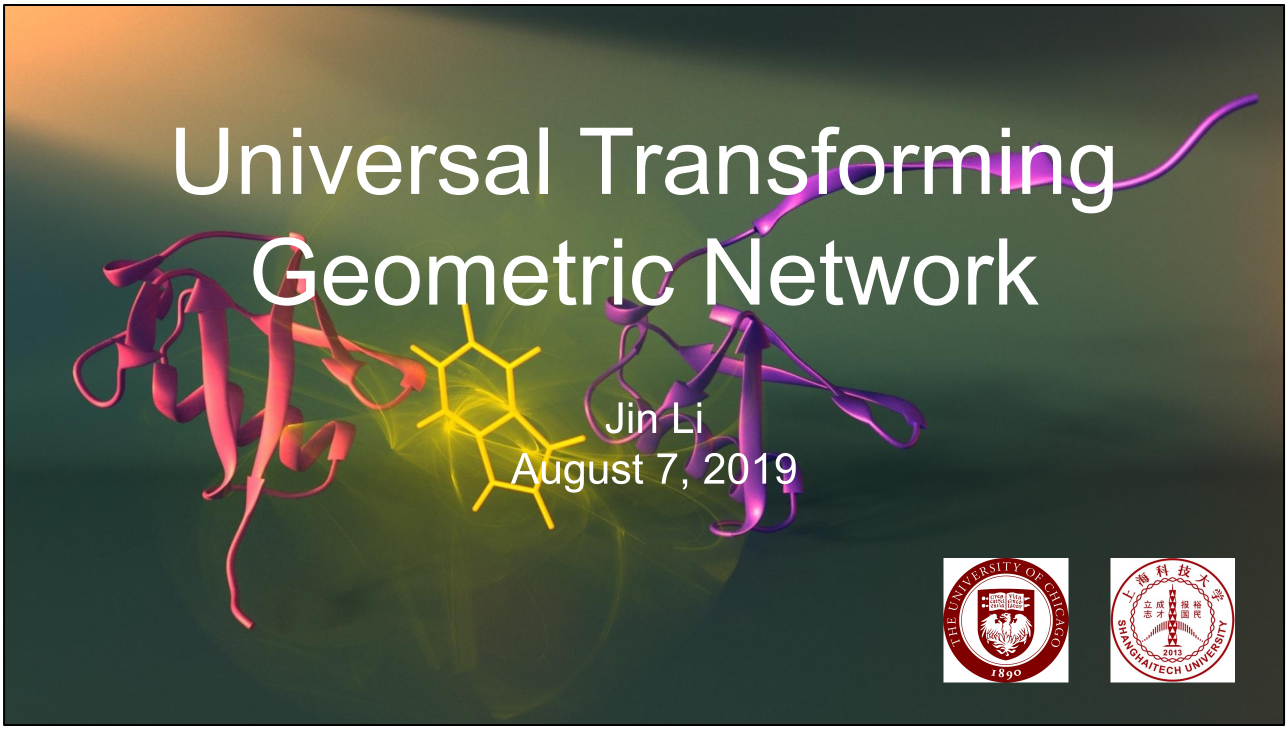


Universal Transforming Geometric Network

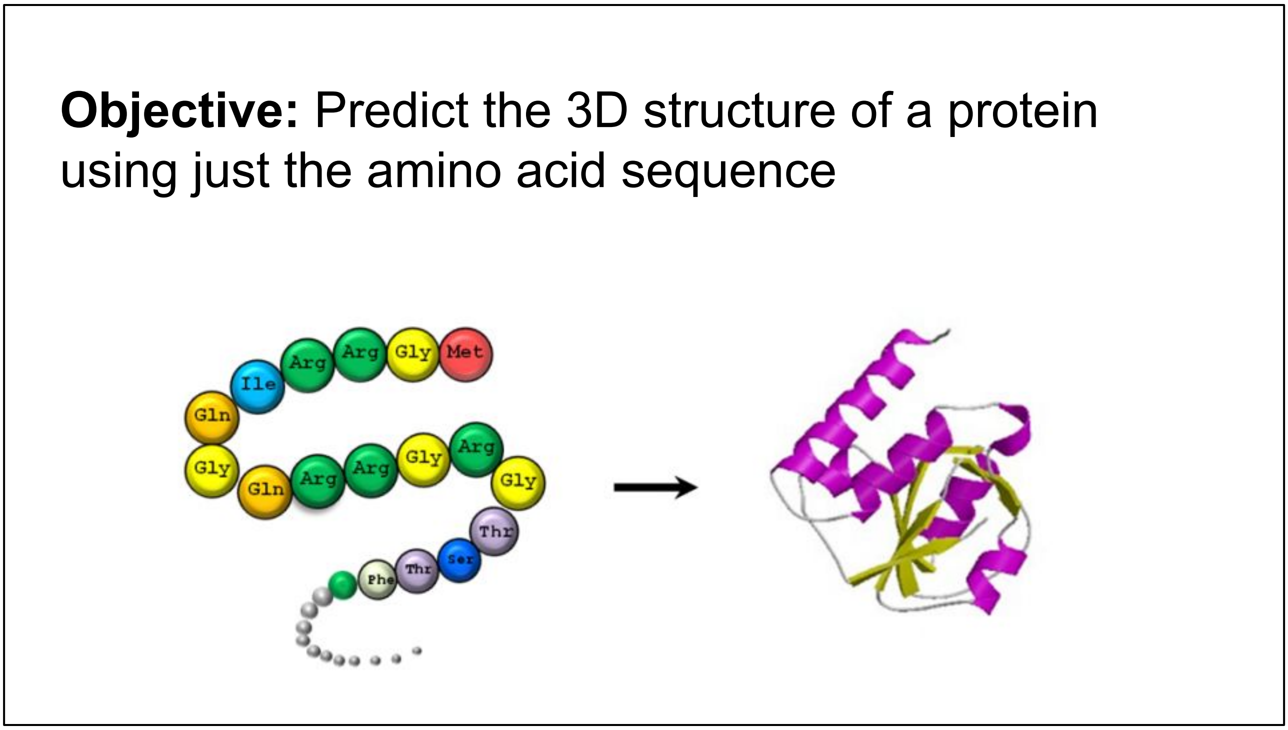
Jin Li
August 7, 2019

Audience: People who are familiar with deep learning and structural biology.
Presentation Time: ~20 minutes.
Acknowledgements: Thank you to Professor Zheng and Professor Zhao for providing guidance on this project. Work performed at ShanghaiTech University, School of Information Science and Technology.

Notes:

- I avoided the usage of mathematical notation. See the paper if interested in the math.
- Though the audience is expected to know machine learning, I will include some definitions for review.

[Paper]
[Code]



Objective: Predict the 3D structure of a protein using just the amino acid sequence

Basic Objective

In other words: Use deep learning to solve the protein folding problem.
Contribution: Proposal of a modified end-to-end differentiable neural network to predict 3D structure from a sequence of amino acids.
The original architecture is from the paper End-to-End Differentiable Learning of Protein Structure.

Diagram:

20 different amino acids in primary structure
Folds into a 3D structure.

How do we know there's any correlation between the input and output?

In 1950s, the Anfinsen Experiment showed that under the right conditions, all the information needed to form the three-dimensional structure of the polypeptide is stored in the specific sequence of amino acids.

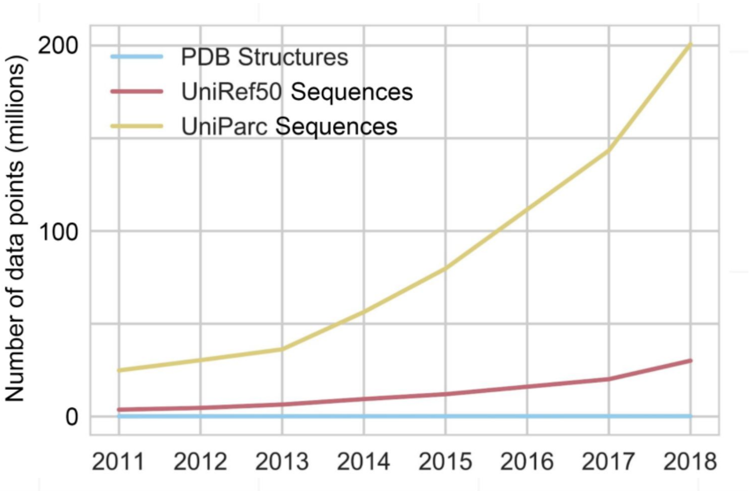
Definitions:

Neural network: stacked layers of computational blocks, called **artificial neurons**

End-to-end differentiable: whole process (from start to finish) can be optimized together via backpropagation

Protein folding problem: is it possible to predict the structure of a protein using just the amino acid sequence?

Number of available sequences exponentially outpaces the number of available structures



Motivation

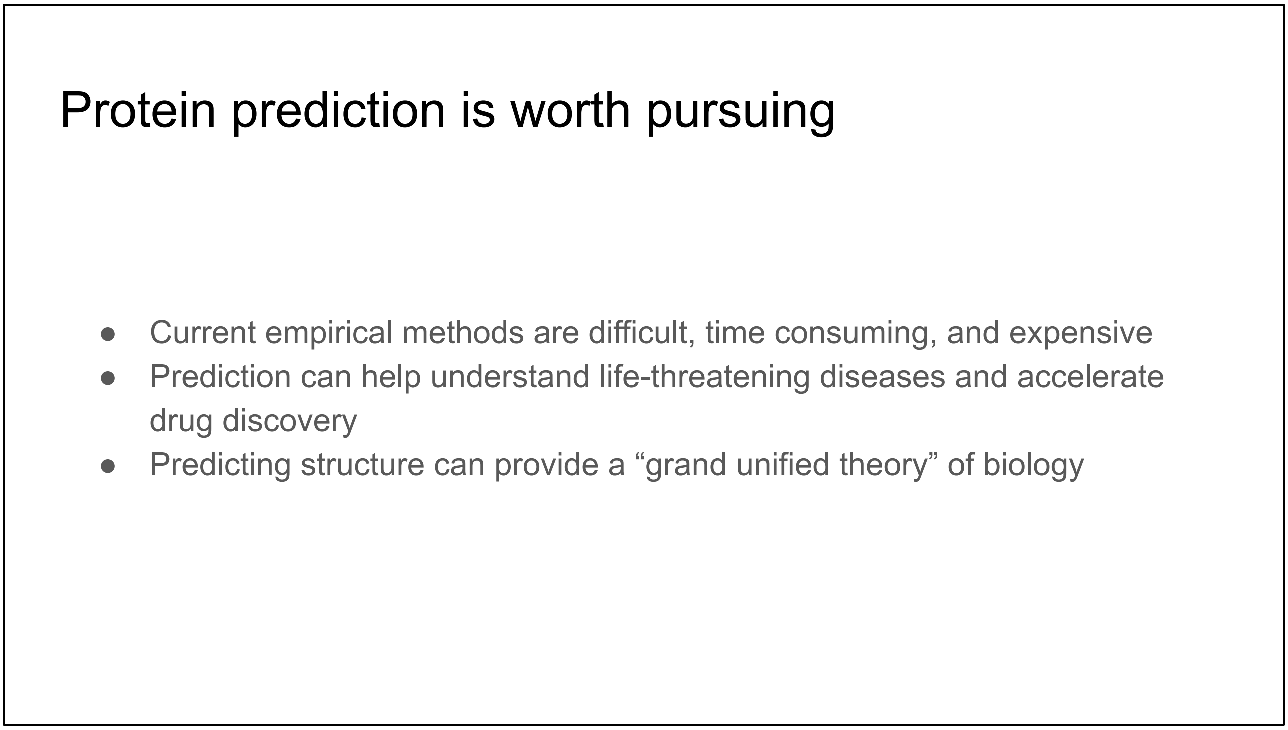
Why are we in need of a protein prediction algorithm?

We are far from documenting all possible protein structures.

200 million amino acid sequences, growing **exponentially**.

150,000 protein structure submissions, growing **linearly.**

6 million different proteins in the human proteome.



Protein prediction is worth pursuing

- Current empirical methods are difficult, time consuming, and expensive
- Prediction can help understand life-threatening diseases and accelerate drug discovery
- Predicting structure can provide a “grand unified theory” of biology

Motivation

Why is this worth doing?

Empirical methods:
X-ray crystallography
nuclear magnetic resonance
cryo-electron microscopy.

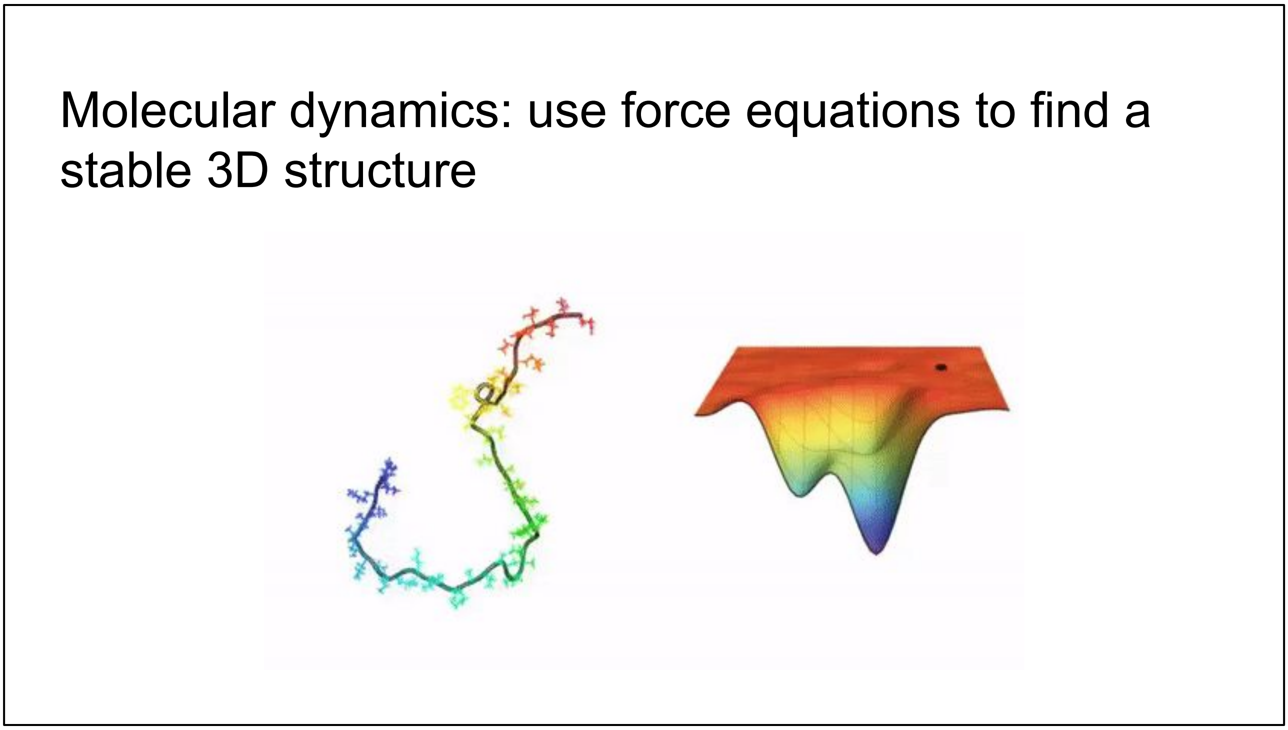
They have low success rate and the average cost is around $50,000 per protein.

Example of using prediction for drug discovery:
Goal: find a protein to bind to a specific enzyme.
Problem: Doing that experimentally is far too time consuming, costly.
Solution: predict structure of potential amino acid sequences and use computational methods to verify whether that sequence can bind to that enzyme.

Proteins are the key building blocks of life, they do everything everything from influencing an organism's growth to managing its internal state. If you can predict a protein’s structure, and by extension its function, you can argue that you have complete control over all fields of biology.

Definitions:

Grand unified theory: a formulation that combines all branches of the field together.



Molecular dynamics: use force equations to find a stable 3D structure

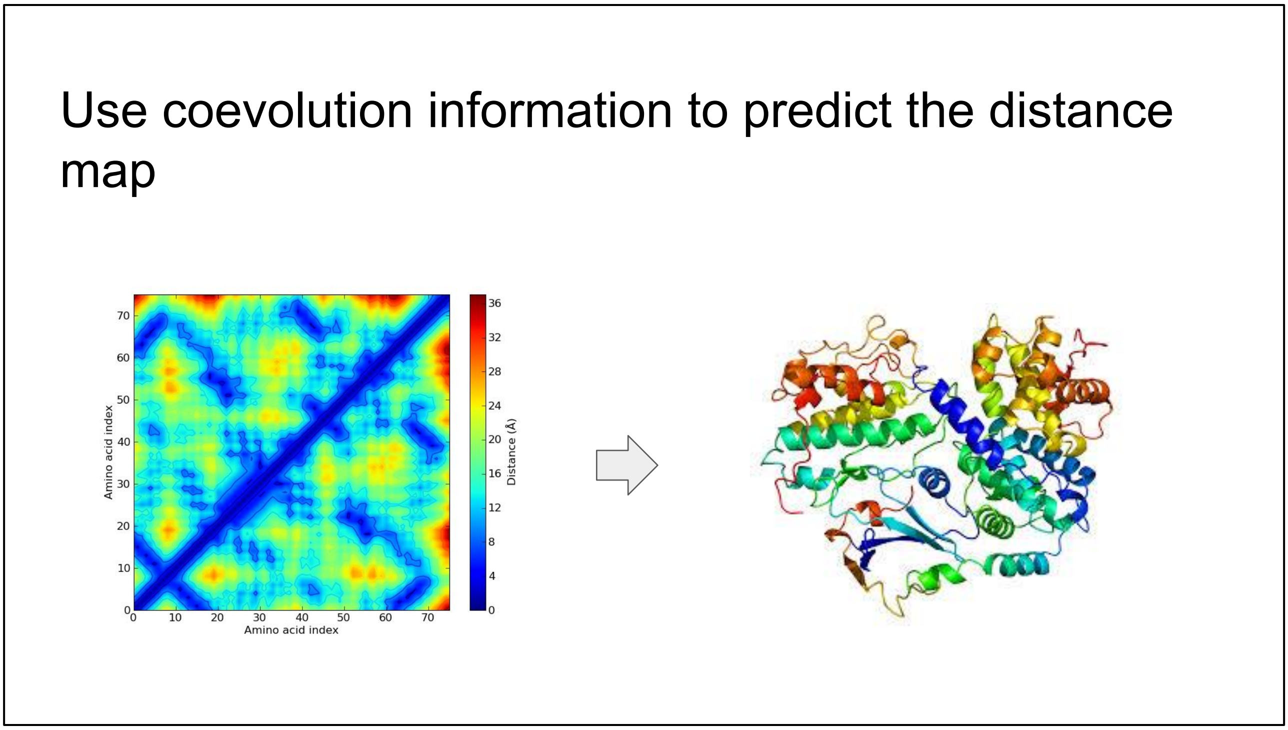
Previous Methodologies

Molecular Dynamics: use physics based principles to simulate the dynamical trajectory of the folding process to find a energetically stable 3D structure

Diagram: as the protein folds, the black dot moves around, trying to settle for the lowest energy state (the state that is most stable for the protein)

Cons:

Computationally expensive
Only useful for short protein chains



Use coevolution information to predict the distance map

Previous Methodologies

Contact map prediction:

1. Use multiple sequence alignments (MSAs) to search for signatures of co-evolving residues.
2. Use the coevolution information to predict the contact / distance map (like incorporating it into deep neural networks like CNNs)
3. Use the contact map as part of a pipeline for structure reconstruction

Pros:

Very popular and successful in CASP competition (works very well).

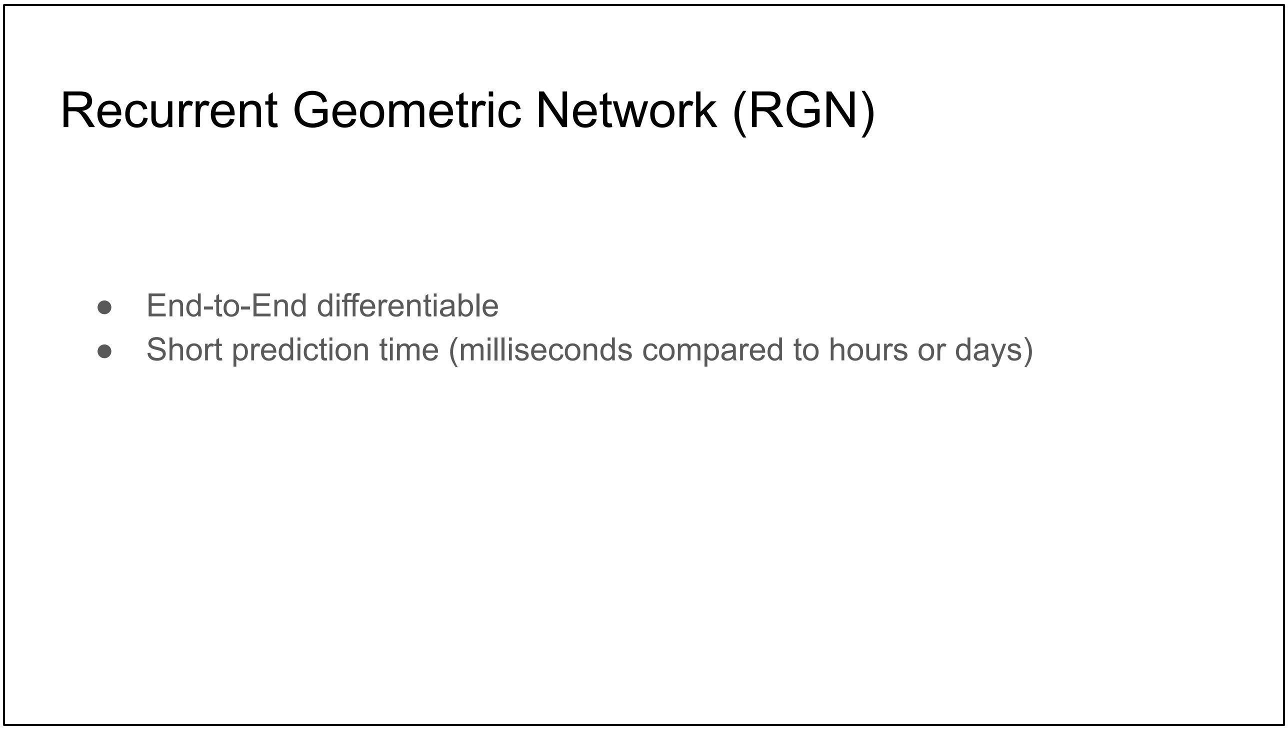
Cons:

Prediction in the order of hours to days.

Not end-to-end differentiable (contact map prediction is only part of the pipeline)

Diagram:

Left: distance map. The two axis indicate the position of the amino acid in the sequence, and color represents the distance between the two amino acids.



Recurrent Geometric Network (RGN)

- End-to-End differentiable
- Short prediction time (milliseconds compared to hours or days)

Recurrent Geometric Network

My model is an extension of this network.

Why RGN:

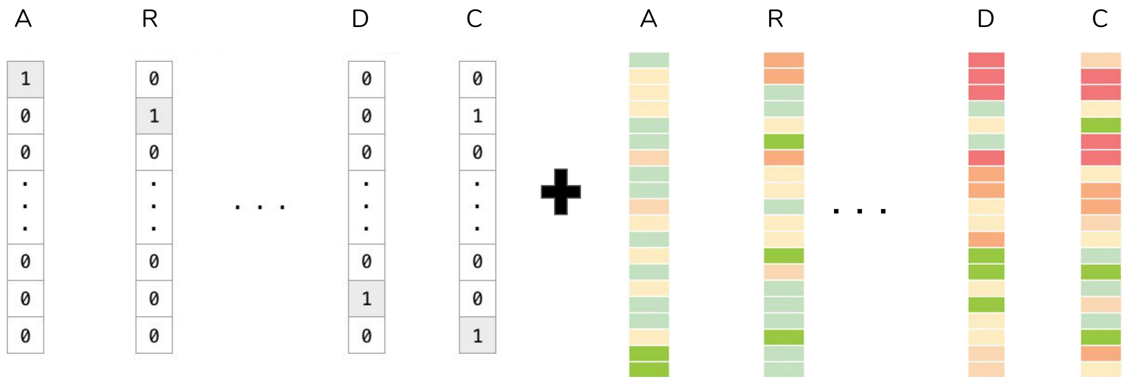
First end-to-end differentiable protein prediction network, meaning the model learns to optimize the entire process at once.

The model is one contingent piece, not multiple pieces stringed together. This decreases complexity.

Short prediction time is very useful for applications that involve trying out many different sequences.

Anecdote: In the early 2000s, the image recognition process was segmented into many different parts, then stringed together. Nowadays, image recognition is all in one end-to-end differentiable pipeline, and it works better.

RGN: Inputs



Amino acid sequence

Position-Specific Scoring Matrix

Recurrent Geometric Network

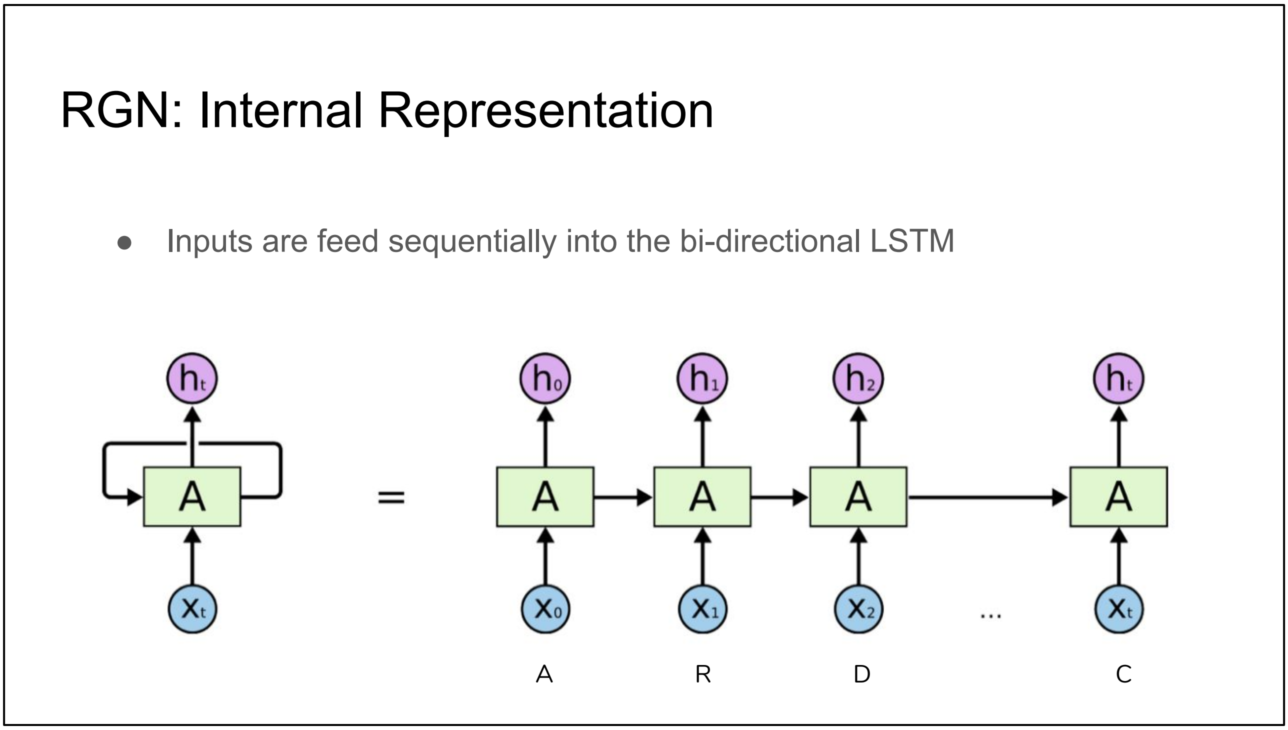
Diagram:

Amino acids are represented as 1 hot encodings.
Plus sign is concat, not add.
Amino acids are concated with the PSSMs.
Color represents scalar values from 0 to 1.

PSSMs are calculated from MSAs

Definitions:

One hot encoding: sparse vector representation of categorical data where the vector is one in a single position and 0 everywhere else.



RGN: Internal Representation

- Inputs are feed sequentially into the bi-directional LSTM

Recurrent Geometric Network

Diagram:

Left: diagram of RNN

Right: same RNN, but unraveled through the time steps.

At time step t, we feed in the next input (in this case, it is the vector representation of the amino acids) and we produce a state representation.

Definitions:

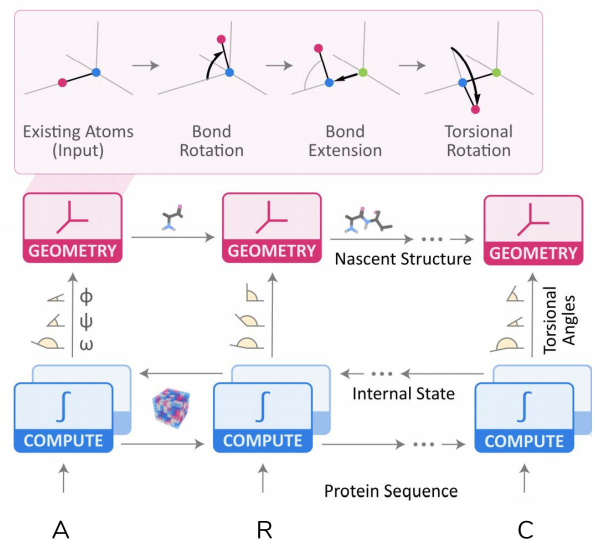
Internal representation: a vector representation of the information provided by the inputs

Recurrent neural network (RNN): a neural network that is fed input sequentially.

LSTM: a better version of RNN (better at controlling information flow)

Bi-directional: we also feed in inputs starting from the end to the beginning along with beginning to end

RGN: Internal states produce cartesian coordinates

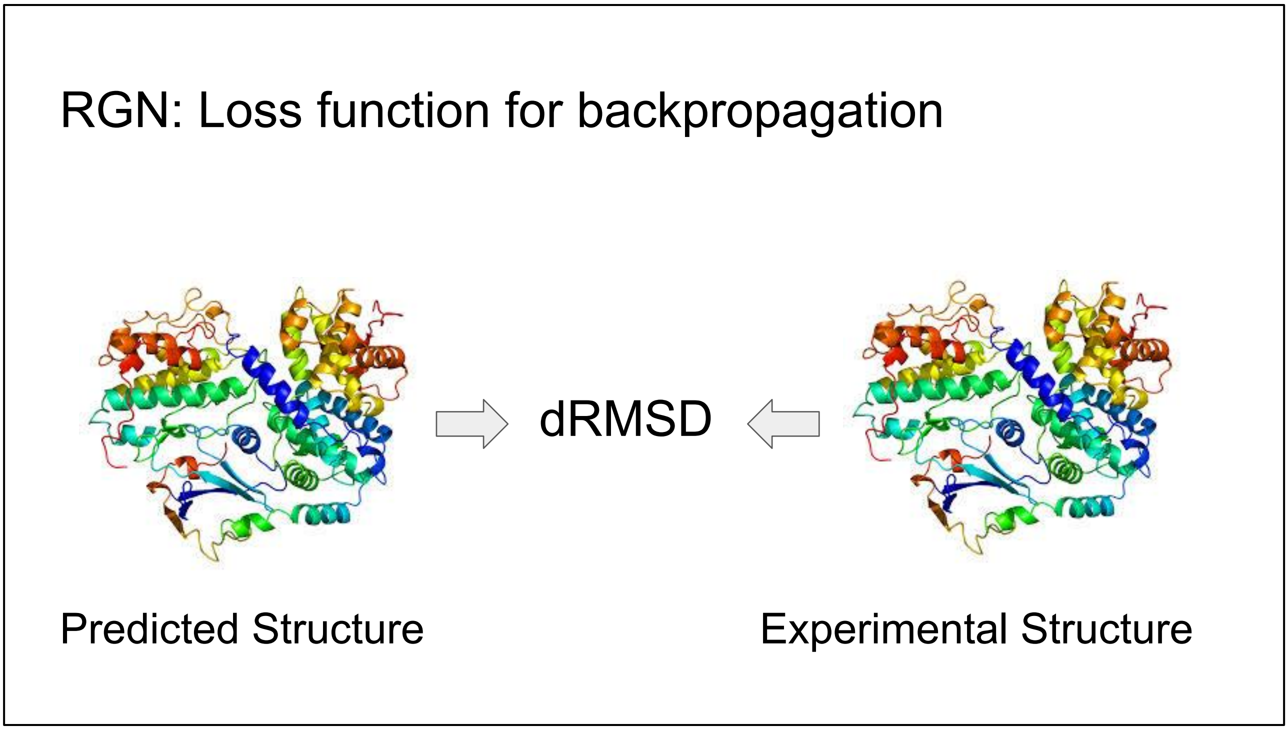


Recurrent Geometric Network

Diagram:

At each position, the state produces the omega, phi, psi torsional angles.

Those angles help construct the 3D cartesian coordinates sequentially.



RGN: Loss function for backpropagation

dRMSD

Predicted Structure

Experimental Structure

Recurrent Geometric Network

Diagram:

For training only.

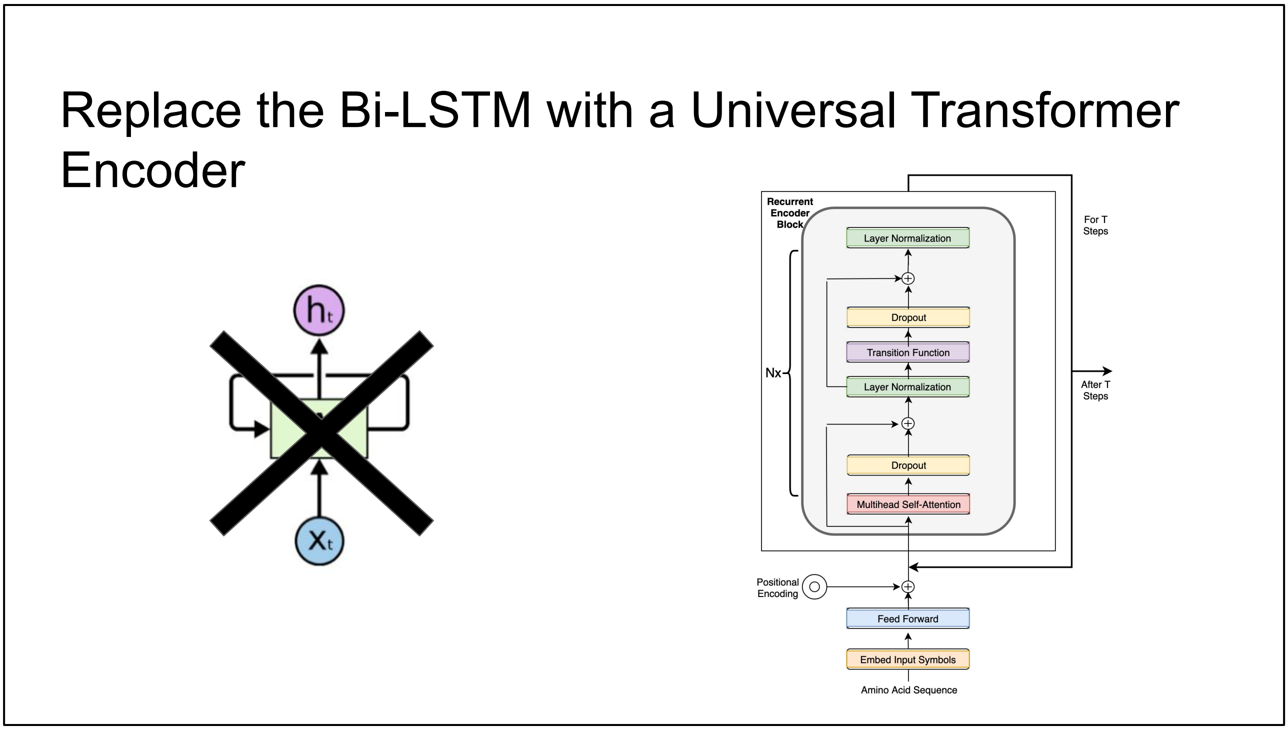
Calculate the dRMSD by creating the distance map of predicted and experimental structure. Then find the distance between the distance maps.

Perform backpropagation to update and optimize weights

Why dRMSD:

This loss function is translationally and rotationally invariant.

It is differentiable.



Replace the Bi-LSTM with a Universal Transformer Encoder

UT Encoder

Diagram:

Keep everything from RGN architecture, except the way the internal states are represented.

Will explain the model piece by piece.

Definitions:

Feed forward layer: a “regular” layer of neurons

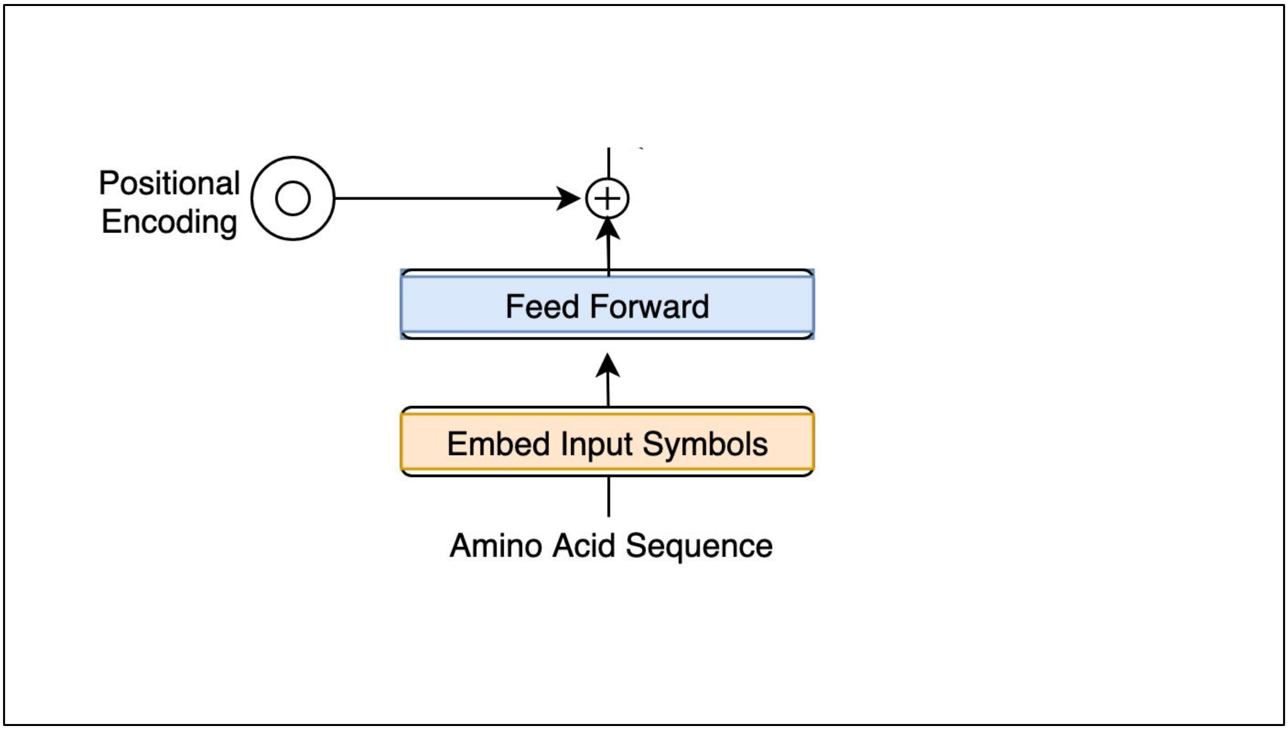
Residual connection: add the states of a certain layer to later layer

Positional encodings: constant sinusoidal matrix that represent positional information.

Dropout: For each iteration, random neurons in the model are not used.

Layer normalization: centers the weights to have a certain mean and standard deviation

Adaptive Computation Time (ACT) algorithm: determines which state to stop changing during recurrence.



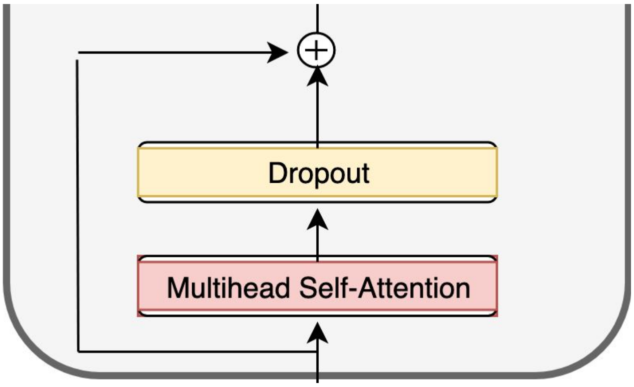
UTGN

Diagram:

Represent amino acid sequence just like in the RGN.

Feed into a feed forward layer.

Add positional encodings to include information of the position of the amino acid.



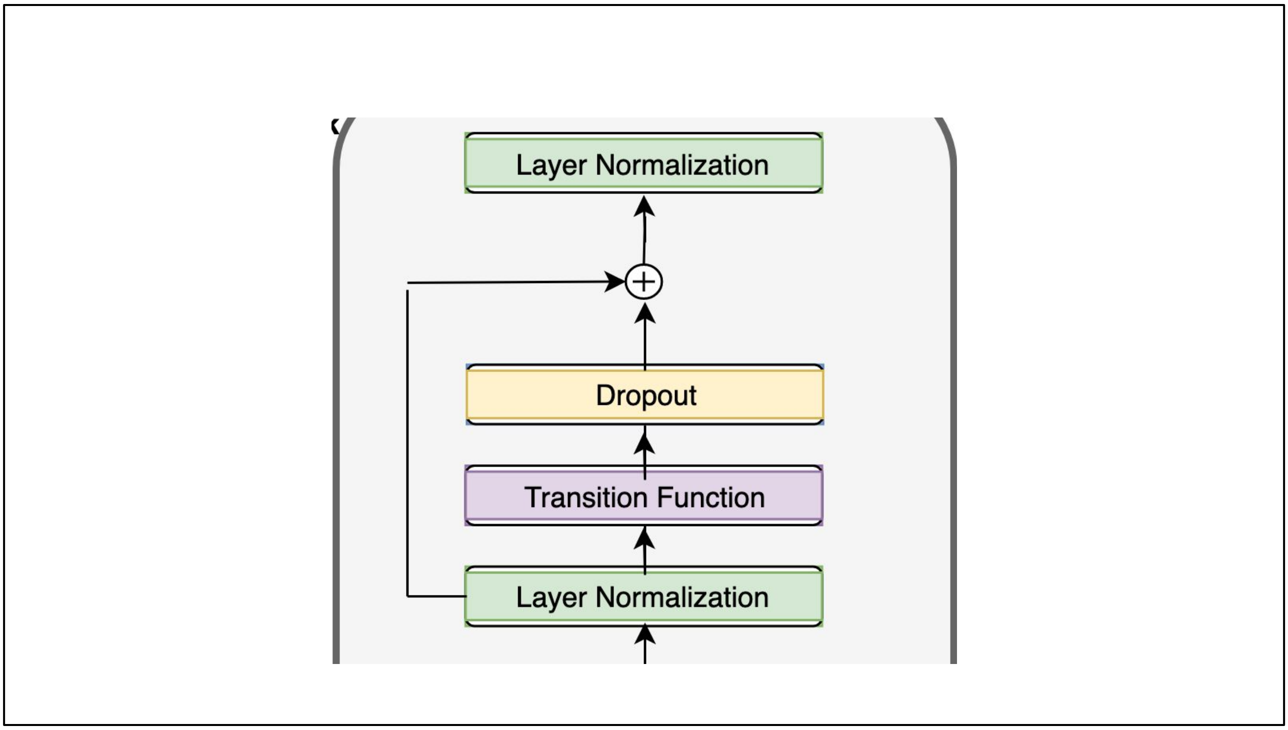
UTGN

Diagram:

Apply multihead self attention. At each position, the internal representation learns to pay different amounts of attention to previous inputs.

Next, apply a dropout to help prevent overfitting.

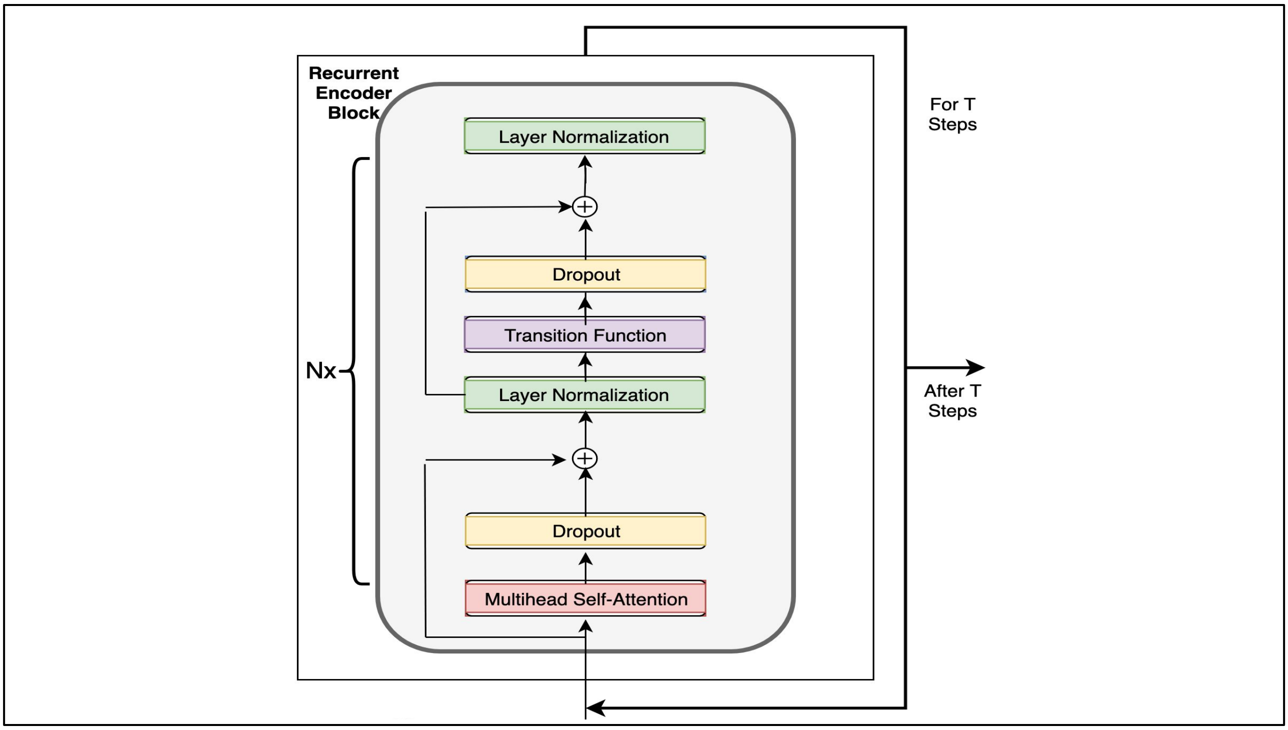
Residual connection to allow information flow to skip layers.



UTGN

Diagram:

Layer normalization to allow better gradient flow.
Transition function can either be feed forward or separable convolution.
Dropout.
Residual connection.
Layer normalization.



UTGN

Diagram:

We call the entire gray part a layer.

After passing through the gray area once, the internal representation will be passed through the same layer architecture for N times. Each layer has different weights.

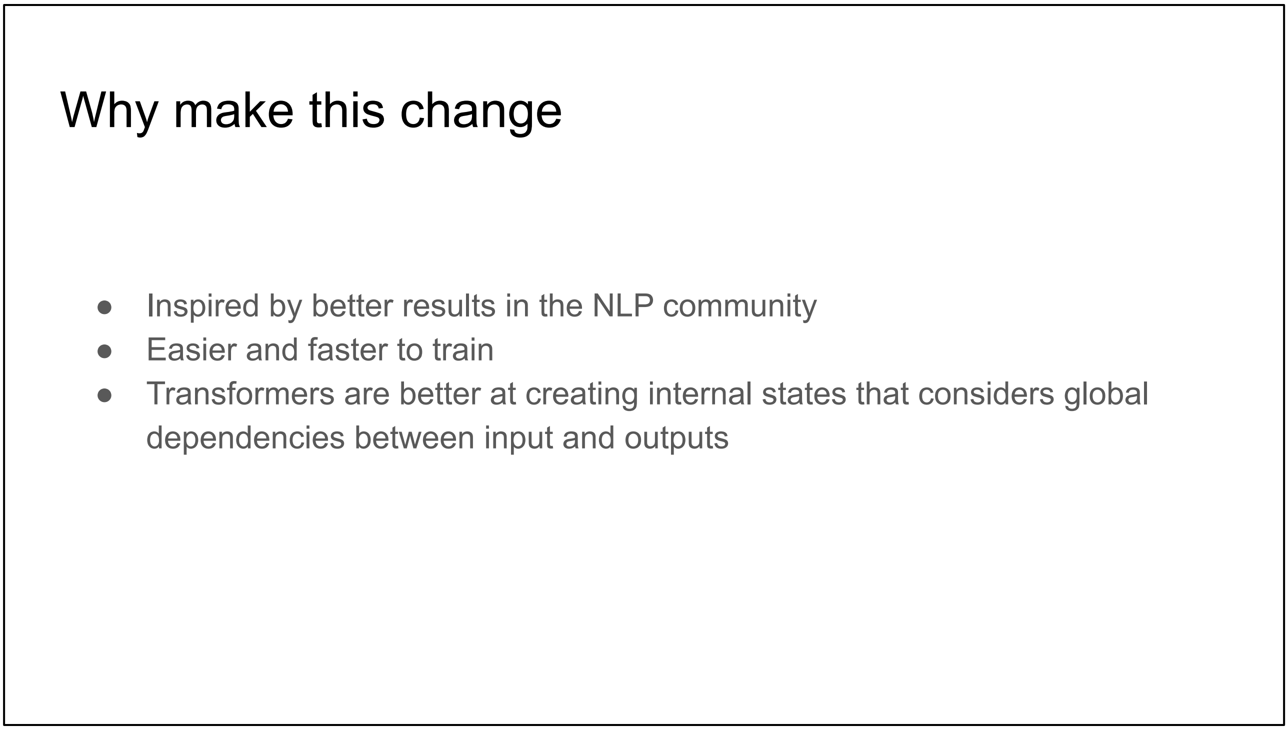
We call the layers together the encoder architecture.

Once that is done, the resulting states are passed back into the encoder architecture T times. This refines the states.

We also apply ACT mechanism to better focus on more ambiguously used amino acids.

Definitions:

ACT mechanism: an algorithm that halts changes in representations.



Why make this change

- Inspired by better results in the NLP community
- Easier and faster to train
- Transformers are better at creating internal states that considers global dependencies between input and outputs

UTGN

How is NLP related to protein prediction? Why do we want to use ideas from NLP?

Example: The way words in a sentence are related create the meaning in a sentence. How do we represent the relationship between the noun and the verb to represent action or intention?

In the same way, we want to use how the amino acids are related to one another to construct a structure. How do we represent the relations between amino acids to represent hydrogen bonds, ionic bonds, disulfide bridges, hydrophobic and hydrophilic interactions?

Both tasks involve storing information about relationships, which neural networks can do.

Faster to train because:

RNNs takes inputs sequentially, making it harder to parallelize.
Transformers take the entire sequence at once as an input.

Easier to train because:

Transformers are less susceptible to exploding / vanishing gradient problems found in training RNNs.

RNNs are unstable (different initialization produces very different results).

Global dependencies:

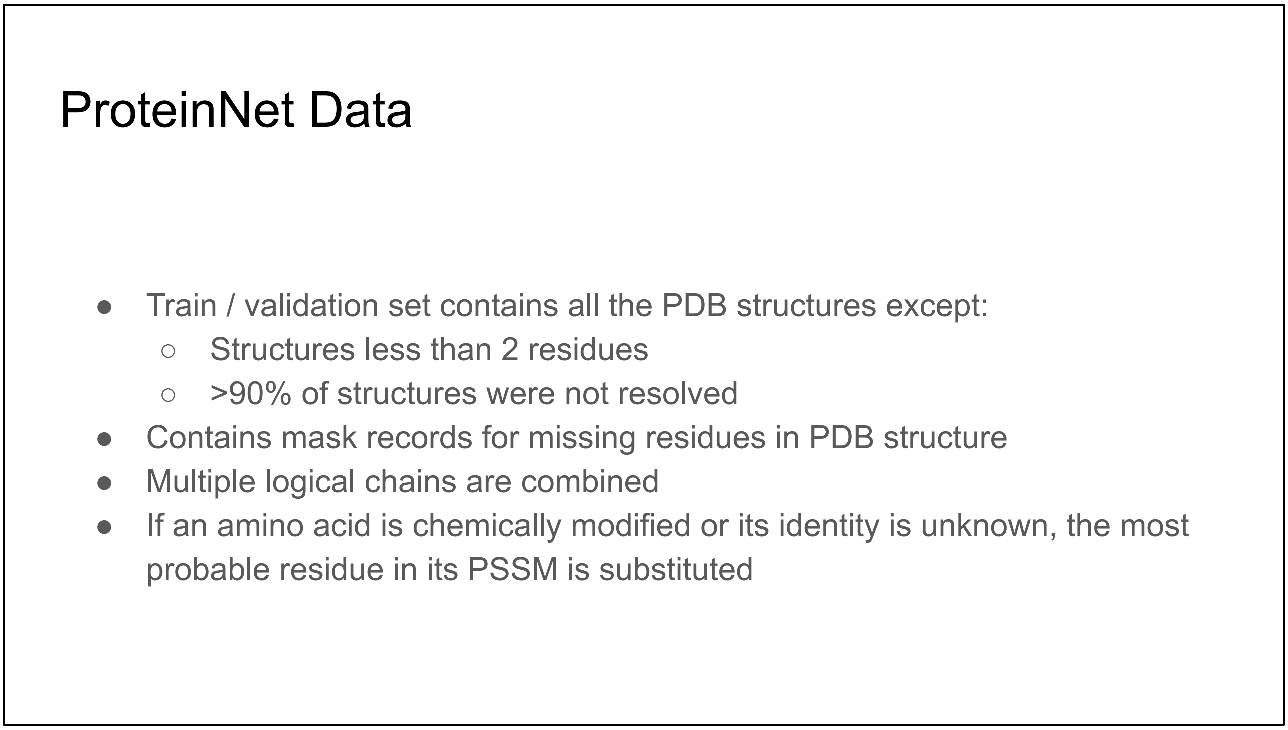
In RNNs, Information fades as time step increases.

Definitions:

NLP: natural language processing: study of language.

Exploding gradient problem: gradients become too large, making the model hard to stabilize to a good configuration.

Vanishing gradient problem: gradients become zero, making some neurons “die” and become useless.

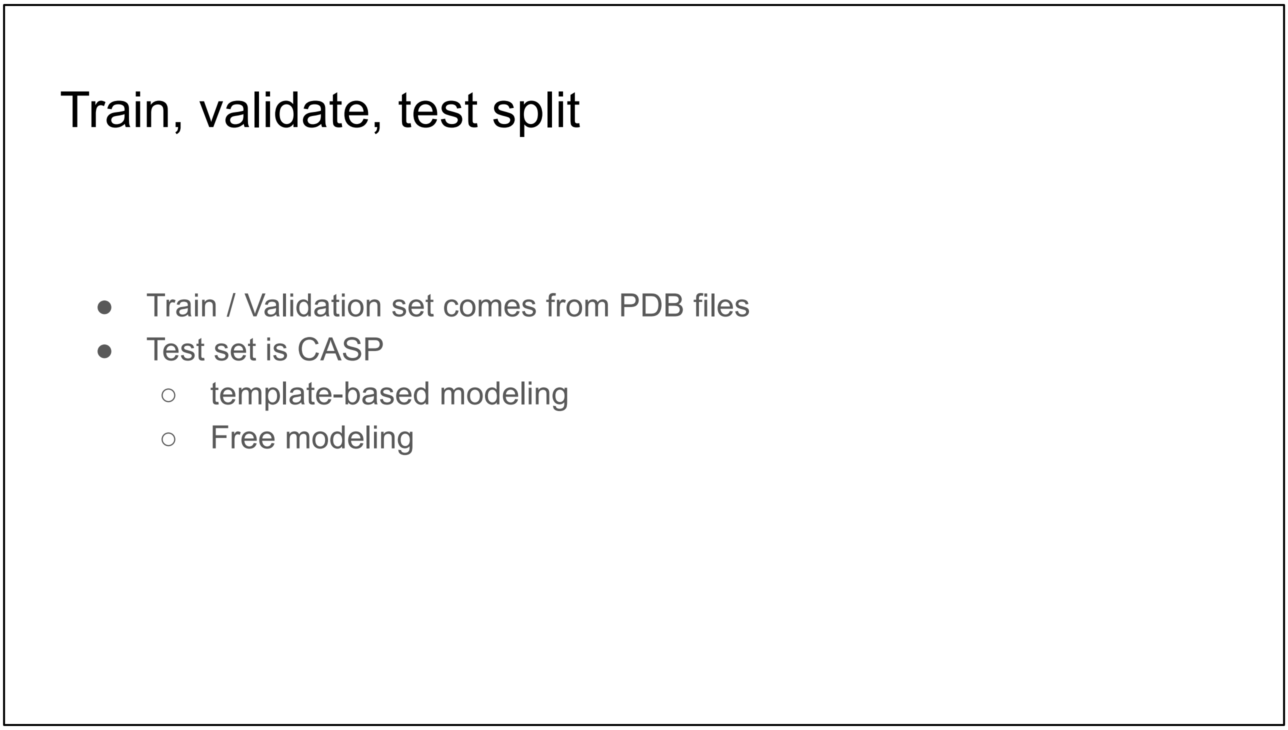


ProteinNet Data

- Train / validation set contains all the PDB structures except:
  - Structures less than 2 residues
  - >90% of structures were not resolved
- Contains mask records for missing residues in PDB structure
- Multiple logical chains are combined
- If an amino acid is chemically modified or its identity is unknown, the most probable residue in its PSSM is substituted

ProteinNet Dataset

Dataset is quite recent (2018)
Before this, lack a standardized dataset that everyone can use
Very liberal inclusion of data.



Train, validate, test split

- Train / Validation set comes from PDB files
- Test set is CASP
  - template-based modeling
  - Free modeling

ProteinNet Dataset

Data split:

Train / validation set split is not trivial like images (because can randomly shuffle images such that it would be iid).

Proteins are very similar to one another (has evolutionary relations).

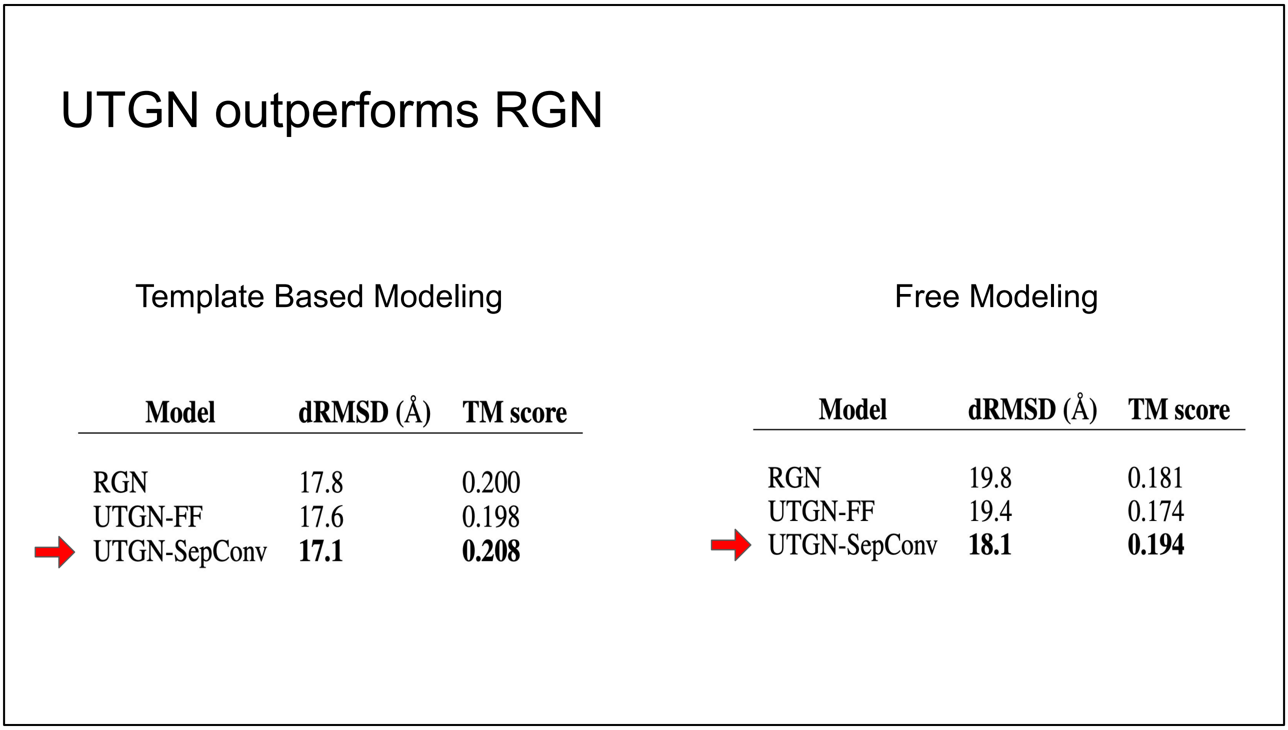
We want the validation set to be representative of CASP.

Definitions:

CASP: gold standard for evaluating protein prediction models. CASP comes once every two years.

template-based modeling (TBM): proteins with clear structural homology to PDB entries

Free modeling (FM): proteins with novel folds unseen in PDB



UTGN outperforms RGN

Template Based Modeling

Model	dRMSD (Å)	TM score
RGN	17.8	0.200
UTGN-FF	17.6	0.198
➡ UTGN-SepConv	**17.1**	**0.208**

Free Modeling

Model	dRMSD (Å)	TM score
RGN	19.8	0.181
UTGN-FF	19.4	0.174
➡ UTGN-SepConv	**18.1**	**0.194**

Results

Things to keep in mind:

We compared using smaller models for both RGN and UTGN because we did not have enough time for training. As a result, results are not that great.

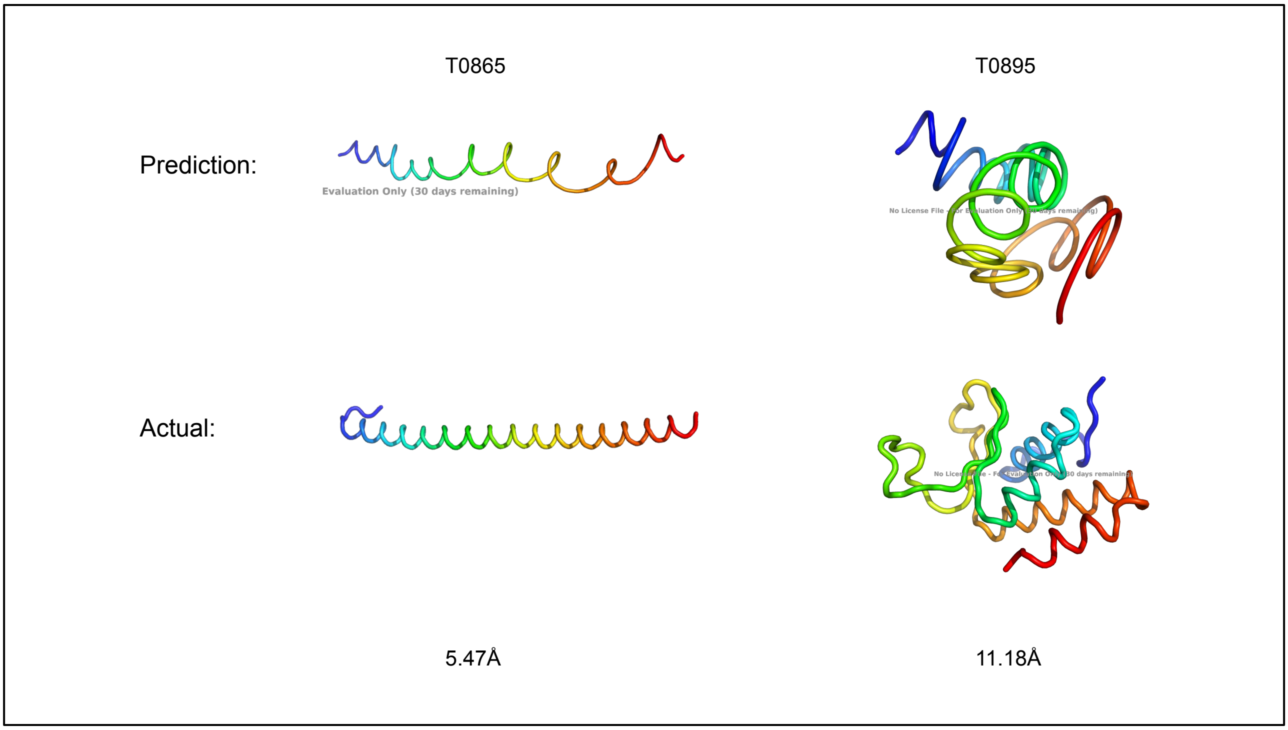
~2 million parameters for RGN, UTGN-FF, UTGN-SepConv

Definitions:

dRMSD: metric for protein prediction. Lower the better.

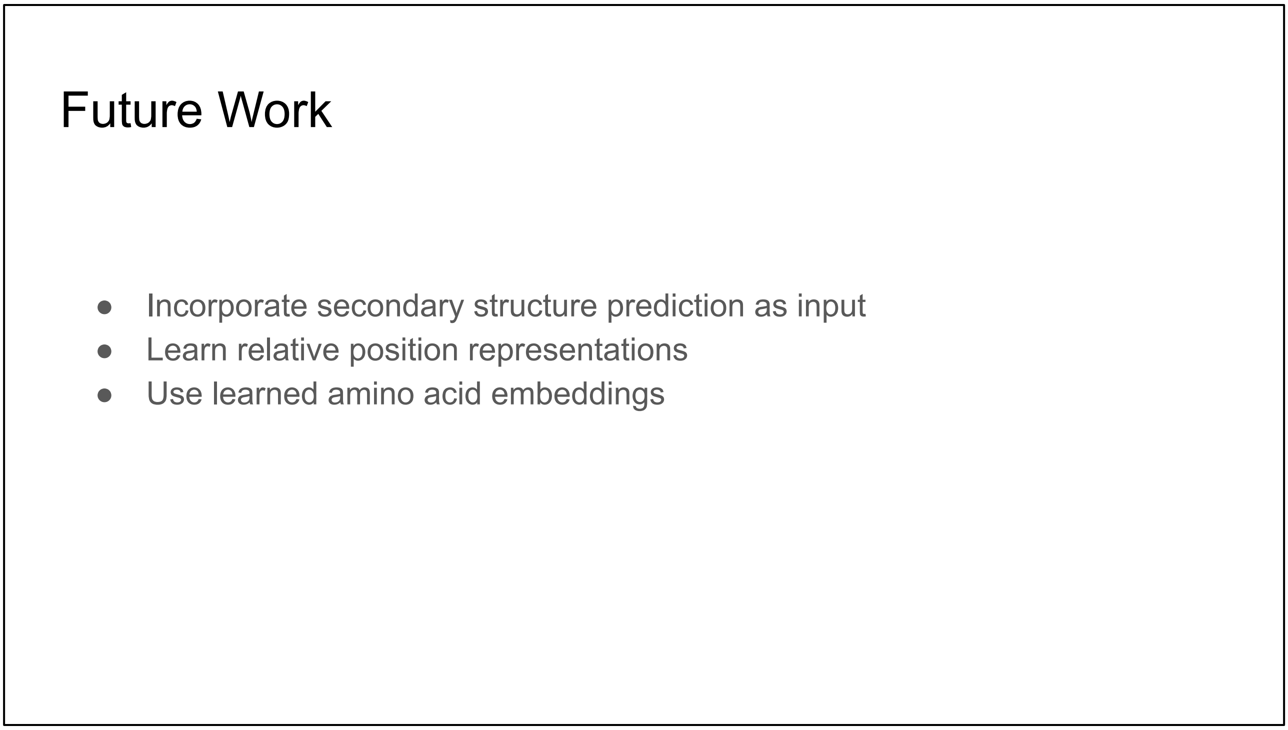
TM score: metric for protein prediction. Higher is better.

Takes values between 0 and 1.
0-0.17: random
0.5-1.00: in about the same fold



Results

Sample predictions.



Future Work

- Incorporate secondary structure prediction as input
- Learn relative position representations
- Use learned amino acid embeddings

Future Work