

Formal statement of inverse folding

Autoregressive sequence generation conditioned on backbone structure:

$$p(Y|X) = \prod_{i=1}^{n} p(y_i|y_{i-1}, \ldots, y_1; X)$$

Sequence Backbone structure

What are some limitations of describing the problem this way?

Augmenting inverse folding with predicted structures

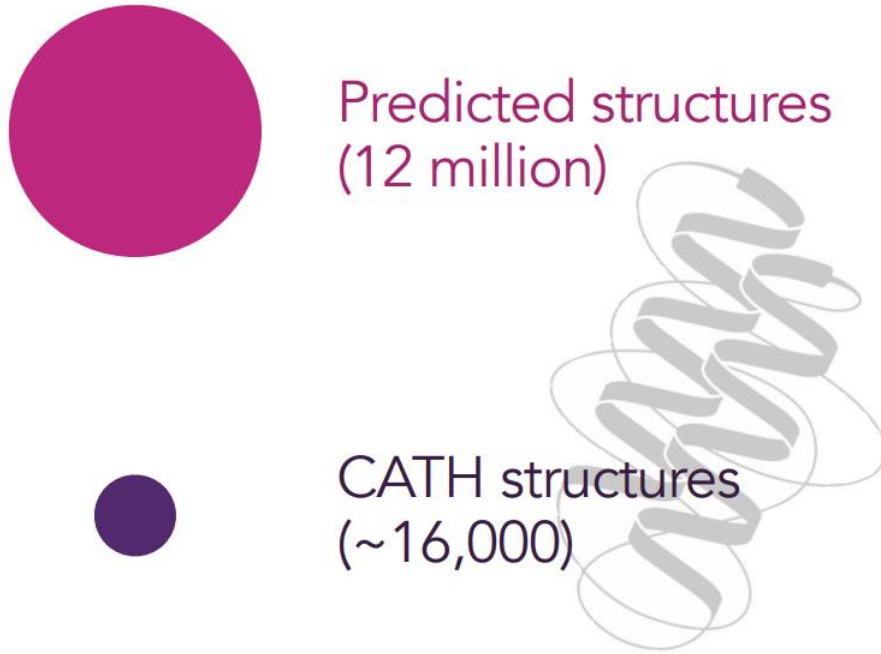
Inverse folding is limited by lack of experimental structures

Solution: generate 12M predicted structures

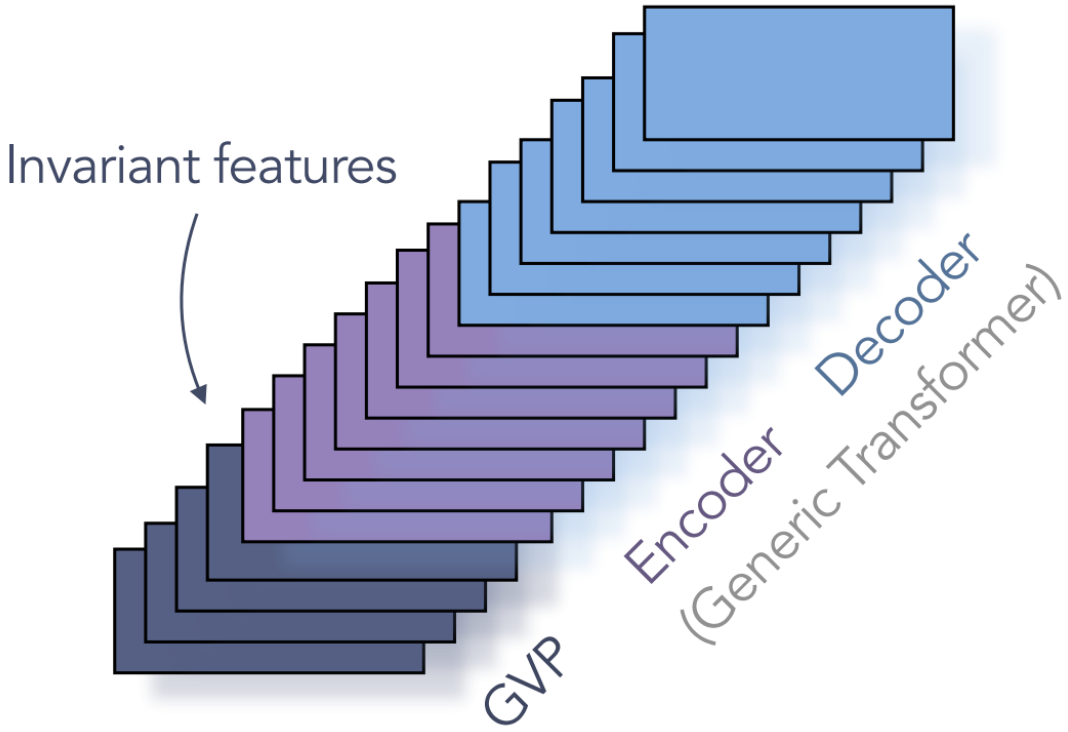
1:80 experimental:predicted structures
in each training epoch

Test set: structurally held-out subset of CATH

Randomly mask 15% of backbone coordinates
in addition to all side chain coordinates



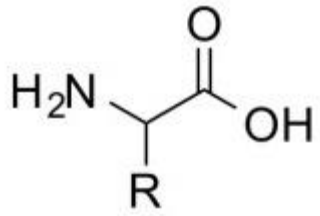
Model architecture



Geometric vector perceptron (GVP) layers

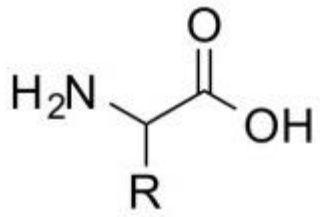
GVP: An alternative to multi-layer perceptron specialized for graph data

MLPs normally encode residues with ‘scalar features’



0.2, 0.9...0.2, 0.1, 1.3 ← Each feature is a scalar

GVP layers process both scalar features and vector features



There are both scalar and vector features

0.2, 0.9...0.2, 0.1, 1.3 , 

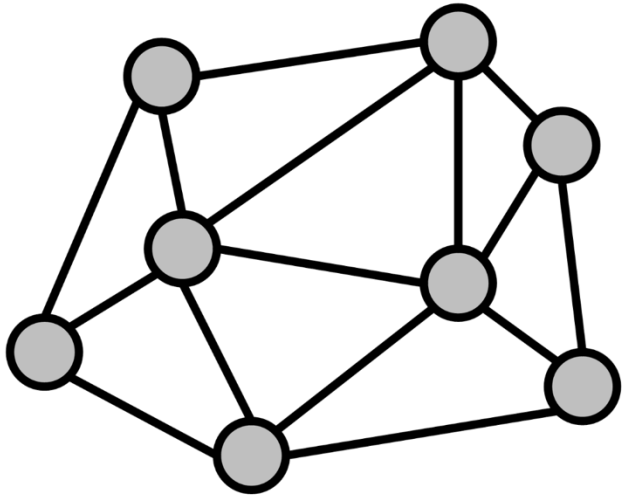
Jing et. al., *ICLR* 2021

Geometric vector perceptron (GVP) layers

GVP: An alternative to multi-layer perceptron specialized for graph data

Input KNN graph: Each node is a residue; each node has an edge connected to its *k*-nearest neighbors in 3D space

Nodes and edges have both scalar and vector features



Protein KNN graph

Jing et. al., *ICLR* 2021

Geometric vector perceptron (GVP) layers

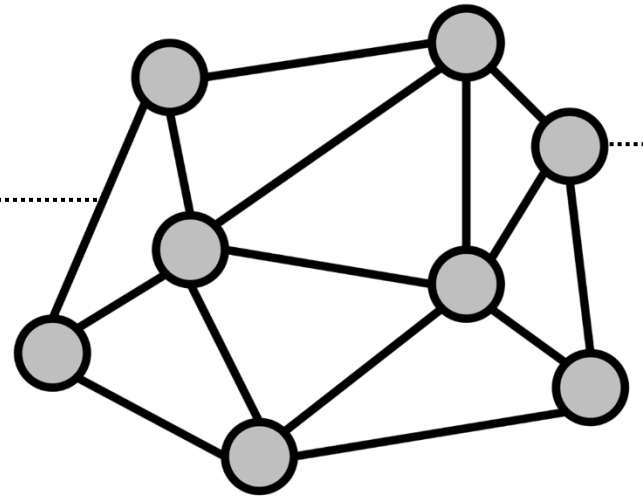
GVP: An alternative to multi-layer perceptron specialized for graph data

Input KNN graph: Each node is a residue; each node has an edge connected to its *k*-nearest neighbors in 3D space

Nodes and edges have both scalar and vector features

Edge (i, j) features
Scalar:
- Cαi to Cαj distance in 3D

Vector:
- Unit vector in direction Cαj − Cαi



Protein KNN graph

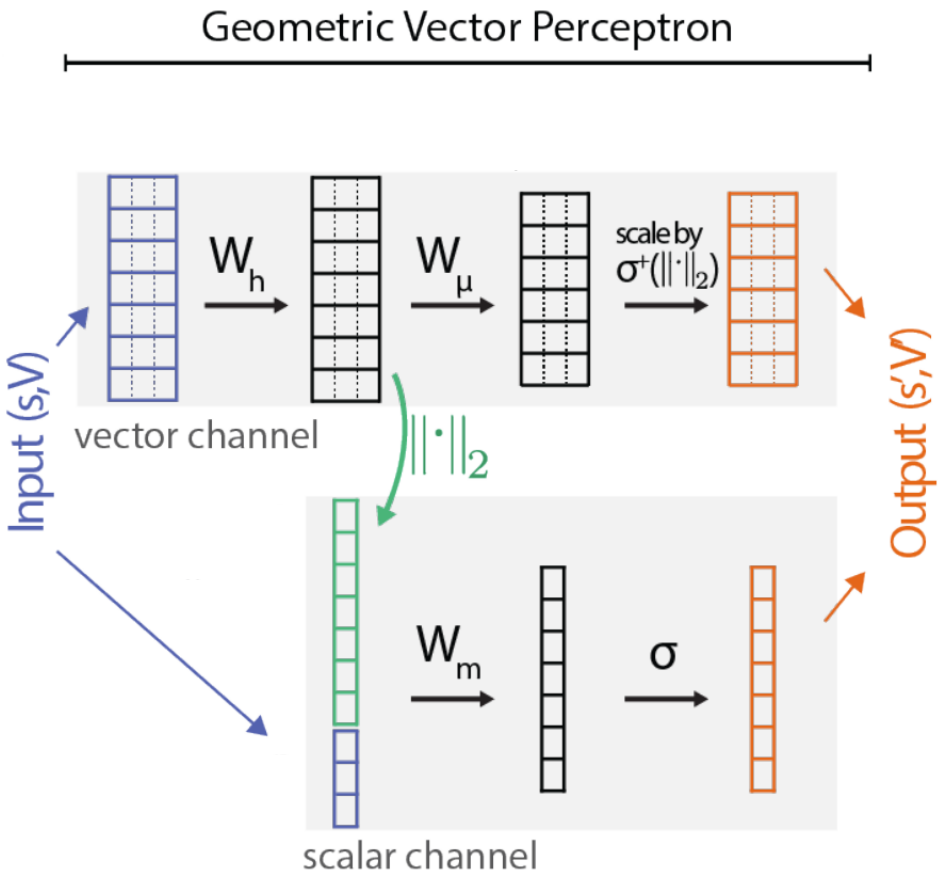
Node features
Scalar:
- Sin & Cos of φ, ψ, ω

Vector:
- Unit vectors in directions Cαi+1 − Cαi and Cαi−1 − Cαi

Geometric vector perceptron (GVP) layers

GVP layer: an alternative to multi-layer perceptron specialized for graph data

Input:
- Scalar features s
- Vector features V



Output:
- Updated scalar feats s'
- Updated vector feats V'

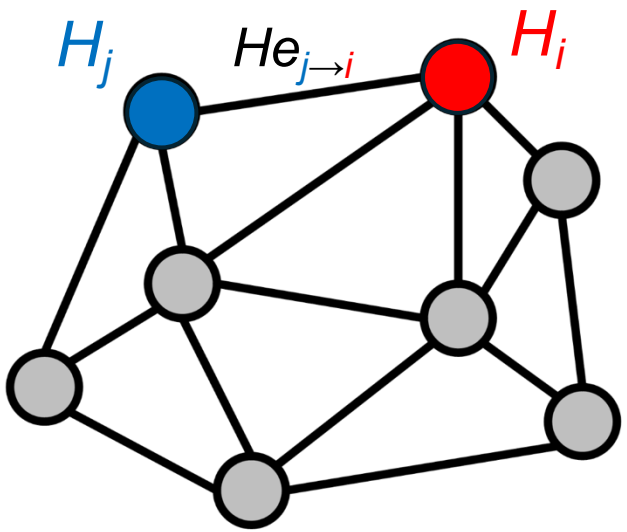
Jing et. al., *ICLR* 2021

Message passing with GVPs

Goal: update the features of each node using information from surrounding nodes

H_i = scalar and vector features of residue i

$He_{j \to i}$ = scalar and vector features of edge connecting residue j to residue i

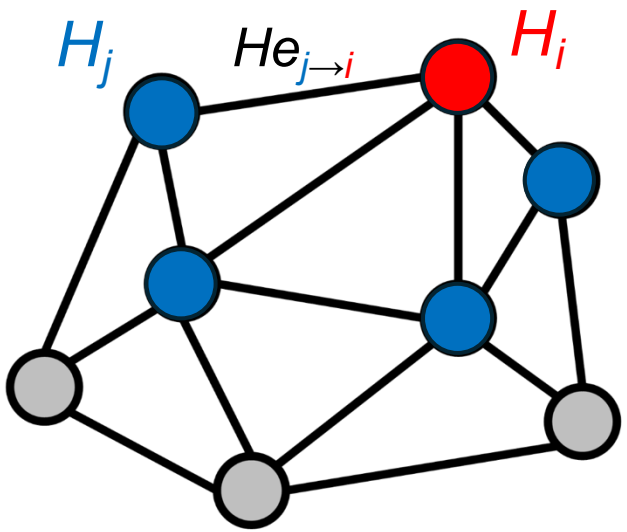


Jing et. al., *ICLR* 2021

Message passing with GVPs

Goal: update the features of each node using information from surrounding nodes

For all adjected residues *j*, compute GVP(H_j, $He_{j \to i}$) = $\mathbf{h}_m^{(j \to i)}$



Jing et. al., *ICLR* 2021

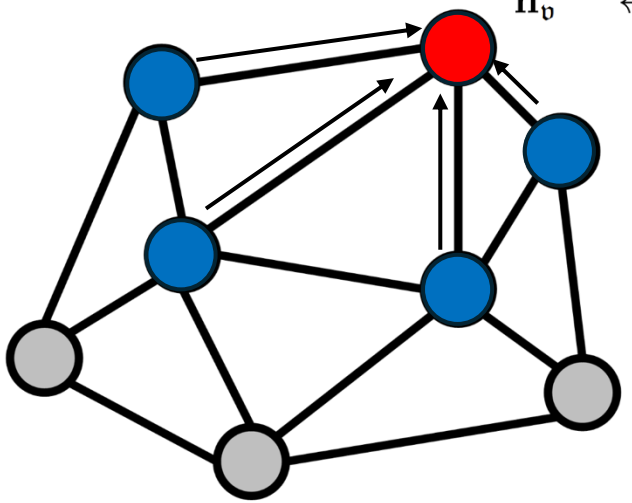
Message passing with GVPs

Goal: update the features of each node using information from surrounding nodes

For all adjected residues *j*, compute GVP(H_j, $He_{j \to i}$) = $\mathbf{h}_m^{(j \to i)}$

Average the contributions of each neighbor to update H_i's features

$$\mathbf{h}_{\mathfrak{v}}^{(i)} \leftarrow \text{LayerNorm}\left(\mathbf{h}_{\mathfrak{v}}^{(i)} + \frac{1}{k'}\text{Dropout}\left(\sum_{j:\mathbf{e}_{j\to i}\in\mathcal{E}} \mathbf{h}_m^{(j\to i)}\right)\right)$$



'Passing a message' from each neighbor

Jing et. al., *ICLR* 2021

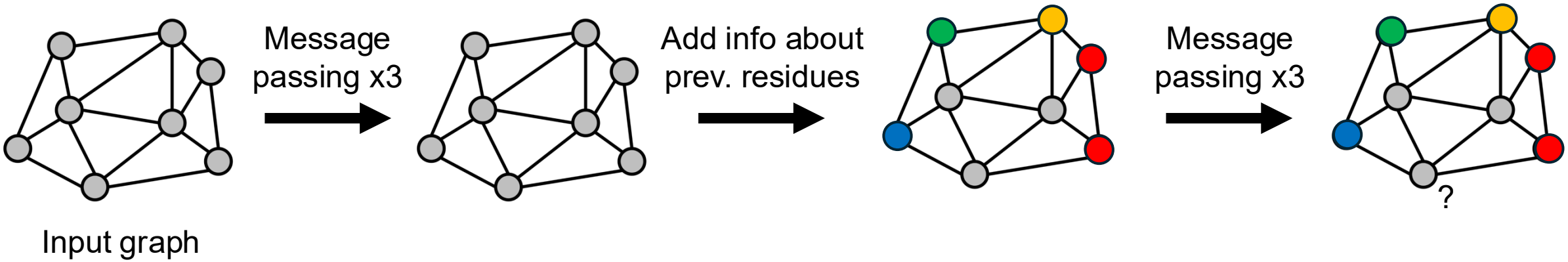
Entire GVP-GNN inverse folding architecture

Autoregressive encoder-decoder

1. Encoder: 3 message passing steps

2. Add sequence information to the graph (concat OHE to scalar features)

3. Decoder: 3 message passing steps and predict the next residue



GVP-Transformer architecture (ESM-IF1)

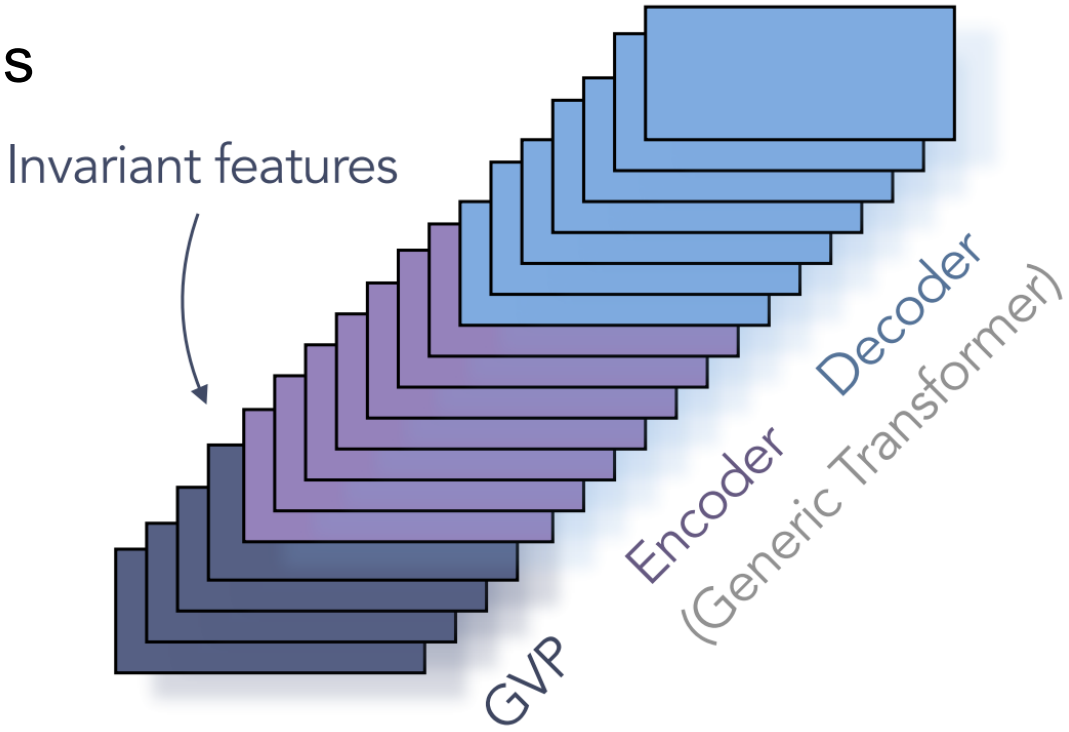
Autoregressive encoder-decoder

1. GVP encoder with 3 message passing steps

2. Add sequence information to the graph

3. Transformer encoder

4. Transformer decoder



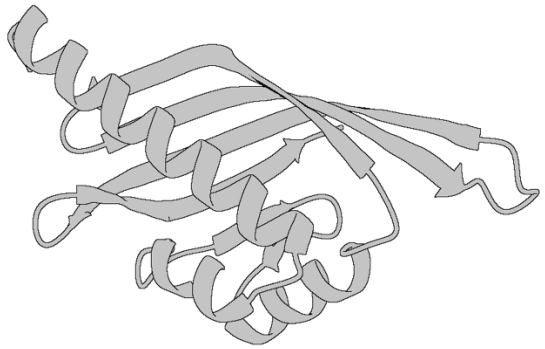
Model evaluation

Fixed backbone sequence design
- Predict sequence given all backbone coordinates
- Predict sequence with some coordinates masked
- Generalize to protein complexes
- Condition on multiple conformations

Zero-shot mutant effects
- Predict stability of sequence variants
- Predict binding affinity of complexes

Fixed backbone sequence design

Goal: Predict sequence given all backbone coordinates



P(seq | structure)

GSELETAMETLIN
VFHAHSGKEGDK
YKLSAKKKELK…

Performance metrics: model perplexity, % sequence recovery

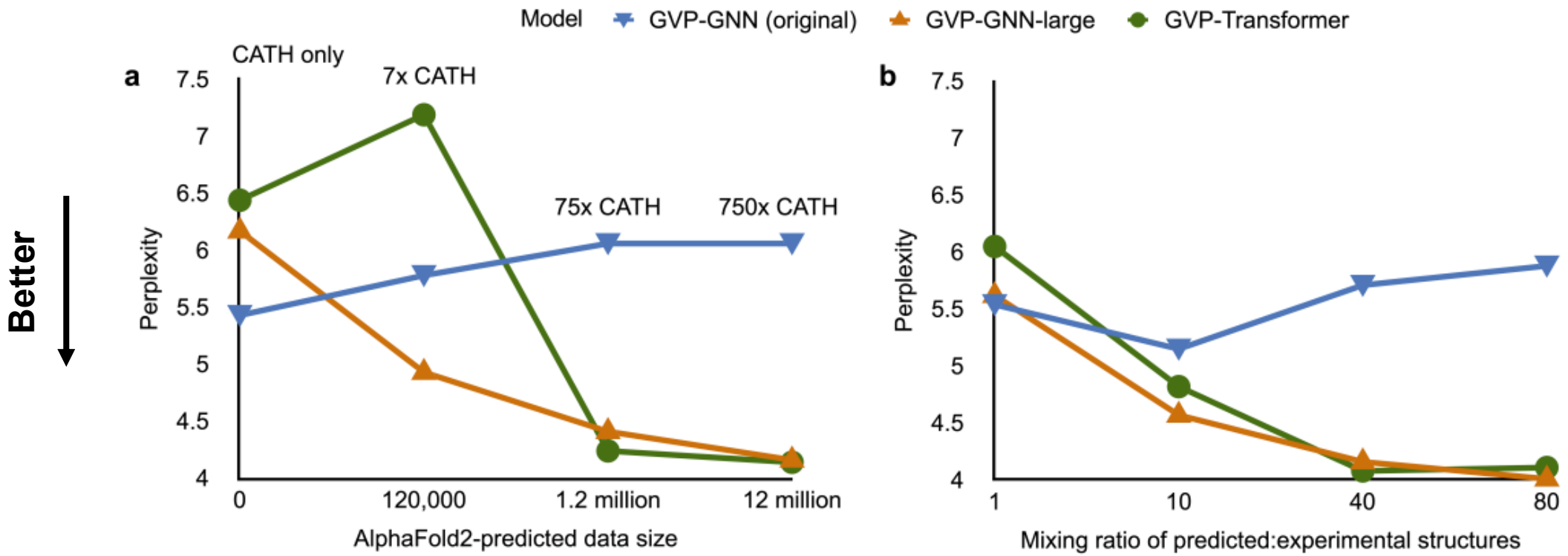
Is 51.6% a good accuracy?

		Perplexity			Recovery %		
Model	Data	Short	Single-chain	All	Short	Single-chain	All
Natural frequencies		18.12	18.03	17.97	9.6%	9.0%	9.5%
Structured GNN	CATH	7.91	6.48	6.49	31.5%	37.1%	37.1%
GVP-GNN	CATH	7.14	5.36	5.43	34.0%	42.7%	42.2%
	+ AlphaFold2	8.55	6.17	6.06	29.5%	38.2%	38.6%
GVP-GNN-large	CATH	7.68	6.12	6.17	32.6%	39.4%	39.2%
	+ AlphaFold2	6.11	4.09	4.08	**38.3%**	50.8%	50.8%
GVP-Transformer	CATH	8.18	6.33	6.44	31.3%	38.5%	38.3%
	+ AlphaFold2	**6.05**	**4.00**	**4.01**	38.1%	**51.5%**	**51.6%**

Training on predicted structures improves inverse folding

↑ predicted structures, ↑ performance

↑ ratio of predicted:experimental structures, ↑ performance

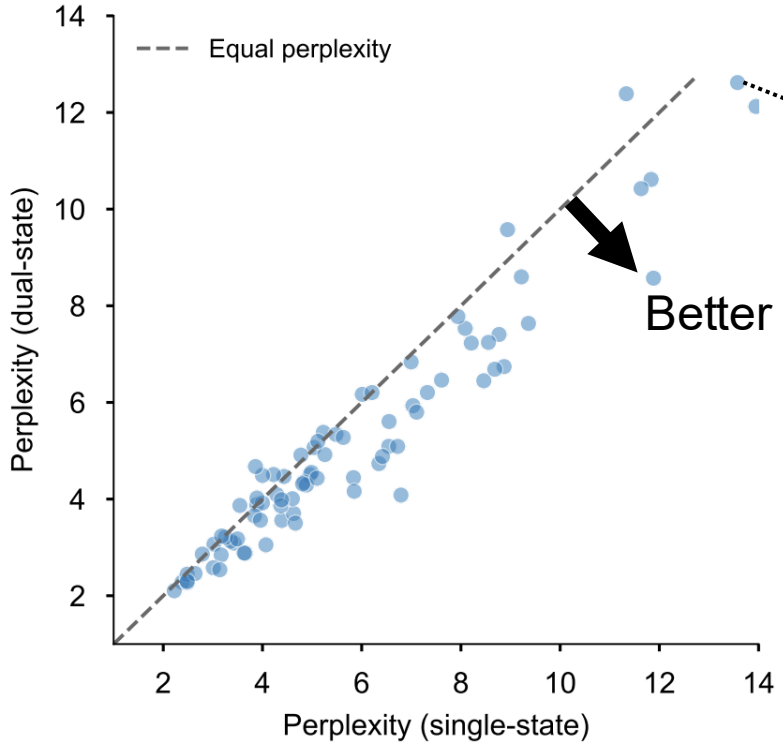


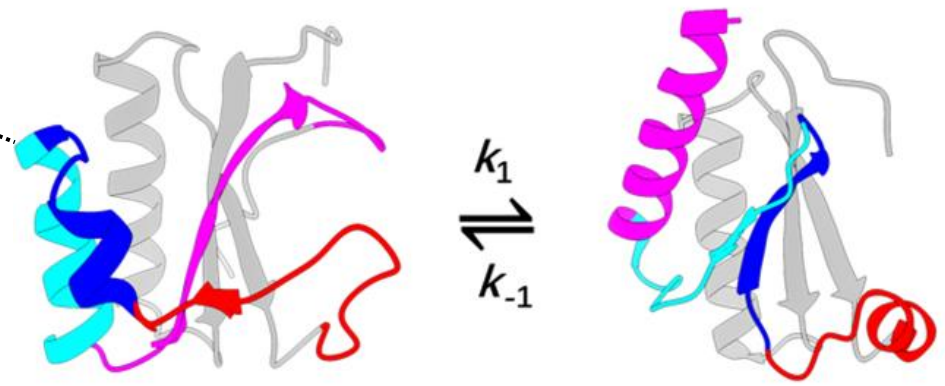
Designing multi-state proteins

Goal: design a sequence that is compatible with multiple conformations: *A, B*

Idea: $P(S \mid A, B) = (P(S \mid A) + P(S \mid B)) / 2$

Perplexity of $P(S \mid A, B)$ should be << Perplexity $P(S \mid A)$ or $P(S \mid B)$





Multistate protein: same sequence, different conformations

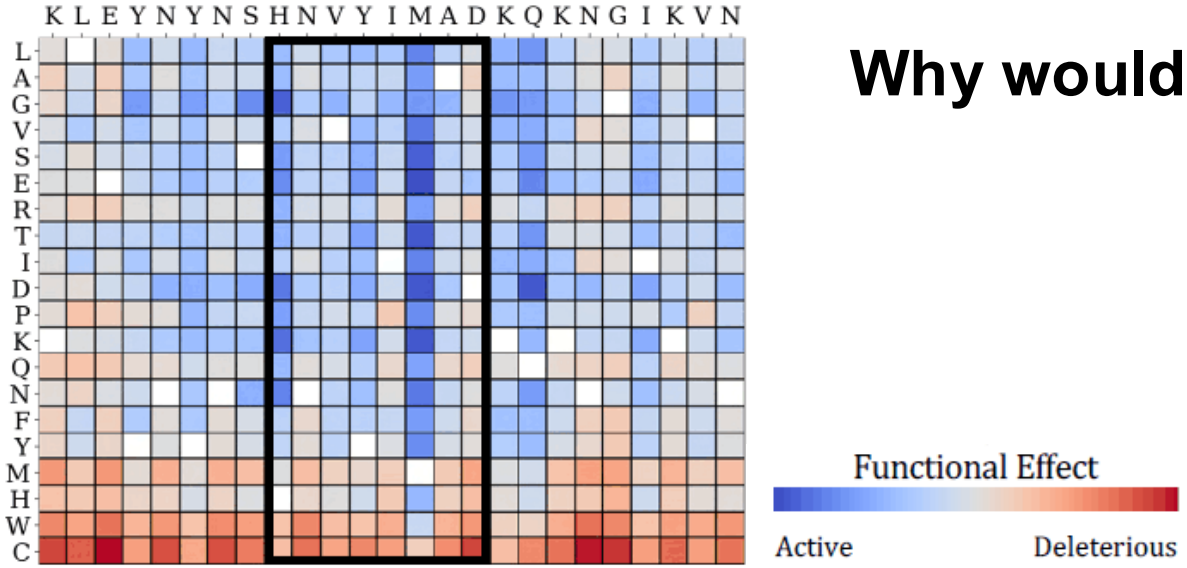
Zero-shot mutation effect prediction

Goal: given a WT protein, predict the stability of mutants

IRL stability = change in free energy upon mutation ($\Delta\Delta G$)

ESM-IF1 stability = *p(MT | backbone) / p(WT | backbone)*

Performance metric: Pearson correlation



Why would this work?

Zero-shot mutation effect prediction

Goal: given a WT protein, predict the stability of mutants

IRL stability = change in free energy upon mutation ($\Delta\Delta G$)

ESM-IF1 stability = *p(MT | backbone) / p(WT | backbone)*

Fold	Pearson correlation			
	Structured GNN (Ingraham et al., 2019)	GVP-GNN (Jing et al., 2021a)	GVP-GNN-large+AF2	GVP-Transformer+AF2
$\beta\beta\alpha\beta\beta_{37}$	0.47	0.53	0.62	**0.70**
$\beta\beta\alpha\beta\beta_{1498}$	**0.45**	0.39	0.37	0.33
$\beta\beta\alpha\beta\beta_{1702}$	0.12	**0.26**	0.24	0.22
$\beta\beta\alpha\beta\beta_{1716}$	0.47	0.57	**0.60**	0.58
$\alpha\beta\beta\alpha_{779}$	0.57	0.48	0.62	**0.64**
$\alpha\beta\beta\alpha_{223}$	0.36	0.47	**0.57**	0.55
$\alpha\beta\beta\alpha_{726}$	0.21	0.19	0.24	**0.26**
$\alpha\beta\beta\alpha_{872}$	0.23	0.39	0.38	**0.42**
$\alpha\alpha\alpha_{134}$	0.36	0.44	0.46	**0.50**
$\alpha\alpha\alpha_{138}$	0.41	0.44	0.55	**0.58**
Average	0.37	0.42	0.47	**0.48**

Conclusions

Training on predicted structures improves inverse folding

Larger models (e.g., transformer) benefited more from predicted training data

ESM-IF1 learned more than inverse folding
- Stability
- Dynamics?

Future directions

Leverage predicted structures to train *de novo* generative models