On the *Self-supervised Learning* of protein engineering

Boyuan Wang

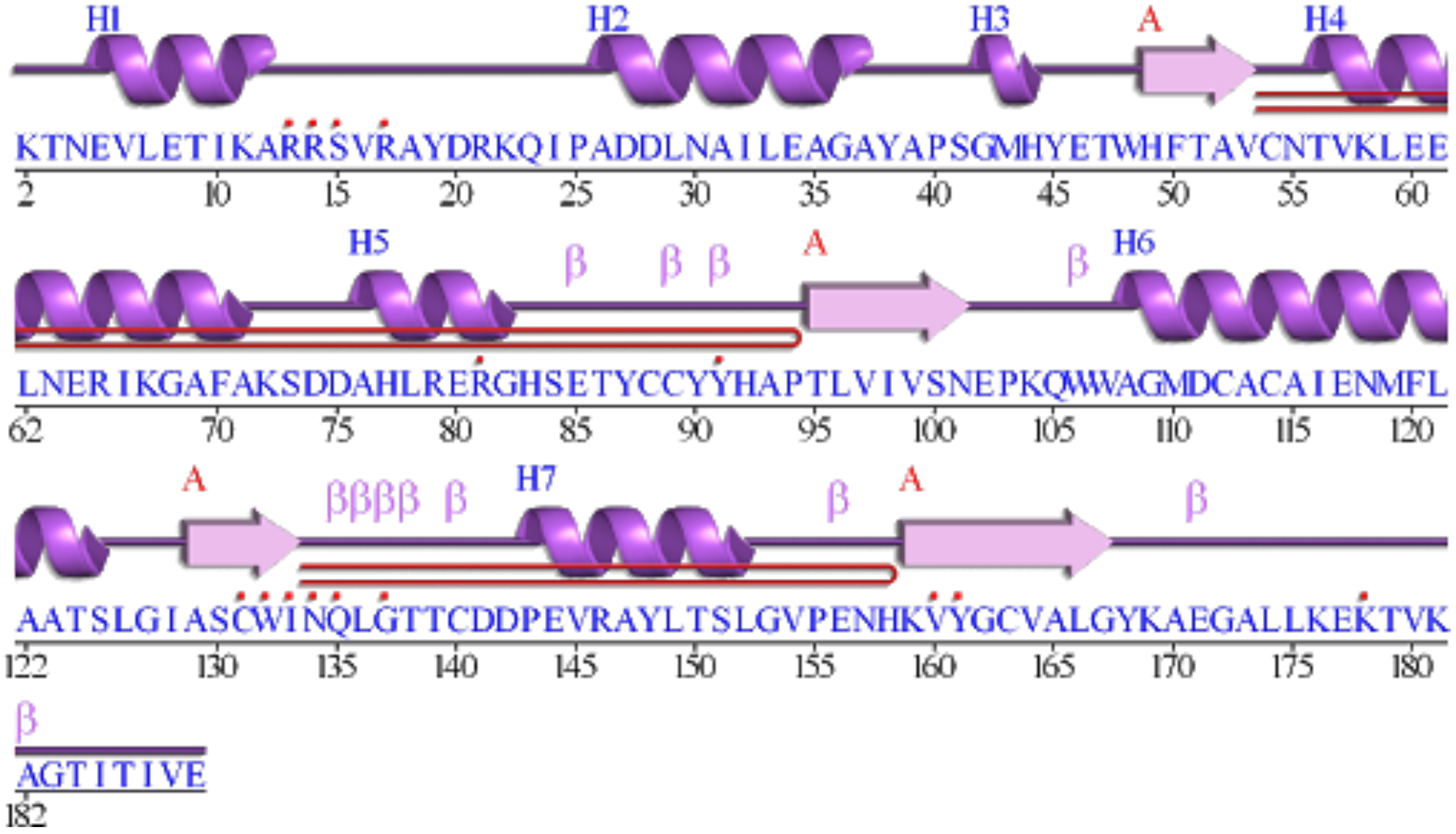
2020-05-01

What is protein engineering?

- **Protein engineering** is the process of developing useful or valuable proteins. It is a young discipline, with much research taking place into the understanding of protein folding and recognition for protein design principles. -*from Wikipedia*
- Common tasks in protein engineering:
    - Secondary structure prediction (1D)
    - Contact map prediction (2D)
    - Protein folding prediction (3D)

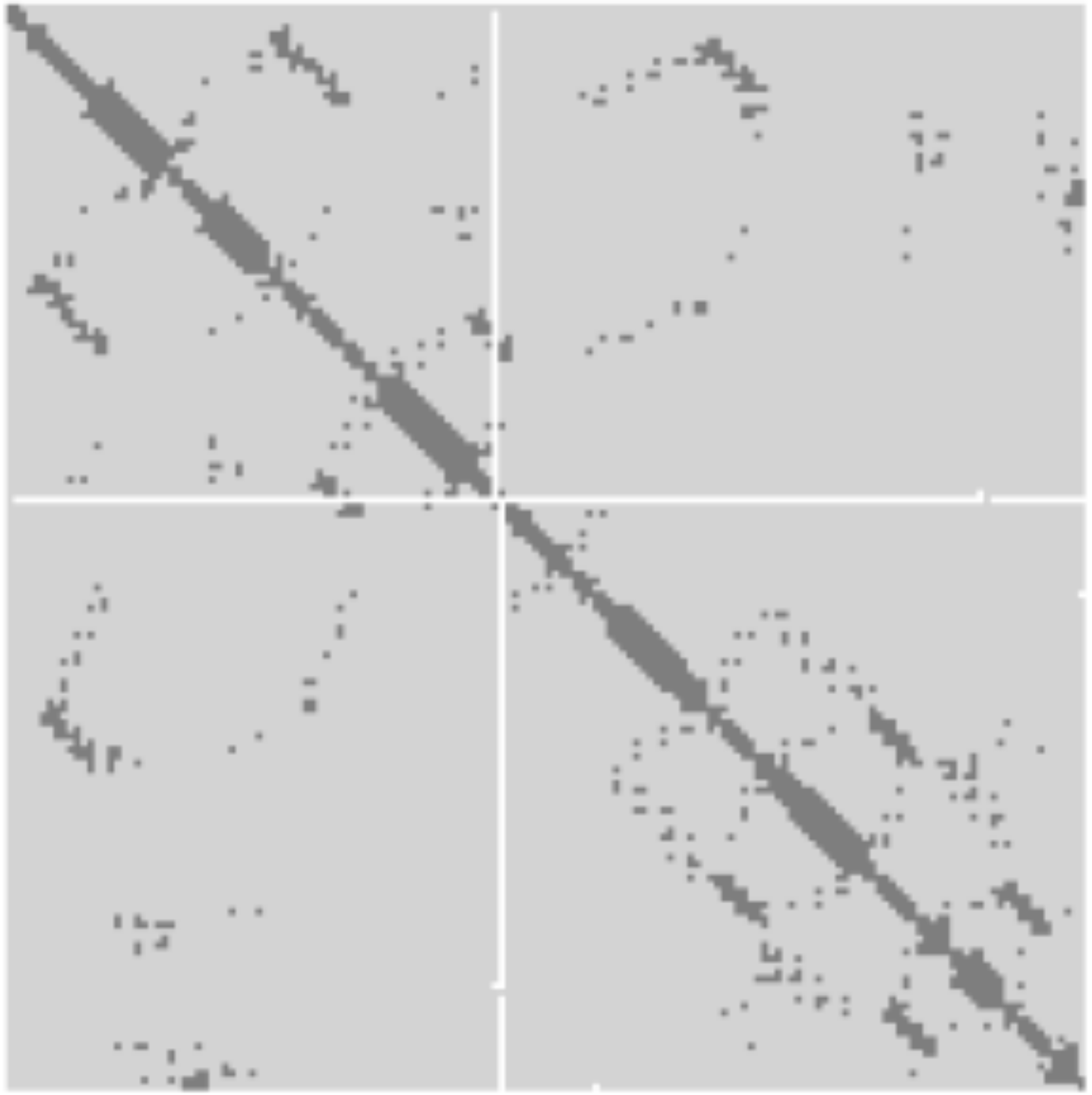
Secondary Structure Prediction

- Predict the position of alpha-helix (H) and beta-strand (E), coil region(C).


H1
H2
H3
A
H4
KTNEVLETIKARRSVRAYDRKQIPADDLNAILEAGAYAPSGMHYETWHFTAVCNTVKLEE
2 10 15 20 25 30 35 40 45 50 55 60
H5
β β β
A
β
H6
LNERIKGAFAKSDDAHLRERGHSETYCCYYHAPTLVIVSNEPKQWWAGMDCACAIENMFL
62 70 75 80 85 90 95 100 105 110 115 120
A
ββββ β
H7
β
A
β
AATSLGIASCWINQLGTTCDDPEVRAYLTSLGVPENHKVYGCVALGYKAEGALLKEKTVK
122 130 135 140 145 150 155 160 165 170 175 180
β
AGTITIVE
182

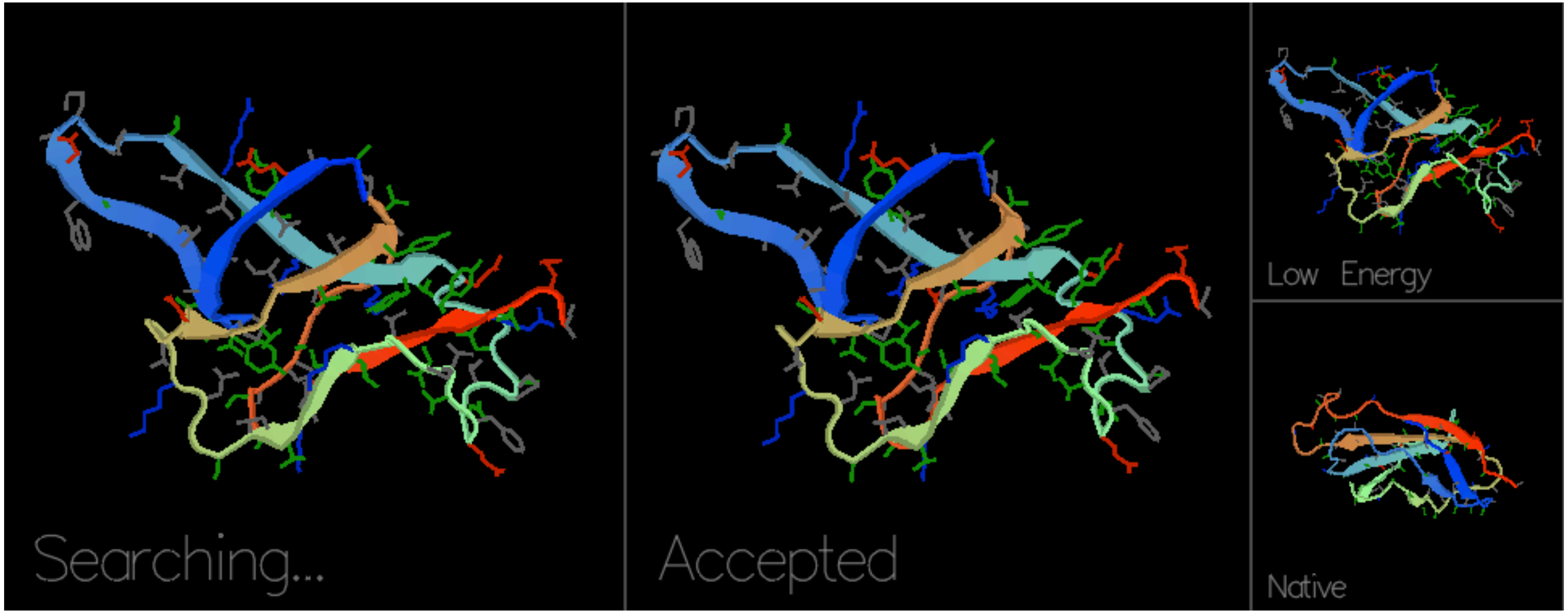
Contact map prediction

- Predict the contact information of amino acid residue



Protein folding prediction

- Predict the 3D geometric folding shape of proteins like Google Alpha-fold. (Hardest)



Why Do We Care Self-supervised Learning?

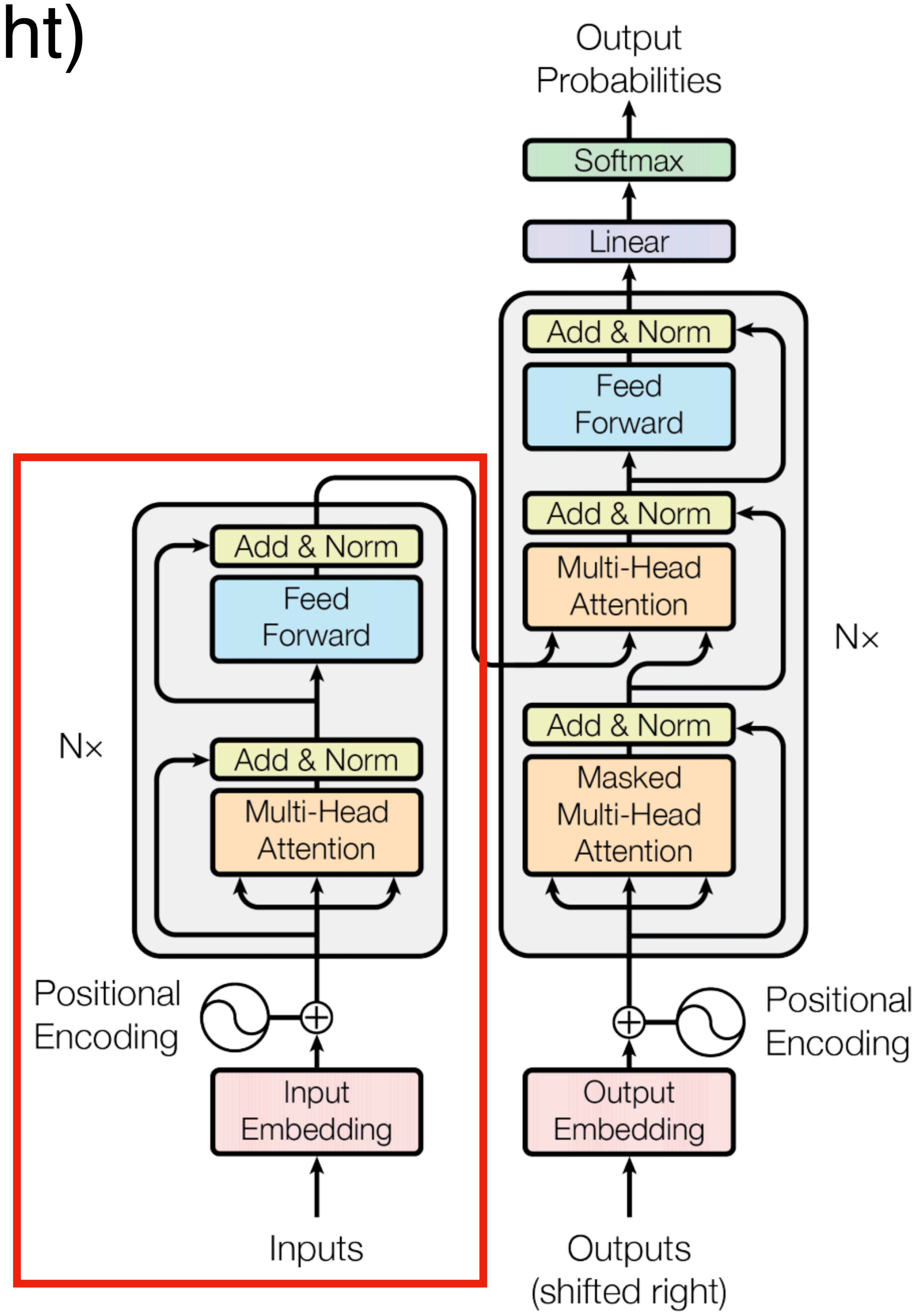
- Old methods involves too much human engineering work from selecting features to define functions for specific tasks.
- Recent use of deep supervised learning in protein engineering alleviates human laboring and brings exciting improvement in many tasks.
- However, data is scarce and obtaining supervised dataset is extremely costly in protein domain.
- Unlabelled protein data is abundant and contains the fundamental knowledge of proteins.
- Self-supervised learning is able to utilize the massive unlabelled data and extract knowledge from it.

Overview

- BERT: An Brief Introduction
    - Bidirectional Encoder Representations from Transformers, which is a pertained masked language model.
- Unified rational protein engineering with sequence-based deep representation learning (Nature Method 2019) Jumper
    - Rational protein engineering requires a holistic understanding of protein function. This paper proposed to use RNN based model to learn the holistic knowledge of protein sequences.
- Evaluating Protein Transfer Learning with TAPE (NeurIPS 2019)
    - This paper implements a more extensive comparison work between three different self-supervised models. It also provides 5 benchmark tasks and results.
- Generative models for graph-based protein design (NeurIPS 2019)
    - This paper introduces a conditional generative model for protein sequences given 3D structures based on graph representations.

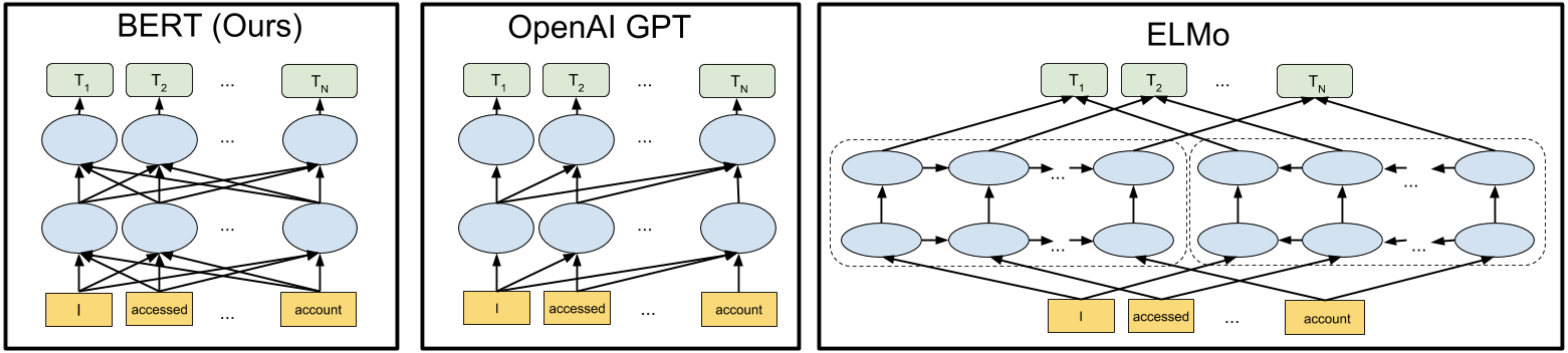
BERT - Architecture

- A stack of Transformer Encoder. (red box in the right)
- Bidirectional representation.



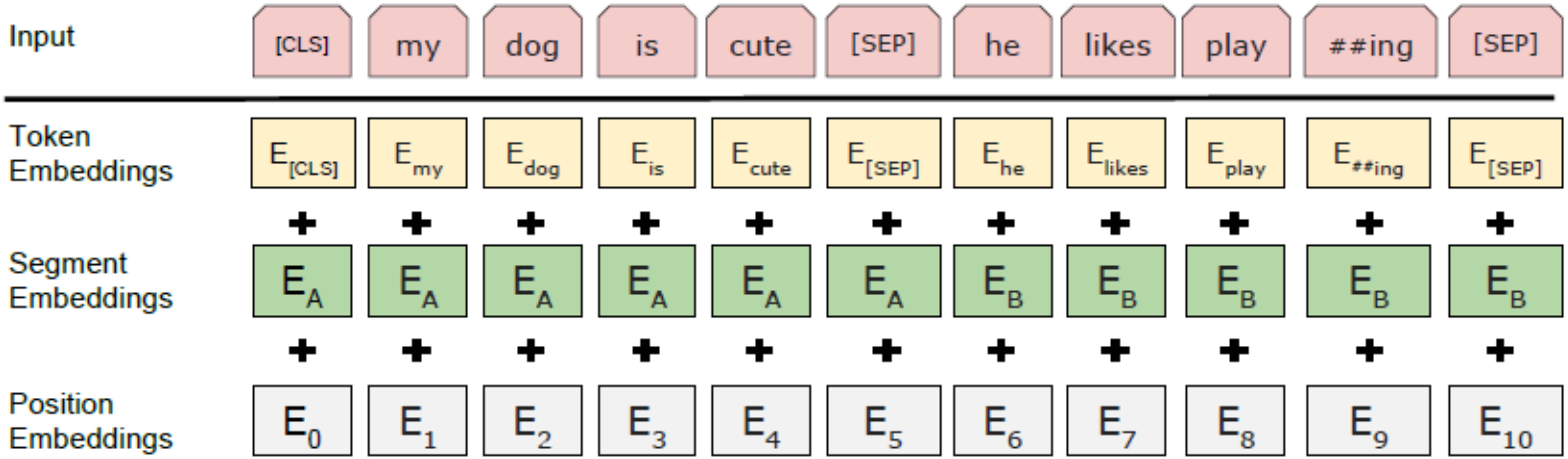
BERT - Architecture

- A stack of Transformer Encoder.
- Bidirectional representation.



BERT - Input Features

- Token embedding + position embedding +. Segment embedding (sentence pairs)



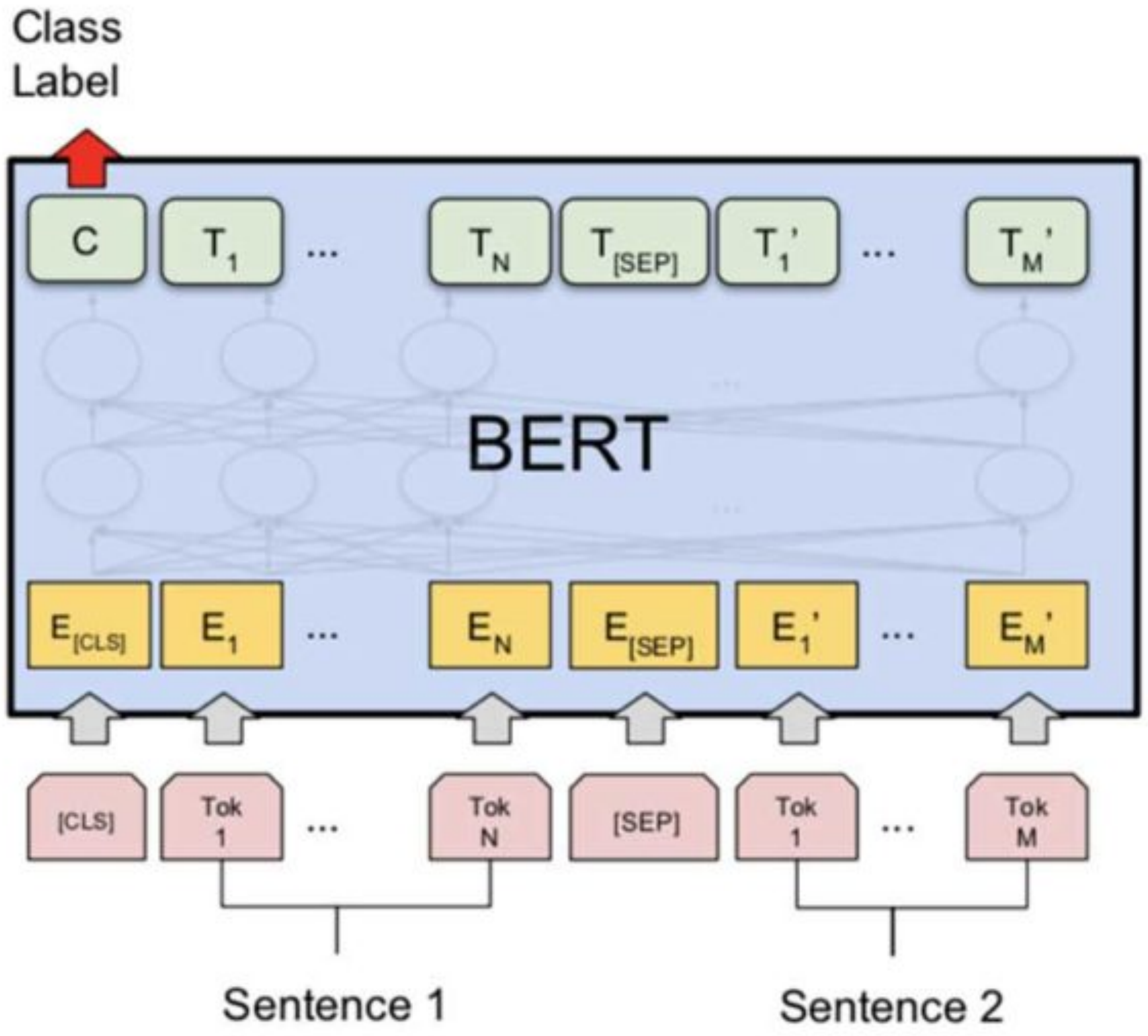
Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

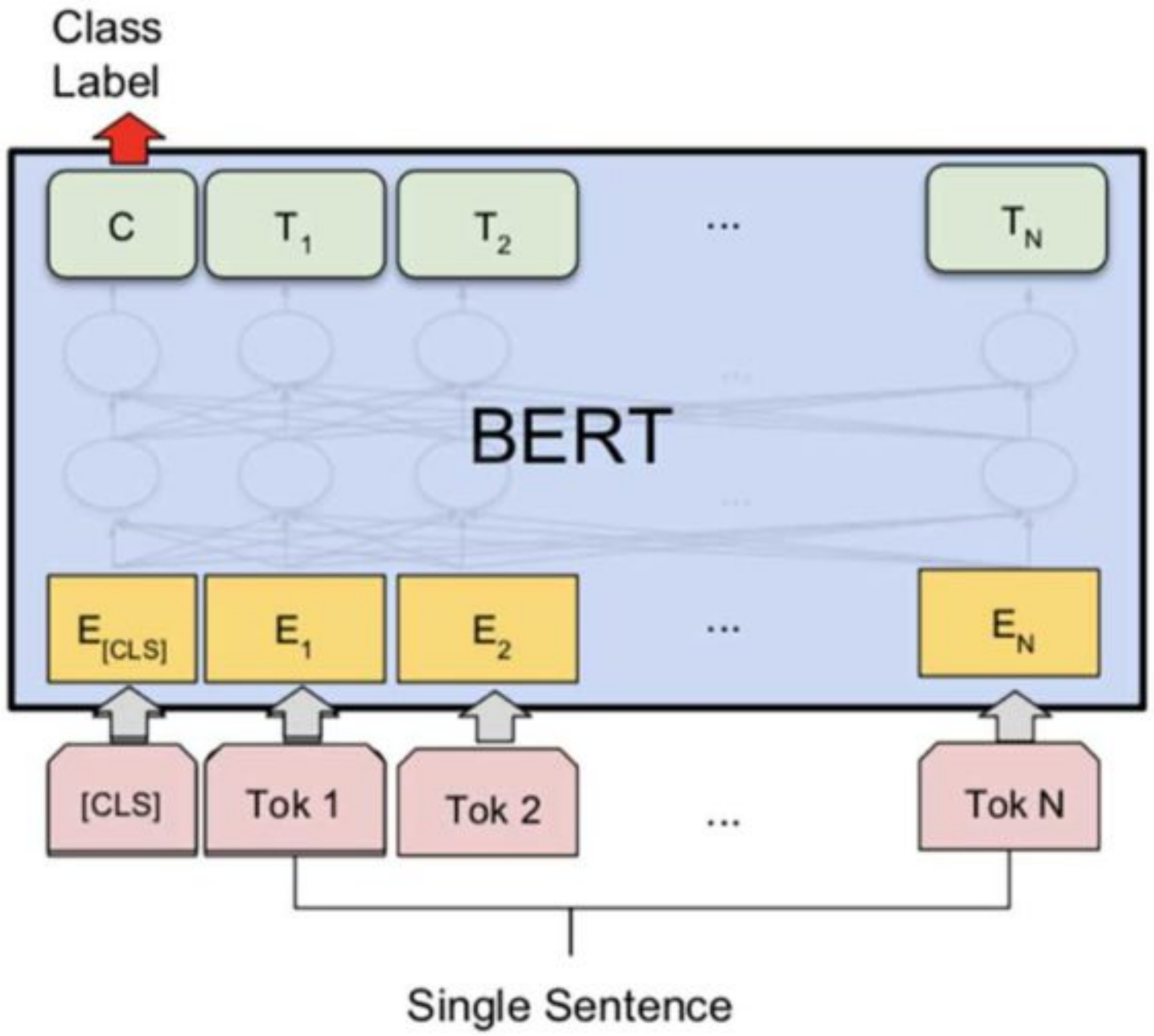
BERT - Pretrain Task

- Masked Language Model (MLM)
  - Mask 15% of tokens. Amount this 15%, 10% replaced, 10% unchanged.
  - 80%：my dog is hairy -> my dog is [mask]
  - 10%：my dog is hairy -> my dog is apple
  - 10%：my dog is hairy -> my dog is hairy
- Next Sentence Prediction (NSP)
  - Input sentence pairs (A, B), 50% of time B is the next sentence of A.
  - For question answering and natural language inference.

BERT - Fine-Tuning

- Fine-tuning on your specific tasks.
- [CLS] or token-level representation.





Overview

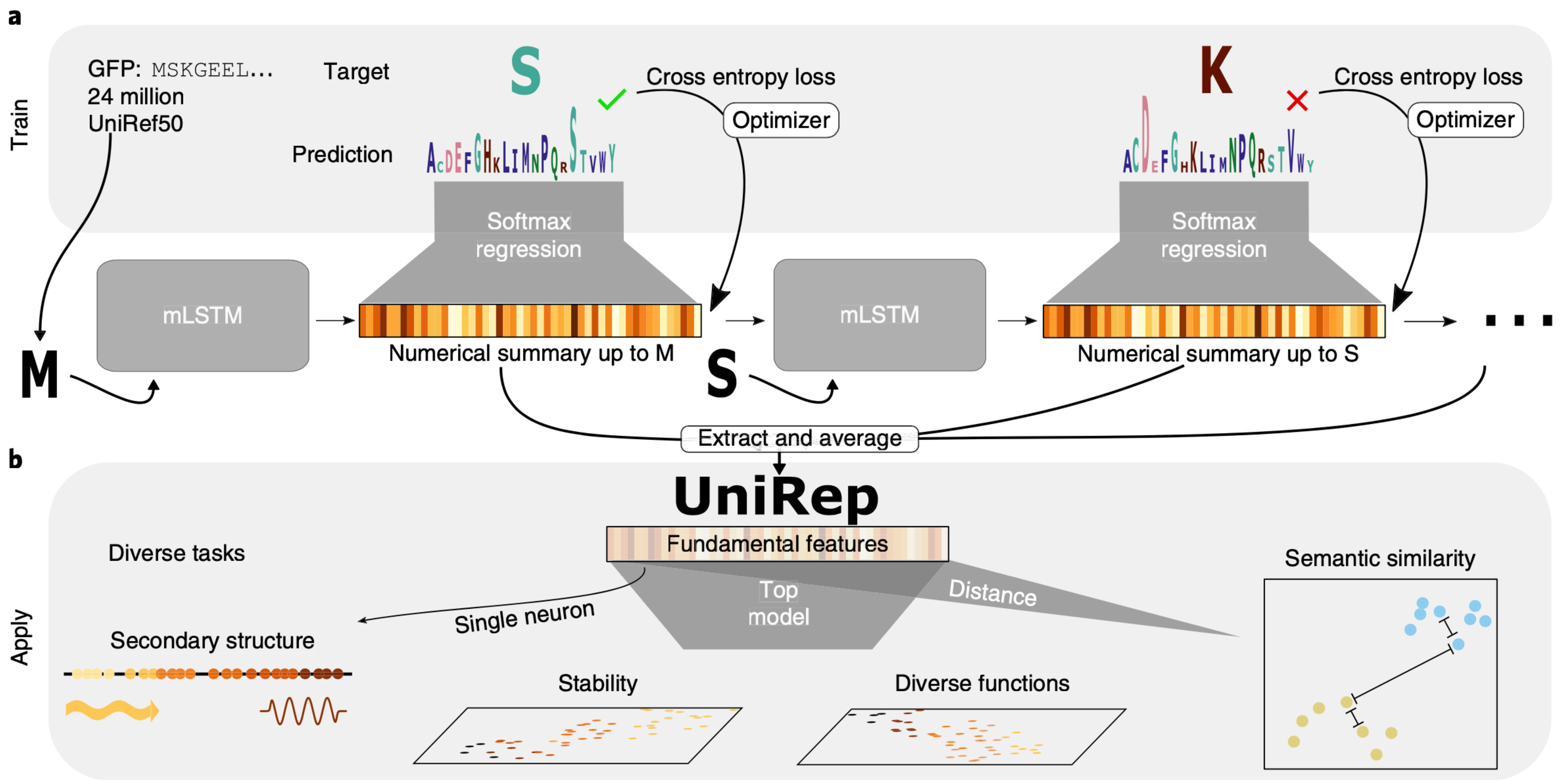
- Unified rational protein engineering with sequence-based deep representation learning (Nature Method 2019)
    - Rational protein engineering requires a holistic understanding of protein function. This paper proposed to use RNN based model to learn the holistic knowledge of protein sequences.
- Evaluating Protein Transfer Learning with TAPE (NeurIPS 2019)
    - This paper implements a more extensive comparison work between three different self-supervised models. It also provides 5 benchmark tasks and results.
- Generative models for graph-based protein design (NeurIPS 2019)
    - This paper introduces a conditional generative model for protein sequences given 3D structures based on graph representations.

Paper 1 - Motivation

- Protein sequence are sequential data. We want to learn the internal knowledge.
- Likewise, natural language process (NLP) also deal with sequential data. We can adopt the algorithms from NLP domain to protein domain.
- Self-supervision serves as pertaining scheme brings significant improvement to many NLP tasks because it learns some fundamental knowledge of language. It could also be the case for proteins.

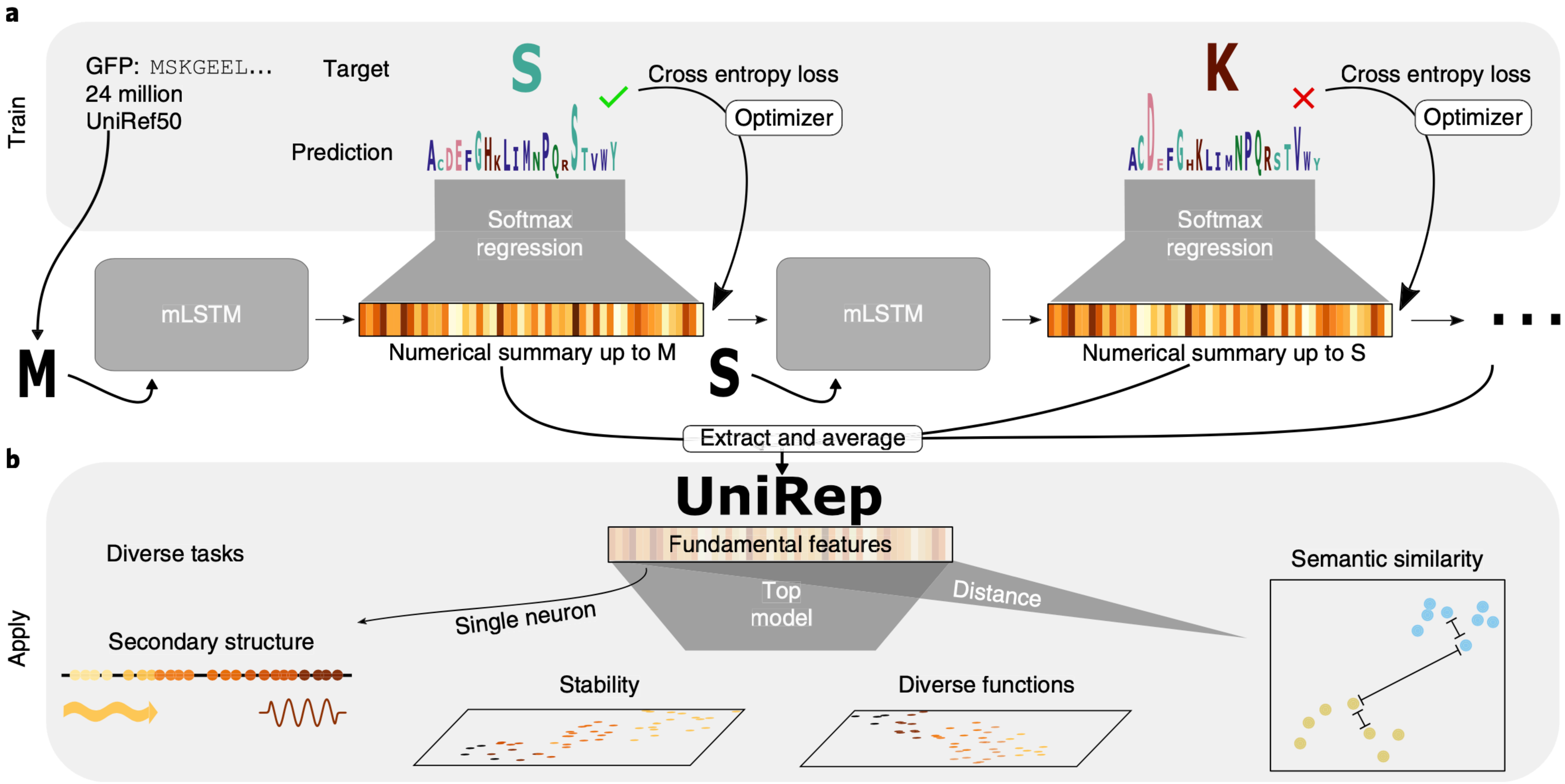
Paper I - Approach

- Self-supervision setup:
  - Architecture:
    - LSTM
    - Single-layer, 1900 hidden-size.
  - Loss:
    - Cross-entropy for all tokens.
  - Data:
    - UniRef: ~24 millions sequence.
    - Dictionary size: 20
  - Training Time:
    - ~770K steps, 1 epoch.



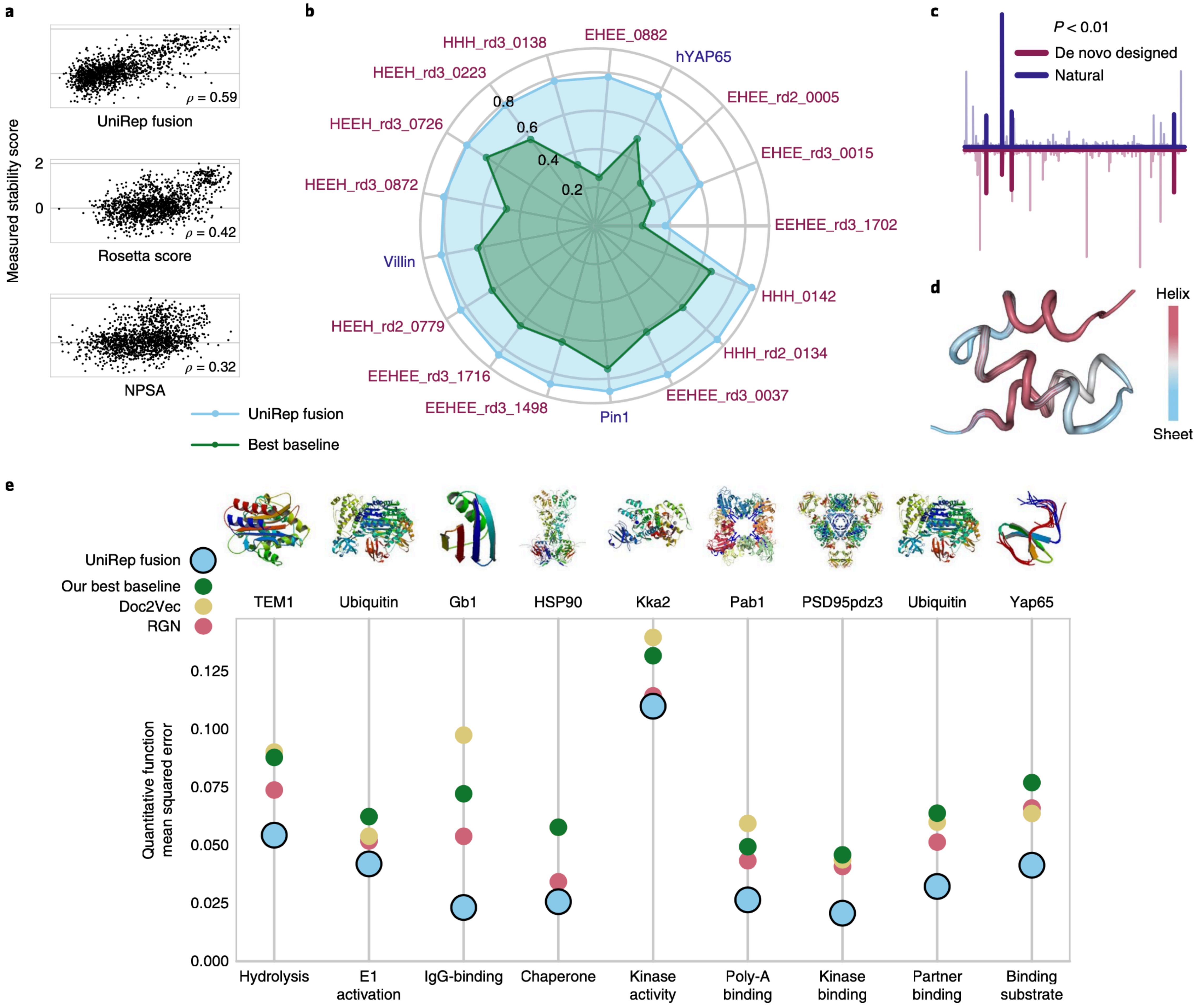
Paper I - Approach

- Training process:
  - Self-supervision: language modeling.
  - Downstream tasks: supervised learning.



Paper I - Experimental Results

- UniRep Feature:
  - averages all hidden states across time axis to make it more longterm dependent.
- Some results:



Paper I - Conclusion

- UniRep learns from raw data.
- It is unconstrained by a specific task, so features can be used in many tasks.
- It shows that protein informatics can potential go well directly from sequence to design.

Overview

- Unified rational protein engineering with sequence-based deep representation learning (Nature Method 2019)
  - Rational protein engineering requires a holistic understanding of protein function. This paper proposed to use RNN based model to learn the holistic knowledge of protein sequences.
- Evaluating Protein Transfer Learning with TAPE (NeurIPS 2019)
  - This paper implements a more extensive comparison work between three different self-supervised models. It also provides 5 benchmark tasks and results.
- Generative models for graph-based protein design (NeurIPS 2019)
  - This paper introduces a conditional generative model for protein sequences given 3D structures based on graph representations.

Paper II - Motivation

- The first attempt for systematically evaluating semi-supervised learning on protein sequences.
- TAPE includes a set of five biologically relevant supervised tasks that evaluate the performance of learned protein embeddings across diverse aspects of protein understanding.
- A framework for multi-tasks benchmark.

Paper II - Tasks

- **Task 1**: Secondary Structure (SS) Prediction
  - Impact: understanding the function of a protein. Important for high level of structure prediction.
- **Task 2**: Contact Prediction
  - Impact: global information. Important for final 3D structure prediction.
- **Task 3**: Remote Homology Detection
  - Type: multilabel classification
  - Impact: detection of emerging antibiotic resistant genes and discovery of new enzymes.
- **Task 4**: Fluorescence Landscape Prediction
  - Type: regression
  - Impact: efficient exploration of the landscape.
- **Task 5**: Stability Landscape Prediction
  - Type: regression
  - Impact: important to ensure that drugs are delivered before they are degraded.

Paper II - Datasets

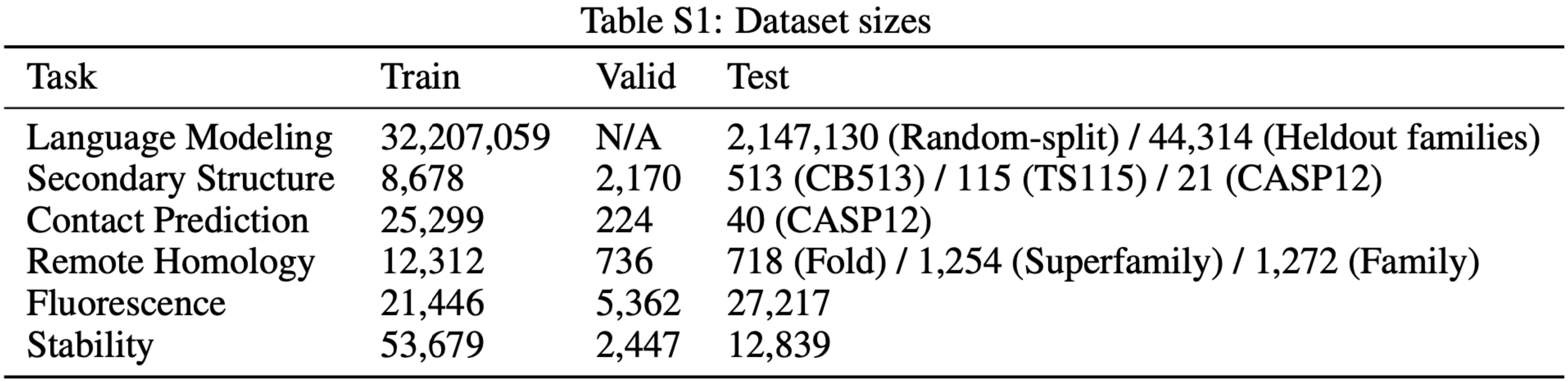


Table S1: Dataset sizes

Task	Train	Valid	Test
Language Modeling	32,207,059	N/A	2,147,130 (Random-split) / 44,314 (Heldout families)
Secondary Structure	8,678	2,170	513 (CB513) / 115 (TS115) / 21 (CASP12)
Contact Prediction	25,299	224	40 (CASP12)
Remote Homology	12,312	736	718 (Fold) / 1,254 (Superfamily) / 1,272 (Family)
Fluorescence	21,446	5,362	27,217
Stability	53,679	2,447	12,839

Paper II - Models

- Self-supervised Learning Setup:
  - LSTM (RNN)
    - forward 3-layer LSTM+ backward 3-layer LSTM, 1024 hidden size.
    - loss: language modeling + task fine-tune.
  - Bert (SAN)
    - 12-layer, 512 hidden size, 8 attention head.
    - loss: Masked language modeling + fine-tune.
  - ResNet (CNN)
    - 35*(2 conv-layer with 256 filter), kernel size 9, dilation rate 2.
    - loss: language model + fine-tune.

Paper II - Experiment Results

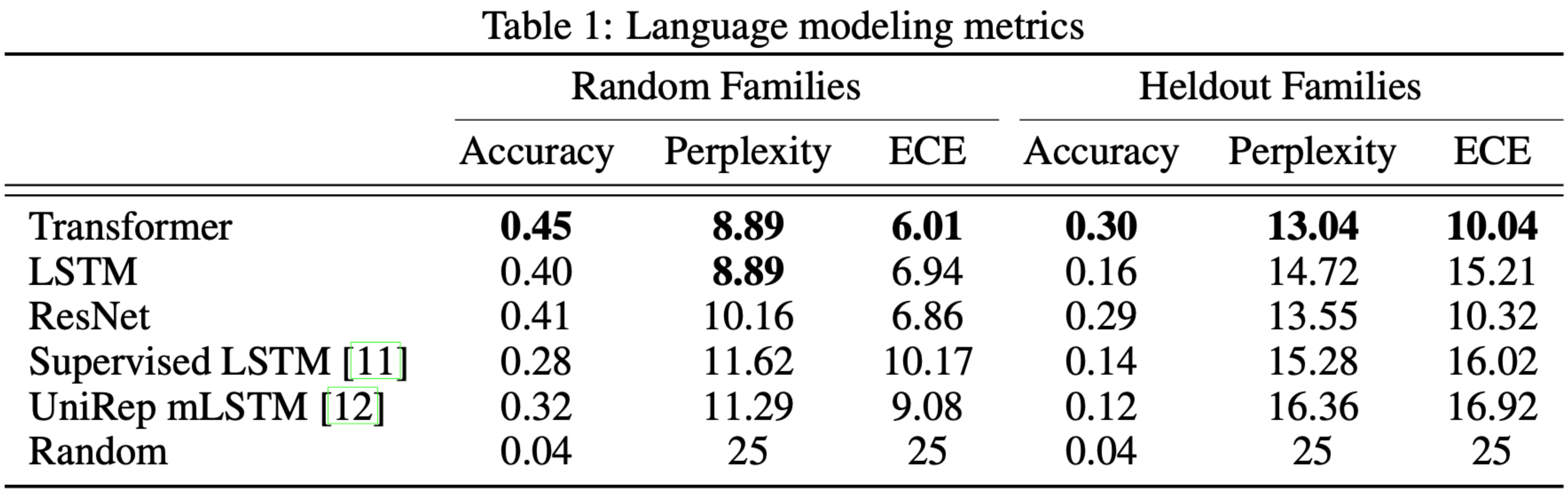


Table 1: Language modeling metrics

	Random Families			Heldout Families		
	Accuracy	Perplexity	ECE	Accuracy	Perplexity	ECE
Transformer	**0.45**	**8.89**	**6.01**	**0.30**	**13.04**	**10.04**
LSTM	0.40	**8.89**	6.94	0.16	14.72	15.21
ResNet	0.41	10.16	6.86	0.29	13.55	10.32
Supervised LSTM [11]	0.28	11.62	10.17	0.14	15.28	16.02
UniRep mLSTM [12]	0.32	11.29	9.08	0.12	16.36	16.92
Random	0.04	25	25	0.04	25	25

Paper II - Experiment Results

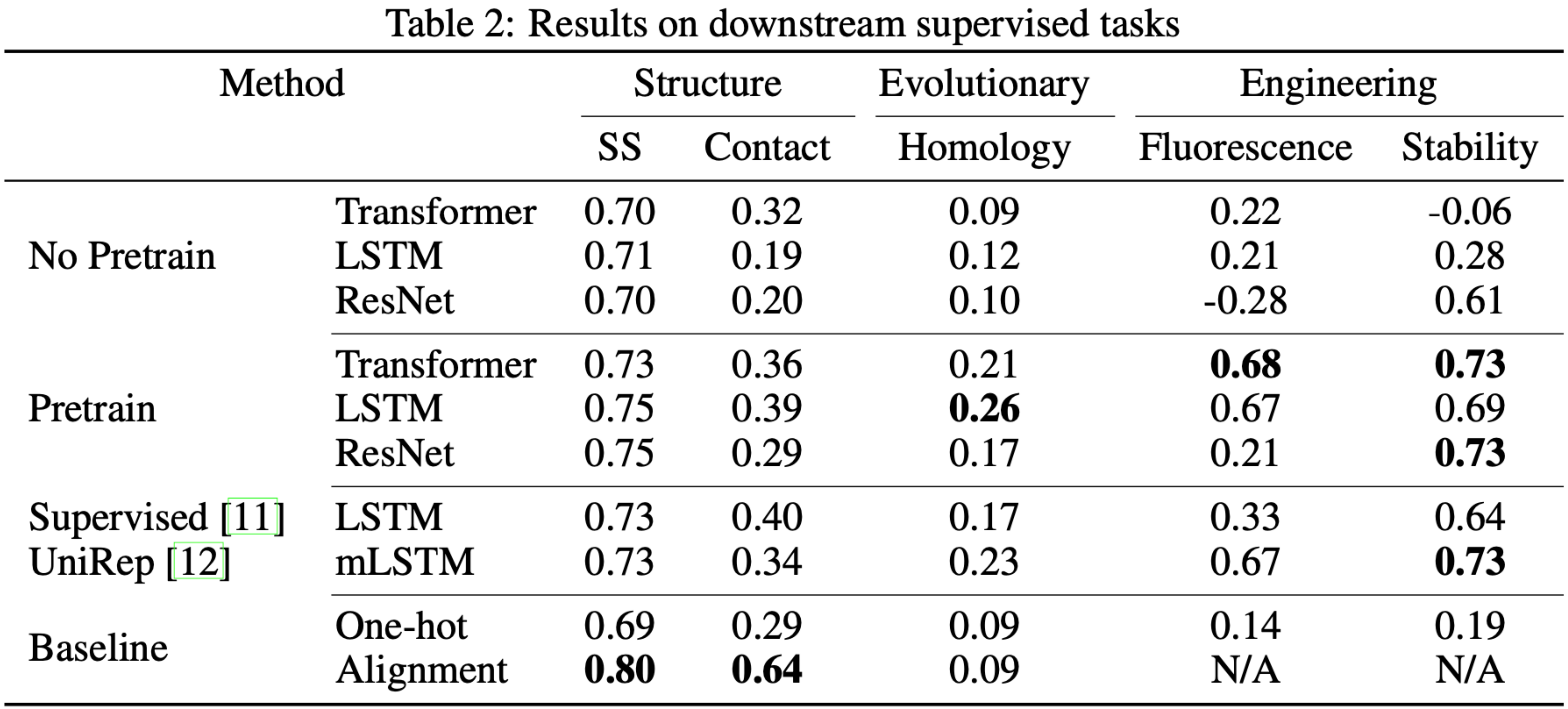


Table 2: Results on downstream supervised tasks

Method		Structure		Evolutionary	Engineering	
		SS	Contact	Homology	Fluorescence	Stability
No Pretrain	Transformer	0.70	0.32	0.09	0.22	-0.06
	LSTM	0.71	0.19	0.12	0.21	0.28
	ResNet	0.70	0.20	0.10	-0.28	0.61
Pretrain	Transformer	0.73	0.36	0.21	**0.68**	**0.73**
	LSTM	0.75	0.39	**0.26**	0.67	0.69
	ResNet	0.75	0.29	0.17	0.21	**0.73**
Supervised [11]	LSTM	0.73	0.40	0.17	0.33	0.64
UniRep [12]	mLSTM	0.73	0.34	0.23	0.67	**0.73**
Baseline	One-hot	0.69	0.29	0.09	0.14	0.19
	Alignment	**0.80**	**0.64**	0.09	N/A	N/A

Paper II - Conclusion

- The improve over labelled data shows promising future for self-supervision in protein prediction.
- No single self-supervised model performs best across all protein tasks. Needs the extensive benchmark to evaluate the models.
- Structure prediction still is inferior to the alignment method. In need for better self-supervision design and studying the relationship between alignment and learned-based representation.

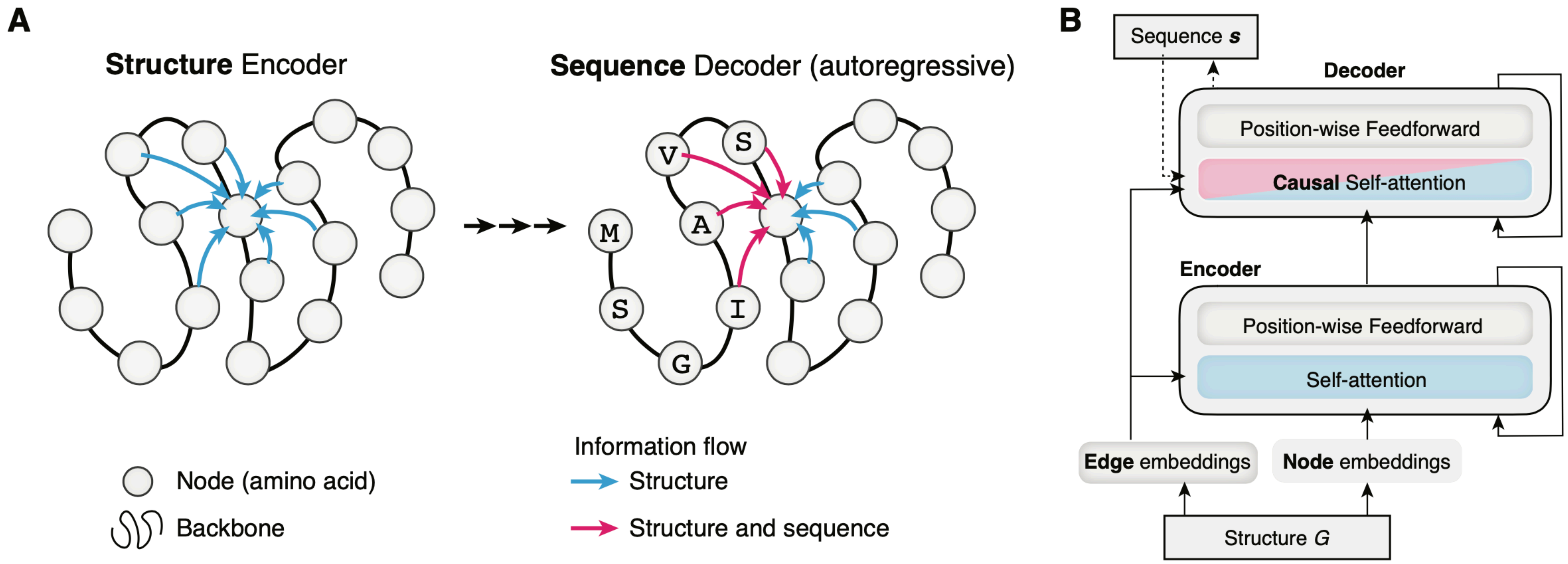
Overview

- Unified rational protein engineering with sequence-based deep representation learning (Nature Method 2019)
  - Rational protein engineering requires a holistic understanding of protein function. This paper proposed to use RNN based model to learn the holistic knowledge of protein sequences.
- Evaluating Protein Transfer Learning with TAPE (NeurIPS 2019)
  - This paper implements a more extensive comparison work between three different self-supervised models. It also provides 5 benchmark tasks and results.
- Generative models for graph-based protein design (NeurIPS 2019)
  - This paper introduces a conditional generative model for protein sequences given 3D structures based on graph representations.

Paper III - Motivation

- Protein design takes a protein and its structural information to predict its sequential form.
- Traditional methods depends on complex energy functions which are unreliable and hard to analyze the unreliability.
- This paper proposed a top-down framework that directly learns generative model from the proteins' 3D structural information, which represented as graph, to generate sequences.

Paper III - Approach



- Structured Transformer: (Graph structural features + sequence features)
  - Encoder: node feature + edge feature (only structure)
  - Decoder: node + edge + sequence feature (structure and sequence)

Paper III - Approach

- Presentation structure as a graph G = (V, E)
  - V: node feature - describing each residue (amino acid).
  - E: edge feature - relationships between edges and a node.

- For 3D cases, graph representation needs two properties:
  - Invariance to rotation and translation.
  - Locally informative, neighbor edge features contains sufficient information to reconstruct their coordinates. E.g. for a node with coordinate x_i, the pairwise distance D_ia, D_ib can not determine whether x_a and x_b are on the same side or not.

Paper III - Approach

- Based on the two properties, the structural features are designed as:
  - Relative spatial encodings:

$$\boldsymbol{e}_{ij}^{(s)} = \left(\mathbf{r}\left(||\boldsymbol{x}_j - \boldsymbol{x}_i||\right), \quad \boldsymbol{O}_i^T \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{||\boldsymbol{x}_j - \boldsymbol{x}_i||}, \quad \mathbf{q}\left(\boldsymbol{O}_i^T \boldsymbol{O}_j\right)\right)$$

    - The three terms are distance, direction, and orientation(quaternion) respectively.

$$\boldsymbol{O}_i = [\boldsymbol{b}_i \ \ \boldsymbol{n}_i \ \ \boldsymbol{b}_i \times \boldsymbol{n}_i],$$

$$\boldsymbol{u}_i = \frac{\boldsymbol{x}_i - \boldsymbol{x}_{i-1}}{||\boldsymbol{x}_i - \boldsymbol{x}_{i-1}||}, \ \ \boldsymbol{b}_i = \frac{\boldsymbol{u}_i - \boldsymbol{u}_{i+1}}{||\boldsymbol{u}_i - \boldsymbol{u}_{i+1}||}, \ \ \boldsymbol{n}_i = \frac{\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}}{||\boldsymbol{u}_i \times \boldsymbol{u}_{i+1}||}$$

    - *O_i* defines a local coordinate system at x_i
  - Relative positional encodings:
    - Represent the sequential position of each neighbor relative to a node.
    - Defined as sin(gap_i,j). Note, relative position is different from original transformer's global position.
  - Edge encoding = spatial encoding + positional encoding
  - Node encoding: three dihedral angles of the protein backbone (ϕ_i, ψ_i, ω_i) and embed these on the 3-torus (三环) as {sin, cos}×(ϕ_i, ψ_i, ω_i).

Paper III - Approach

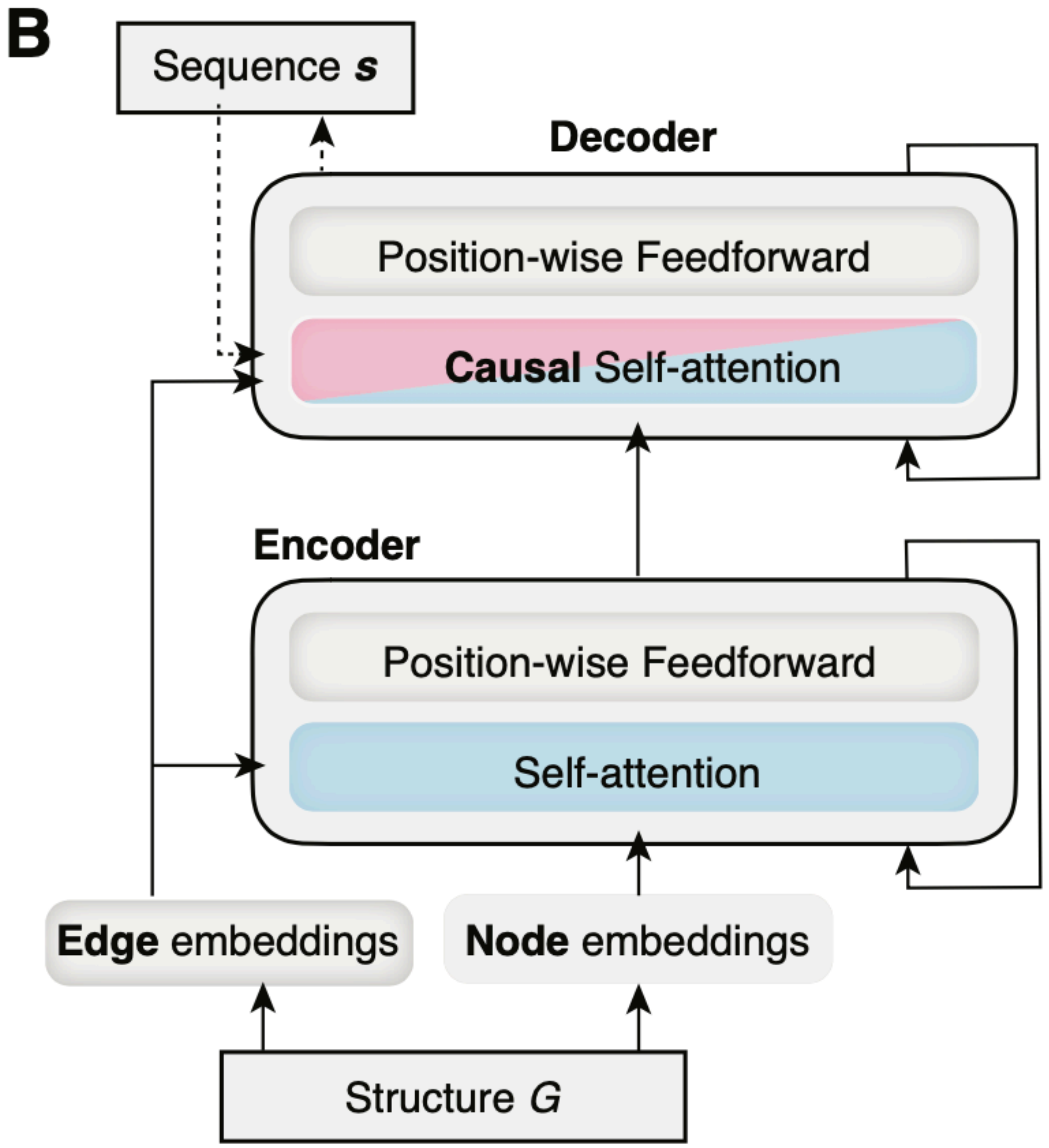
- Structural Transformer (Encoder)
  - Node embeddings:
    - h_i = W_h(v_i)
  - Self-attention:
    - query: q_i = W_q(h_i)
    - key: z_ij = W_z(r_ij), r_ij = (h_j, e_ij)
    - value: v_ij = W_v(r_ij)
    - j belongs to N(i, k), k neighbors of i.
    - Attention a_ij:

$$a_{ij}^{(\ell)} = \frac{\exp(m_{ij}^{(\ell)})}{\sum_{j' \in \mathrm{N}(i,k)} \exp(m_{ij'}^{(\ell)})}, \qquad \text{where} \quad m_{ij}^{(\ell)} = \frac{\boldsymbol{q}_i^{(\ell)\top} \boldsymbol{z}_{ij}^{(\ell)}}{\sqrt{d}}$$

    - Self-attention output:

$$\boldsymbol{h}_i^{(\ell)} = \sum_{j \in N(i,k)} a_{ij}^{(\ell)} \boldsymbol{v}_{ij}^{(\ell)},$$

$$\Delta \boldsymbol{h}_i = \boldsymbol{W}_o \operatorname{Concat}\left(\boldsymbol{h}_i^{(1)}, \ldots, \boldsymbol{h}_i^{(L)}\right)$$

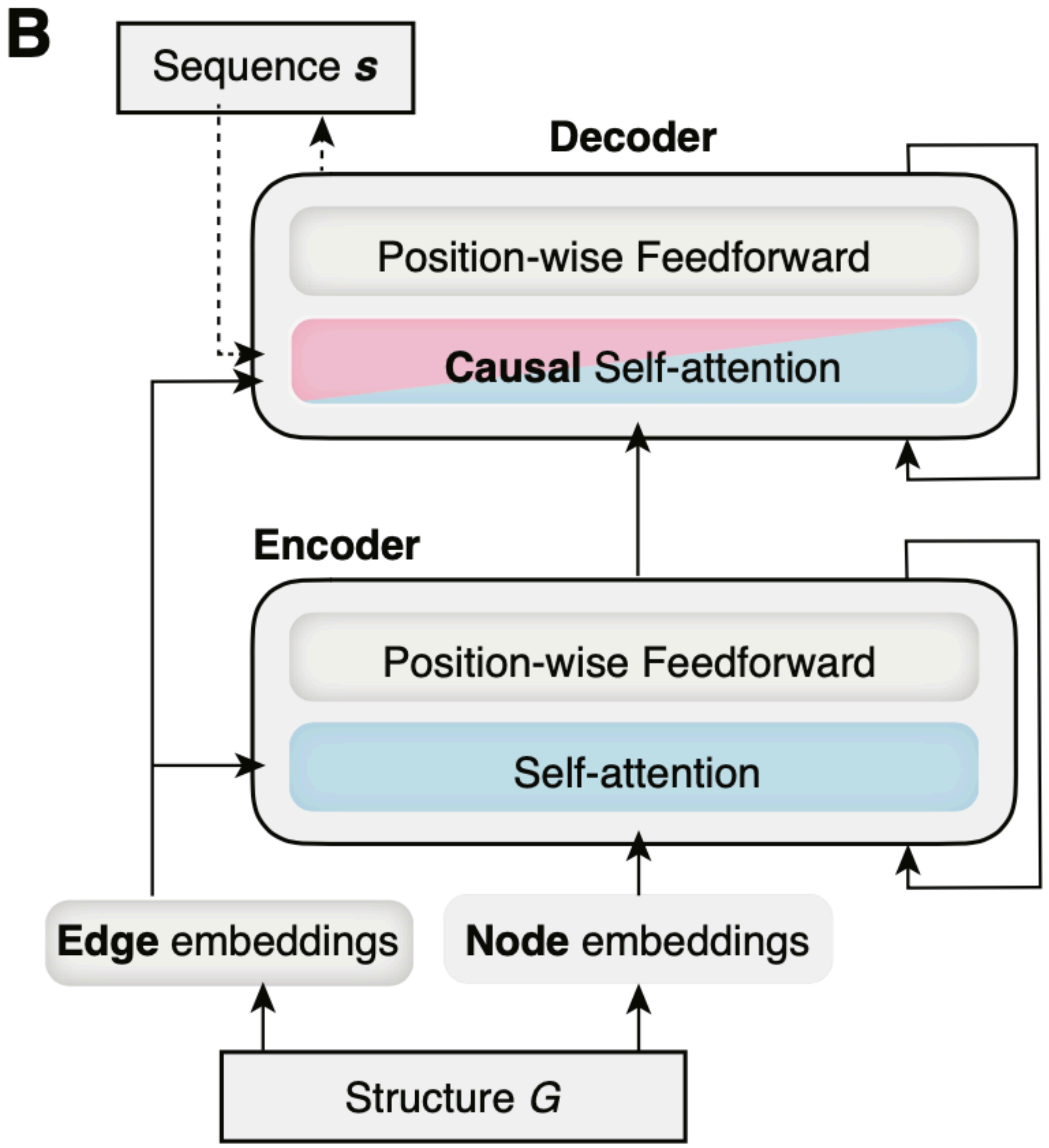


Paper III - Approach

- Structural Transformer (Decoder)
  - The same with the encoder with augmented relational information r_ij,

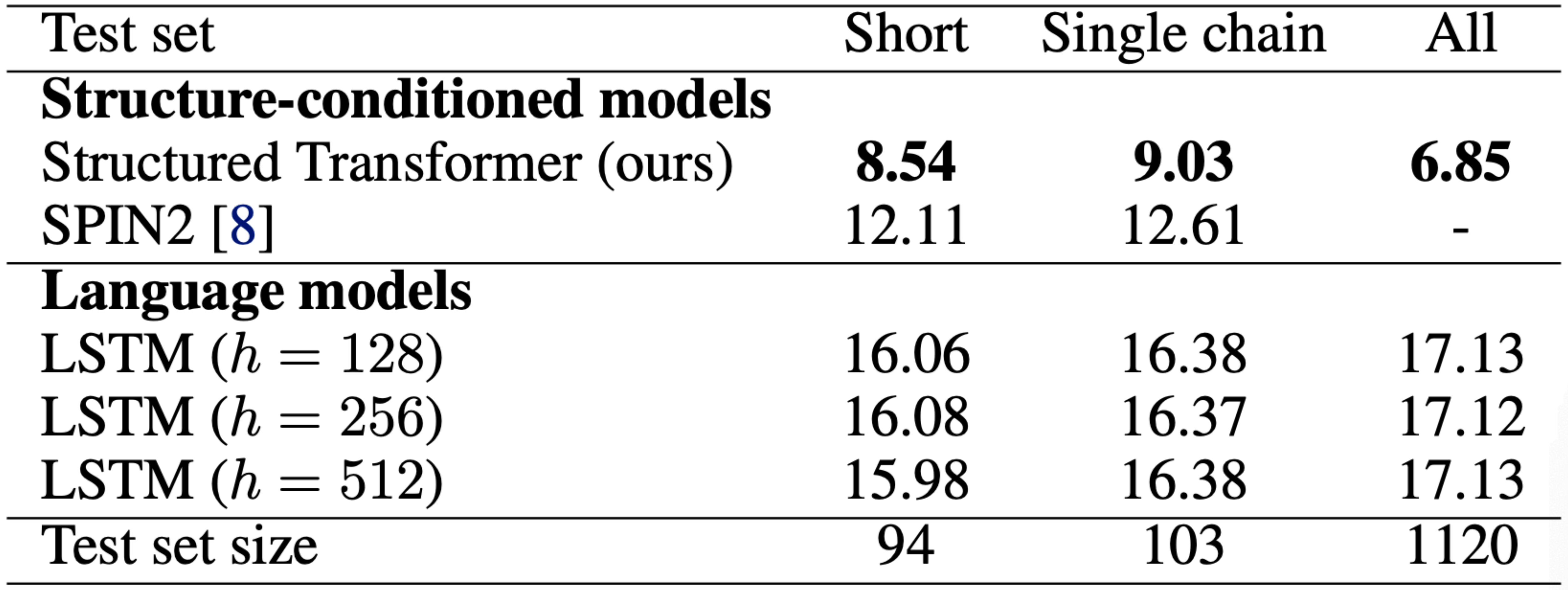
$$\boldsymbol{r}_{ij}^{(\mathrm{dec})} = \begin{cases} (\boldsymbol{h}_j^{(\mathrm{dec})}, \boldsymbol{e}_{ij}, \mathbf{g}(s_j)) & i > j \\ (\boldsymbol{h}_j^{(\mathrm{enc})}, \boldsymbol{e}_{ij}, \mathbf{0}) & i \leq j \end{cases}$$

  - g(s_j) is a sequence embedding of amino acid s_j prior to node i.
  - Historical sequential information + overall structural information of the neighbors.



Paper III - Experiments

- Training
  - Architecture: 3-layers, hidden_size = 128.
  - Optimization: learning and initialization same with transformer, dropout = 10%, label_smoothing = 10%.
- Dataset:
  - CATH 4.2, 18024 train, 608 valid, 1120 test.
  - Zero overlap.
- Main result: in terms of perplexity, the lower the better.

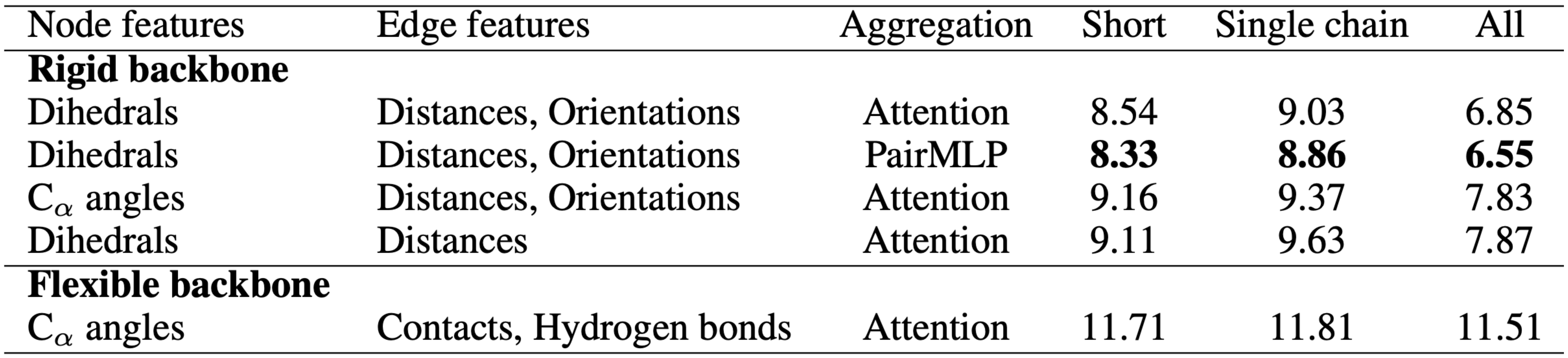


Test set	Short	Single chain	All
Structure-conditioned models			
Structured Transformer (ours)	**8.54**	**9.03**	**6.85**
SPIN2 [8]	12.11	12.61	-
Language models			
LSTM ($h = 128$)	16.06	16.38	17.13
LSTM ($h = 256$)	16.08	16.37	17.12
LSTM ($h = 512$)	15.98	16.38	17.13
Test set size	94	103	1120

Paper III - Experiments

- Ablation study

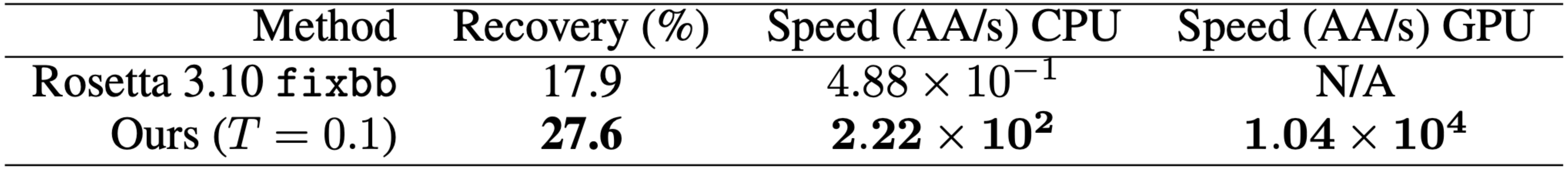
Table 3: **Ablation of graph features and model components**. Test perplexities (lower is better).



Node features	Edge features	Aggregation	Short	Single chain	All
Rigid backbone					
Dihedrals	Distances, Orientations	Attention	8.54	9.03	6.85
Dihedrals	Distances, Orientations	PairMLP	**8.33**	**8.86**	**6.55**
C_α angles	Distances, Orientations	Attention	9.16	9.37	7.83
Dihedrals	Distances	Attention	9.11	9.63	7.87
Flexible backbone					
C_α angles	Contacts, Hydrogen bonds	Attention	11.71	11.81	11.51

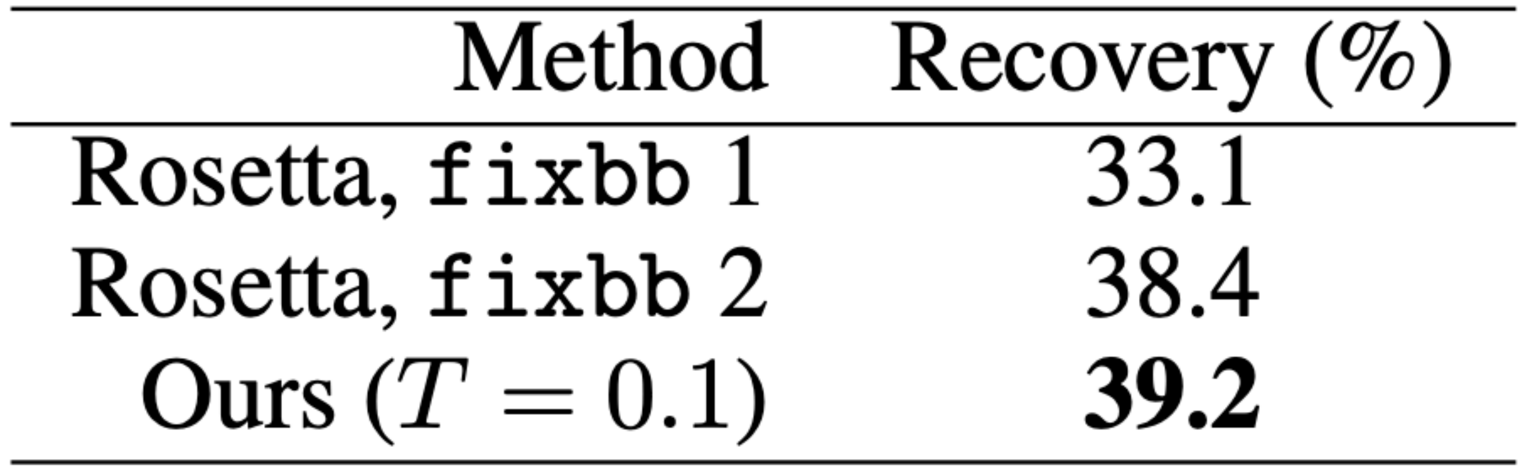
Paper III - Experiments

- Compare with SOTA Rosetta model:



Method	Recovery (%)	Speed (AA/s) CPU	Speed (AA/s) GPU
Rosetta 3.10 `fixbb`	17.9	4.88×10^{-1}	N/A
Ours ($T = 0.1$)	**27.6**	$\mathbf{2.22 \times 10^{2}}$	$\mathbf{1.04 \times 10^{4}}$

(a) Single chain test set (103 proteins)



Method	Recovery (%)
Rosetta, `fixbb` 1	33.1
Rosetta, `fixbb` 2	38.4
Ours ($T = 0.1$)	**39.2**

(b) Ollikainen benchmark (40 proteins)

Paper III - Conclusion

- New generative model with 3D graph representation.
- Augment original transformer with structural encoding to leverage spatial locality of dependencies in molecular structures.
- Improves perplexity, accuracy and speed.
- Underscores the importance of modeling sparse, long-range dependencies in biological sequences.