SAINT: **S**elf-**A**ttention Augmented **I**nception-Inside-Inception **N**e**t**work Improves Protein Secondary Structure Prediction

Mostofa Rafid Uddin[1,†], Sazan Mahbub[1,†], M Saifur Rahman[1], and Md Shamsuzzoha Bayzid[1,*]

[1]Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
[†]These authors contributed equally to this work
[*]Corresponding author: shams_bayzid@cse.buet.ac.bd

Abstract

Protein structures provide basic insight into how they can interact with other proteins, their functions and biological roles in an organism. Experimental methods (e.g., X-ray crystallography, nuclear magnetic resonance spectroscopy) for predicting the secondary structure (SS) of proteins are very expensive and time consuming. Therefore, developing efficient computational approaches for predicting the secondary structure of protein is of utmost importance. Advances in developing highly accurate SS prediction methods are mostly constrained in 3-class (Q3) structure prediction. However, 8-class (Q8) resolution of secondary structure contains more useful information and is much more challenging than the Q3 prediction. We present SAINT, a highly accurate method for Q8 structure prediction, which incorporates self-attention mechanism (a concept from natural language processing) with the Deep Inception-Inside-Inception (Deep3I) network in order to effectively capture both the *short-range* and *long-range dependencies* among the amino acid residues. SAINT offers a more interpretable framework than the typical black-box deep neural network methods. We report, on an extensive evaluation study, the performance of SAINT in comparison with the existing best methods on a collection of benchmark dataset (CB513, CASP10, and CASP11). Our results suggest that self-attention mechanism improves the prediction accuracy and outperforms the existing best alternate methods. SAINT is the first of its kind and offers the best known Q8 accuracy and interpretable results. Thus, we believe SAINT represents a major step towards the accurate and reliable prediction of secondary structures of proteins. We have made SAINT freely available as open source code at `https://github.com/SAINTProtein/SAINT`.

Keywords: Protein secondary structure, deep learning, self-attention.
Running title: SAINT: Accurate and interpretable secondary structure prediction.

1 Introduction

Proteins are bio-molecules made of long chains of amino acid residues connected by peptide bonds. The functions of proteins are usually determined by the their tertiary structure and for determining the tertiary structure and related properties, the secondary structure information is crucial. Protein structure can be experimentally determined by X-ray crystallography and multi-dimensional magnetic resonance in laboratory, but these methods are very costly and time consuming and are yet to be consistent with the proliferation of protein sequence data [1]. Thus, the proteins with known primary sequence continue to outnumber the proteins with experimentally determined secondary structures. The structural properties of a protein depend on its primary sequence [2–5], yet it remains as a difficult task to accurately determine the secondary and tertiary structures of proteins. Hence, the problem of predicting the structures of a protein – given its primary sequence – is crucially important and remains as one of the greatest challenges in computational biology.

Secondary structure – a conformation of the local structure of the polypeptide backbone – prediction dates back to the work of Pauling and Corey in 1951 [6]. The secondary structures of proteins are traditionally characterized as 3 states (Q3): helix (H), strand (E), and coil (C). Afterwards, a more fine-grained characterization of the secondary structures was proposed [7] for more precise information by extending the three states into eight states (Q8): α-helix (H), 3_{10}-helix (G), π-helix (I), β-strand (E), isolated β-bridge (B), turn (T), bend (S), and Others (C). Q8 prediction is more challenging and can reveal more precise and high resolution on the structural properties of proteins.

Protein secondary structure prediction is an extensively studied field of research [8–30]. Developing computational approaches (especially using machine learning techniques) for 3-state SS prediction has a long history which dates back to the works of Qian & Sejnowski [8] and Holley & Karplus [9] who first used neural networks to predict SS. In the 1980s, only statistical model based methods were used on raw sequence data which could ensure Q3 accuracy merely below 60%. Afterwards, significant improvement was achieved [10–12] by leveraging the evolutionary information such as the position-specific score matrices (PSSM) derived from multiple sequence alignments. Subsequently, many machine learning methods have been developed for Q3 prediction which include support vector machines (SVM) [13–15, 31], probabilistic graphical models [16, 32, 33], hidden Markov models [17, 18], bidirectional recurrent neural networks [19–22, 34, 35], and deep learning frameworks [23].

The performance of Q3 prediction methods has approached the postulated theoretical limit [24]. At the same time, there has now been a growing awareness that 8-state prediction can reveal more valuable structural properties. As such, the interest of the research community has recently shifted from Q3 prediction to relatively more challenging Q8 prediction. Quite a few deep learning methods for Q8 prediction have been proposed over the last few years [19,25,26,28–30,36]. To the best of our knowledge, the first notable success in Q8 prediction methods was SSpro8 [19] which was published in 2002 and achieved 63.5% Q8 accuracy on the benchmark CB513 dataset [37], 64.9% on CASP10 and 65.6% on CASP11 [25]. Later in 2011, RaptorX-SS8 [36], another 8 state predictor using conditional neural fields, surpassed SSpro8 by demonstrating 64.9% Q8 accuracy on CB513. In 2014, Zhou and Troyanskaya [26] highlighted the challenges in 8-state prediction and obtained 66.4% Q8 accuracy on CB513 dataset using deep generative stochastic network (GSN). One of their major contributions was making their training dataset *CB6133* publicly available. Afterwards many used their dataset for training various deep learning architectures and tried to improve Q8 accuracy on the CB513 dataset. Some of the notable works include deep conditional random fields (DeepCNF) [25], cascaded convolutional and recurrent neural network (DCRNN) [27], next-step conditioned deep convolutional neural network (NCCNN) [28], multi-scale CNN with highway (CNNH_PSS) [29], deep inception-inside-inception (Deep3I) network named MUFOLD-SS [30], and DeepACLSTM [38] with an asymmetric convolutional neural networks (ACNNs) combined with bidirectional long short-term memory (BLSTM). CRRNN (Convolutional, residual, and recurrent neural network) [39] represents another class of methods that uses various physical properties (e.g., e.g., steric parameters (graph-shape index), polarizability, normalized van der Waals volume, hydrophobicity, etc.) in addition to the primary sequence and position-specific scoring matrix (PSSM). Although these works demonstrate a steady improvement in the published Q8 accuracy over the past few years, the improvements across successive publications are very small. Yet, these small improvements are considered significant given the high complexity of 8-state SS prediction.

Usually the models that focus more on short range dependencies (local context of the amino acid residues) face difficulties in effectively capturing the long range dependencies (interactions between amino acid residues that are close in three-dimensional space, but far from each other in the primary sequence) [22, 27, 40]. Various deep learning based models have been leveraged to handle the long-range interactions by using recurrent or highway networks [28,29], deeper networks with convolutional blocks [30], long short-term memory (LSTM) cells [22,27], whereas the short-range interactions have been handled by convolutional blocks of smaller window size [27, 28, 30]. These methods circumvent some challenging issues in capturing the non-local interactions, but have limitations of their own. Models, using recurrent neural networks to capture long range dependencies, may suffer from *vanishing gradient* or *exploding gradient* problems [41–44]. Moreover, these methods may fail to effectively capture the dependencies when the sequences are very long [45]. Furthermore, as the models grow deeper, the number of parameters also grows which makes it prone to over-fitting. It is also likely that the short range relationships captured in the earlier (shallow) layers may disappear as the models grow deeper [29]. As a result, developing techniques which can capture both long-range and short-range dependencies simultaneously is of utmost importance. Another limiting factor of the deep learning methods is that the high accuracy comes at the expense of high abstraction (less interpretability) due to their black-box nature [46–49]. Although there has been a flurry of recent works towards designing deep learning techniques for bio-molecular data, no notable attempt has been made in developing methods with improved interpretablity and explainability – models that are able to summarize the reasons of the network behavior, or produce insights about the causes of their decisions and thus gain trust of users.

In this study, we present SAINT (**S**elf-**A**ttention Augmented **I**nception Inside Inception **N**e**T**work) – a novel method for 8-state SS prediction which uniquely incorporates the *self-attention mechanism* [50] with a state-of-the-art Deep Inception-Inside-Inception (Deep3I) network [30]. We proposed a novel architecture called attention-augmented 3I (2A3I) in order to capture both the local- and long-range interactions. SAINT was compared with a collection of the best alternate methods for Q8 prediction on the publicly available benchmark dataset (CB513, CASP10, and CASP11). It demonstrated Q8 accuracy (70.9% on CB513, 74.7% on CASP10, 73.3% on CASP11) that is superior than or indistinguishable from the other state-of-the-art methods, and obtained high precision, recall and $F1$-score for individual states. Moreover, SAINT provides interesting insights regarding the interactions and roles of amino acid residues while forming secondary structures, which help to interpret how the predictions are made. Hence, we have made the following significant contributions: 1) we, for the first time, successfully translated the success of self-attention mechanism from natural language processing to the domain of protein structure prediction, and demonstrated that self-attention improves the accuracy SS prediction, 2) introduced a method which can capture both the short- and long-range dependencies, and offers the best known Q8 accuracy, and 3) improved the interpretability of the black-box deep neural network based methods which are often criticized for lack of interpretability.

2 Materials and Methods

2.1 Feature Representation

SAINT takes a protein sequence feature vector $X = (x_1, x_2, x_3, \ldots, x_N)$ as input, where $x_i \epsilon \mathbb{R}^d (d = 43)$ is the vector corresponding to the i^{th} residue, and it returns the protein structure label sequence vector $Y = (y_1, y_2, y_3, \ldots, y_N)$ as output, where $y_i \in \mathbb{R}^s$ $(s = 8)$ is the structure label (one of the eight possible states) of the i^{th} residue. The dimension of x_i is 43 as this is the concatenation of both sequence information $x\_seq_i \in \mathbb{R}^{d_{seq}} (d_{seq} = 22)$ and evolutionary information $x\_pssm_i \in \mathbb{R}^{d_{pssm}}$ $(d_{pssm} = 21)$ of the i^{th} residue. In the training dataset *CB6133*, each protein is encoded as a vector of dimension 700, and so the proteins having less than 700 amino acid residues are padded with $NoSeq$ at the end [26]. Proteins containing more than 700 residues are excluded from *CB6133*, however *CB513* contains a protein having more than 700 amino acids and it is splited into two overlapping sequences. The sequence information x_seq is encoded as a 22-dimensional *one-hot vector*, where 21 dimensions represent 21 different amino acids (including 'X' for unknown or unspecified amino acids) and the remaining dimension is for $NoSeq$ marker (a marker used for padding the sequence). The evolutionary information x_pssm is represented in the form of a position specific scoring matrix (PSSM) [12]. To generate PSSM, PSI-BLAST [51] was run against Uniref90 database [52] with inclusion threshold 0.001 and three iterations. PSI-BLAST returns PSSM matrix of dimension $L \times 21$, where L is the size of a protein query sequence. This feature representation is similar to what was proposed by Zhou and Troyanskaya [26], and was subsequently used by [25, 27–29].

2.2 Architecture of SAINT

The architecture of SAINT can be split into three separate discussions: 1) the architecture of our proposed self-attention module, 2) the architecture of the existing inception module and the proposed attention augmented inception module, and finally 3) the overall pipeline of SAINT.

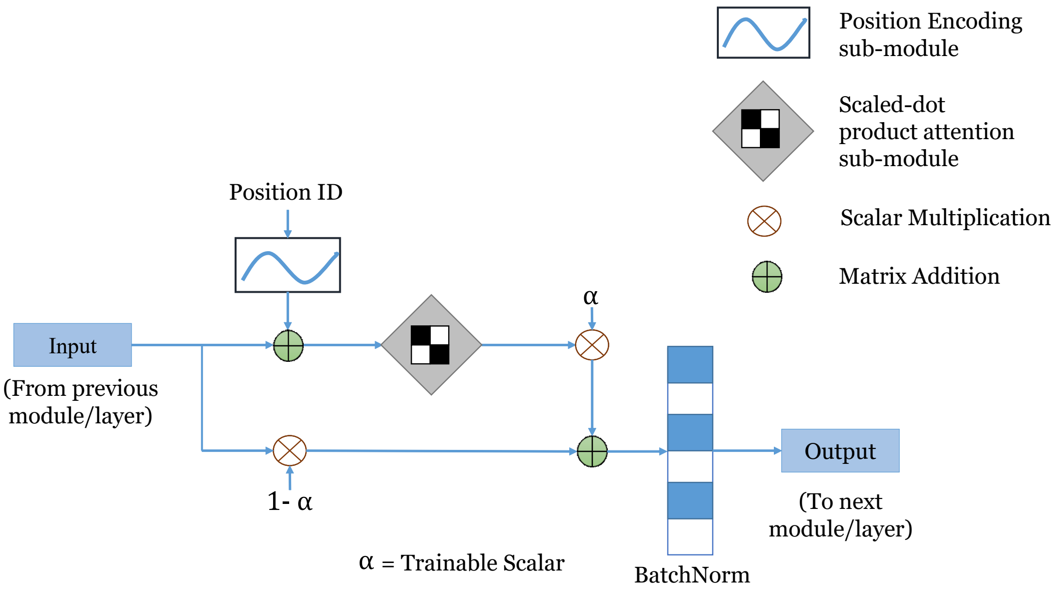

Figure 1: Architecture of the self-attention module used in SAINT.

2.2.1 Self-Attention module

Attention mechanism implies paying attention to specific parts of input data or features while generating output sequence [50,53]. It calculates a probability distribution over the elements in the input sequence and then takes the weighted sum of those elements based on this probability distribution while generating outputs.

In self-attention mechanism [50, 54, 55], each vector in the input sequence is transformed into three vectors- *query*, *key* and *value*, by three different functions. Each of the output vectors is a weighted sum of the *value* vectors, where the weights are calculated based on the compatibility of the *query* vectors with the *key* vectors by a special function, called *compatibility function* (discussed later in this section).

The self-attention module we designed and augmented with the inception modules is inspired from the self-attention module proposed by Vaswani *et al.* [50] and is depicted in Fig. 1. Our self-attention module takes two inputs: 1) the features from the previous inception module, $x \,\epsilon\, \mathbb{R}^{d_{protein} \,\times\, d_{feature}}$, and 2) position identifiers, $pos\_id \,\epsilon\, \mathbb{R}^{d_{protein}}$, where $d_{protein}$ is the length of the protein sequence, and $d_{feature}$ is the length of the feature vector for each position coming from the previous layer or module.

Positional Encoding Sub-module. The objective of positional encodings is to inject some information about the relative or absolute positions of the residues in a protein sequence. The *Positional Encoding* PE_{pos} for a position pos can be defined as follows [50].

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{feature}}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{feature}}) \tag{2}$$

where i is the dimension. We used such function as it may allow the model to easily learn to attend by relative positions since for any fixed offset k, PE_{pos+k} can be represented

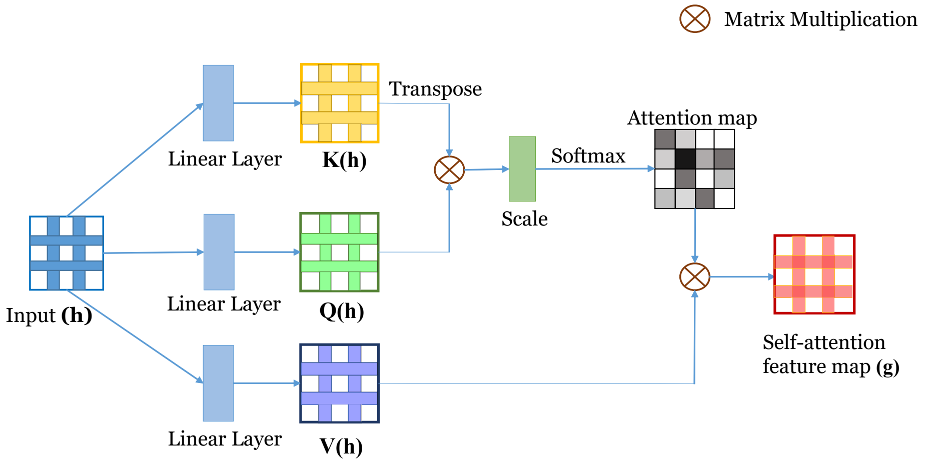

Figure 2: Architecture of the scaled dot-product attention sub-module.

as a linear function of PE_{pos} [50]. For every position pos, PE_{pos} has the dimension $d_{protein} \times d_{feature}$. The output of positional encoding is added with the inputs x, resulting in new representations h (see Eqn. 3) which contain not only the information extracted by the former layers or modules, but also the information about individual positions.

$$h_{pos} = x_{pos} + PE_{pos}. \tag{3}$$

Scaled dot-product attention sub-module The input features in this sub-module, $h \, \epsilon \, \mathbb{R}^{d_{protein} \times d_{feature}}$ are first transformed into three feature spaces Q, K and V, representing query, key and value respectively, in order to compute the scaled-dot attention, where $Q(h) = W_Q h$, $K(h) = W_K h$, $V(h) = W_V h$. Here W_Q, W_K, W_V are parameter matrices to be learned.

Among various compatibility functions (e.g. scaled dot-product attention [50], additive-attention [53], similarity-attention [56], multiplicative-attention [57], biased general attention [58], etc.), we have chosen the scaled dot-product attention as it showed much promise in case of sequential data. Vaswani *et al.* [50] showed that in practice, the dot-product attention is much faster and space-efficient as it can be implemented using highly optimized matrix multiplication code, though theoretically both dot-product and additive attention have similar complexity. Scaled dot-product $s_{i,j}$ of two vectors h_i and h_j is calculated as shown in Equation 4.

$$s_{i,j} = \frac{Q(h_i)K(h_j)^T}{\sqrt{d_K}} \tag{4}$$

where d_K is the dimension of the feature space K. The numerator of the equation, $Q(h_i)K(h_j)^T$ is the dot product between these two vectors, resulting in the similarity between them in a specific vector space. Here $\sqrt{d_K}$ is the scaling factor which ensures that the result of the dot product does not get prohibitively large for very long sequences.

The attention weights $e \in \mathbb{R}^{d_{protein} \times d_{feature}}$ are calculated as shown in Equation 5, where $e_{j,i}$ represents how much attention have been given to the vector at position i while synthesizing the vector at position j.

$$e_{j,i} = \frac{exp(s_{i,j})}{\sum_{n=1}^{d_{protein}} exp(s_{i,j})} \tag{5}$$

The attention distribution e is multiplied with the feature space V and then in order to reduce the internal covariate shift this multiplicand is normalized using *batch normalization* [59], producing g, the output of the scaled dot-product attention sub-module, following the Equation 6.

$$g_j = BatchNorm(\sum_{n=1}^{d_{protein}} e_{j,i}V(h_i)) \tag{6}$$

Here, $BatchNorm$ is the batch-normalization function and g_j is the j-th vector in the output sequence of this sub-module. Finally, according to the Equation 7, the output of the scaled dot-product attention module g is multiplied by a scalar parameter α, the original input feature map x is multiplied by $(1 - \alpha)$ and these two multiplicands are summed to synthesize the final output y.

$$y_i = (\alpha)g_i + (1 - \alpha)x_i \tag{7}$$

where y_i is the ith output and α is a learnable scalar. By introducing weighed sum of g_i and x_i, we give our model the freedom to chose how much weight should be given to each of the features maps, g_i and x_i while generating the output y_i. The optimal value of the parameter α is learnt through back propagation along with the rest of the model.

We have observed in our experiments that the final results are not much sensitive to the initial choice of α. In most cases the training learns a value for α within the range $0.18 \sim 0.3$. However, the further away the initial value α is from the above mentioned range, the longer it takes for the model to converge. As such, for quick reproducibility of our model, we have set the initial value of α to 0.2 in our model training scripts.

2.2.2 Attention augmented inception-inside-inception (2A3I) module

A novel deep convolutional neural network architecture, *Inception*, was first introduced by Szegedy *et al.* [60], which demonstrated state-of-the-art performance for image classification and detection. An inception module has several branches, each having one or more convolutional layers. Fang *et al.* used an assembly of inception modules, which they call Inception-inside-Inception (3I module), in their proposed method MUFOLD-SS to predict protein secondary structure. They tried to leverage the inception blocks to retrieve both short-range and long-range dependencies and achieved the best known accuracy at the time. However, convolutional layers cannot capture enough information about long-range similarities or dependencies among feature vectors of a sequence, synthesized by a certain level of the network [61]. In protein secondary structure prediction, this issue leaves more impact on the overall accuracy when the sequence grows in length. Though these types of neural networks that use only convolutional layers need to be deeper to capture the long range dependency, it is often not feasible to add arbitrarily large numbers of layers. Moreover, the authors of MUFOLD-SS showed that using more than two inception-inside-inception modules sequentially does not result into significant increase in the overall accuracy, rather increases the computational expense. Earlier works [19, 20, 22, 27, 62, 63] used Recurrent Neural Network(RNN) based architectures for capturing global features, but incorporating RNN or its derivatives (Gated Recurrent Units (GRU) [64], Long Short Term Memory (LSTM) [65]) inside 3I module would escalate the complexity and computational cost of the model. Therefore, we incorporated the self-attention mechanism to effectively capture both the short-range and long-range dependencies and to bring a better balance between the ability to model long-range dependencies and the computational efficiency. We placed our self attention modules in each branch of the 3I module as shown in Figure 3. We call this an attention augmented inception-inside-inception (2A3I) module.

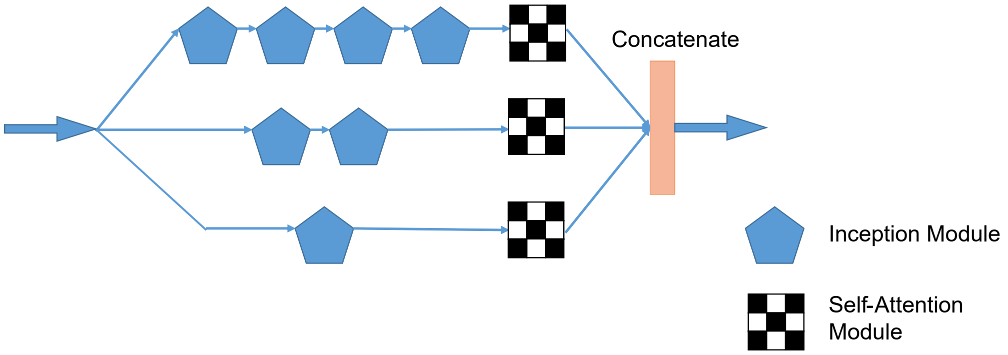

Figure 3: Architecture of our proposed 2A3I module by augmenting self-attention within the inception-inside-inception (3I) network.

2.2.3 Overview of SAINT

A schematic diagram of the overall architecture of SAINT is depicted in Fig. 4. SAINT starts with two consecutive 2A3I modules followed by a self-attention module to supplement the non-local interactions captured by the initial two 2A3I modules. We also observed that this attention module helps achieve faster learning rate. MUFOLD-SS used one convolutional layer with window size 11 after two 3I modules. The level of long-range interactions being captured varies with varying lengths of the window. However, we observed that using window size larger than 11 increases the computational cost without significantly increasing the performance. As a result, we used similar convolutional layer as MUFOLD-SS. However, we included another self-attention module after the convolutional layer to help capture the relations among vectors that the convolutional layer failed to retrieve. The last two dense layers in the MUFOLD-SS were also used in SAINT. However, we placed an attention module in between the two dense layers. We did so to understand how the residues align and interact with each other just before generating the output. This paves the way to have an interpretable deep learning model (as we will discuss in Sec. 3.2.1).

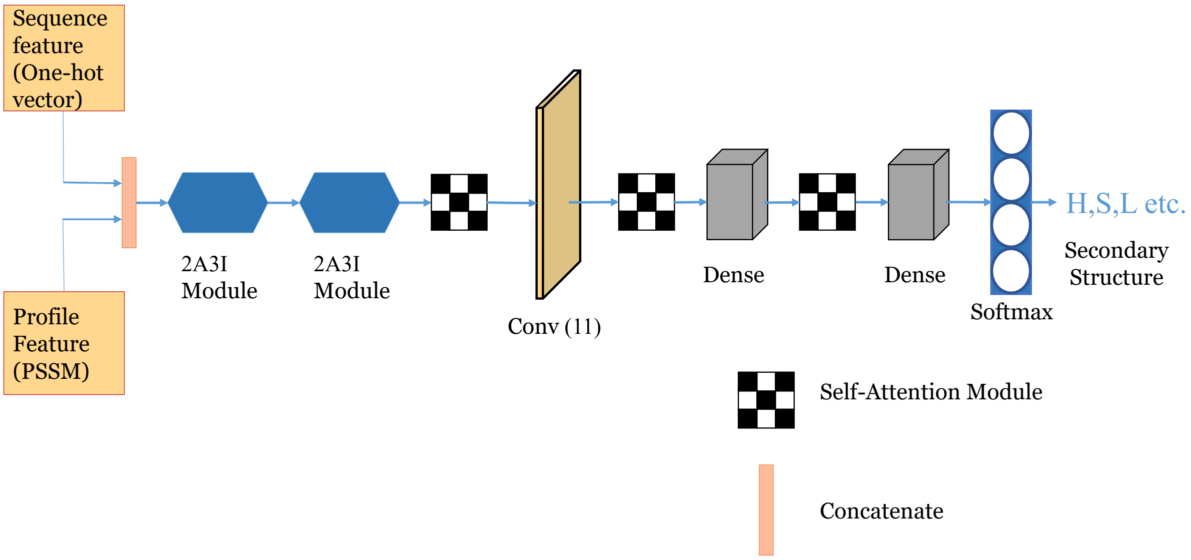

Figure 4: **A schematic diagram of the overall architecture of SAINT.** It comprises two 2A3I modules, three self-attention modules, convolutional layers with window size 11 and two dense layers.

3 Results and discussion

We performed an extensive evaluation study, comparing SAINT with a collection of state-of-the-art Q8 prediction methods on three publicly available benchmark dataset.

3.1 Dataset

We evaluated SAINT on three publicly available and widely used benchmark dataset: CB513, CASP10 and CASP11. We trained our model on a subset of the publicly available CB6133 dataset [66] which was carefully and appropriately filtered by Zhou and Troyanskaya [26].

3.1.1 CB513

CB513 dataset is developed by Cuff and Barton [37] and comprises 513 protein sequences and 84,107 residues. It is the most widely used benchmark dataset for evaluating protein secondary structure prediction methods [25–30, 38]. This dataset, preprocessed by Zhou and Troyanskaya [26], is publicly availabe at `https://www.princeton.edu/~jzthree/datasets/ICML2014/` (last accessed June, 2019).

3.1.2 CASP

CASP stands for Critical Assessment of protein Structure Prediction. This is an biennial competition for protein strcuture prediction and a community wide effort to advance the state-of-the-art in modelling protein structure from its amino acid sequences since 1994 [67]. Among the CASP datasets, we analyzed the widely used CASP10 and CASP11 dataset which contains 123 and 105 domain sequences respectively. They are publicly available at `https://github.com/icemansina/IJCAI2016` [27] (last accessed June, 2019).

3.1.3 CB6133

This dataset was originally produced with PISCES CullPDB server [66] containing 6128 proteins. Zhou and Troyanskaya later modified the dataset by retrieving those proteins sharing less than 30% identity and having better than 2.5Å resolution and made the dataset publicly available [26]. Since some sequences of CB6133 and CB513 were homologous, CB6133 was not directly used for training to ensure a fair evaluation. The sequences having $\geq$ 25% sequence similarity with CB513 were filtered from CB6133, resulting into a set of 5534 proteins. Among the remaining ones, some were duplicates which were removed by the authors [68] and thus a collection of 5365 protein sequences remained. The filtered and duplication free dataset is available at: `https://www.princeton.edu/~jzthree/datasets/ICML2014` named as *'cullpdb+profile_5926_filtered'*. We used this filtered dataset to train SAINT.

3.2 Results on benchmark dataset

We compared SAINT with a collection of existing popular Q8 predictors: SSPro8 [19], RaptorX-SS8 [36], DeepGSN [26], DeepCNF [25], DCRNN [27], NCCNN [28], CNNH_PSS [29], CBRNN [63] and MUFOLD-SS [30]. Another relevant method CRRNN [39] reported 71.4% accuracy on the CB513 dataset but its training dataset is twice as large as the one used by the other methods, and uses several extra features including seven physical properties (e.g., steric parameters (graph-shape index), polarizability, normalized van der Waals volume, hydrophobicity, isoelectric point, helix probability, and sheet probability). Therefore, similar to other studies [38, 68], we exclude CRRNN from the evaluation study to ensure a fair comparison. MUFOLD-SS, which achieves one of the best known Q8 accuracies, was not trained on the filtered CB6133. Rather the authors used their own dataset consisting of 9581 proteins from the CullPDB dataset [66], of which 9000 proteins were used for training and 581 for validations. Since the source code and the dataset used for training are not publicly available, we implemented their method and trained it on the filtered CB6133 dataset. In the spirit of reproducible research, we made our implementation of MUFOLD-SS freely available at `https://github.com/SAINTProtein/MUFOLD-SS`. The comparison of SAINT with other well-known existing methods on CB513, CASP10 and CASP11 is shown in Table 1. The reported accuracies for various methods are based on training the models on the CB6133 dataset, and are obtained from respective publications. SAINT achieves 70.91% Q8 accuracy on the benchmark CB513 dataset, outperforming all the methods by a significant margin. MUFOLD-SS was the second best method with 70.5% accuracy. The accuracy of MUFOLD-SS was also reported to be 70.5% in [39]. However, the accuracy was reported to be 70.63% when it was trained on a much larger dataset containing 9000 proteins [30]. Irrespective of the choice of training dataset for MUFOLD-SS, SAINT is much better than MUFOLD-SS. In addition to the model accuracy, we also investigated the *precision*, *recall* and $F1$-score to obtain better insights on the performances of various methods. Precision, also know as predictivity, denotes the confidence that can be imposed on a prediction. Recall signifies how accurately an algorithm can predict a sample from a particular class. Sometimes an algorithm tends to over-classify which results into high recall but low precision. On the otherhand, some algorithms tend to under-classify, preserving the precision at the cost of recall. In order to get an unbiased evaluation of the performance, $F1$-score is considered to be an appropriate measure and has been being used for over 25 years in various domains [69, 70]. Tables 2, 3, and 4 show the precision, recall and $F1$-score on each of the 8 states obtained by SAINT and other state-of-the art methods. These results suggest that SAINT achieves better $F1$-score than other methods on 5 states (out of 8 states), showing that SAINT produced more balanced and meaningful results than other methods. SAINT substantially outperforms MUFOLD-SS and NCCNN on the non-ordinary states such as G, S and T. However, NCCNN achieved better $F1$-score for the loop (L) state. State 'I' is extremely rare in the test set (only 30 out of 84,765 residues in CB513), and is hard to predict [71].

SAINT also achieved state-of-the-art accuracy on the CASP dataset. On CASP10, SAINT achieved 74.7% accuracy which is lower than DCRNN, but higher than MUFOLD-SS and others. On CASP11, SAINT achieved the best known accuracy of 73.3% outperforming the previous best accuracy (73.1%) achieved by DCRNN.

In order to assess the performance of SAINT in capturing the continuous structure of a protein, we visualized the structures predicted by SAINT for a few proteins from CASP11 and CB513, and superimposed them on the 3D structures obtained from PDB (see Fig. 5). We selected three proteins: 1) T0821-D1:20-274 in CASP11 (PDB ID: 4r7s), for which SAINT achieved the highest accuracy on CASP11 (92.94%), 2) T0840-D2:546-661 in CASP11 (PDB ID: 4qt8) where SAINT achieved the lowest accuracy (51.09%) on CASP11, and 3) a representative protein from CB513 (PDB ID: 154L). These clearly demonstrate the ability of SAINT in predicting reliable structures of proteins.

3.2.1 Interpretability

One notable feature of SAINT is that it can actually visualize and provide insights on how the architecture is making decisions. By leveraging the self-attention alignment matrix, it can be interpreted how different parts of input are dependent on each other

Method	Year	CB513	CASP10	CASP11
SSPro8	2002	63.5	64.9	65.6
RaptorX-SS8	2010	64.9	64.8	65.1
DeepGSN	2014	66.4	-	-
DeepCNF	2016	68.3	71.8	72.3
DCRNN	2016	69.7	**76.9**	73.1
NCCNN	2017	70.3	-	-
CBRNN	2018	70.2	74.5	72.5
CNNH_PSS	2018	70.3	-	-
MUFOLD-SS[1]	2018	70.5	74.1	73.0
SAINT	2019	**70.91**	74.7	**73.3**

[1] Results are obtained from our experiments using the filtered CB6133 as the training dataset.

Table 1: A comparison of the Q8 accuracy (%) obtained by SAINT and other state-of-the-art methods on CB513, CASP10, and CASP11 dataset.

Q8 Label	SAINT	MUFOLD-SS[1]	NCCNN	DCRNN	DeepCNF
H	**0.849**	0.846	0.841	0.832	**0.849**
B	0.475	0.485	**0.676**	0.554	0.433
E	0.748	0.753	**0.767**	0.753	0.748
G	0.420	0.424	0.487	0.429	**0.49**
I	0	0	0	0	0
T	0.568	0.564	**0.577**	0.559	0.53
S	0.539	0.533	**0.548**	0.518	0.487
L	**0.60**	0.596	0.565	0.573	0.571

[1] Results are generated by our experiments.

Table 2: Predictive precision on each of the 8 states obtained by SAINT and other state-of-the-art methods on CB513 dataset.

Q8 Label	SAINT	MUFOLD-SS[1]	NCCNN	DCRNN	DeepCNF
H	0.928	0.926	0.932	**0.933**	0.904
B	**0.056**	0.053	0.041	0.026	0.026
E	**0.851**	0.839	0.821	0.828	0.833
G	**0.364**	0.363	0.285	0.252	0.26
I	0	0	0	0	0
T	**0.550**	0.547	0.524	0.522	0.528
S	**0.271**	0.268	0.24	0.249	0.255
L	0.635	0.644	**0.69**	0.652	0.657

[1] Results are generated by our experiments.

Table 3: Recall on each of the 8 states obtained by SAINT and other state-of-the-art methods on CB513 dataset.

Q8 Label	SAINT	MUFOLD-SS[1]	NCCNN	DCRNN	DeepCNF
H	**0.887**	0.884	0.884	0.879	0.875
B	**0.10**	0.096	0.077	0.05	0.049
E	**0.796**	0.794	0.793	0.789	0.788
G	0.39	**0.391**	0.359	0.317	0.339
I	0	0	0	0	0
T	**0.559**	0.555	0.549	0.54	0.529
S	**0.361**	0.357	0.334	0.336	0.335
L	0.617	0.619	**0.62**	0.61	0.611

[1] Results are generated by our experiments.

Table 4: Comparison of $F1$-score on each of the 8 states obtained by SAINT and other state-of-the-art methods on CB513 dataset.

while generating the output [50, 73–76], and hence it was used to develop interpretable models [77–80]. In SAINT, the attention map can reveal how and to what extent the residues of the proteins interact with each other while forming the secondary structure. SAINT uses the self-attention alignment score matrix of the attention module placed just before the last dense layer (i.e., prior to the final prediction). We show the native structure and the corresponding alignment matrix obtained from SAINT as a gradient image for a sample protein 5MIZ Chain A in CB513 dataset in Fig. 6. We selected a short sequence 5MIZ Chain A (only 21 residues) to easily demonstrate with visualizations how the alignment matrix provides insight about the 3D structure. Higher correlation is represented by deeper hue and lower correlation is represented by lighter hue. In Fig. 6 (b), each residue on the X-axis of the attention map represents a query and each corresponding row represents the results of the query. This alignment matrix suggests that Isoleucine (I) is interacting mostly with one of its distant (in primary sequence) residue Serine (S) and Serine's surrounding residues (see Fig. 6 (c) which shows the interactions of I with other residues). Interestingly, the 3D structure of 5MIZ, obtained from PDB, also shows that I is closer to S (indicated in Fig. 6 (a)). Thus, SAINT provides insights into the roles of the amino acids in a protein's structure and explains how it is making the predictions – laying a firm, broad foundation for interpretable secondary structure predictions.

4 Conclusions

We have presented SAINT, a highly accurate and interpretable method for 8-state SS prediction. We demonstrate for the first time that the self-attention mechanism proposed by Vaswani *et al.* [50] is a valuable tool to apply in the structural analyses of proteins. Another earlier type of attention mechanism proposed by Bahdanau *et al.* [53] coupled with recurrent neural network (RNN) based encoder-decoder architectures achieved state-of-the-art performance on various natural language processing tasks (e.g. neural machine translation [57,81], question answering task [82,83], text summarization [84,85], document classification [86,87], sentiment classification [88,89], etc.). As proteins are also sequences similar to sentences in a language, this type of architecture is expected do well in protein secondary structure prediction as well. However, previous attempts [68] on using attention

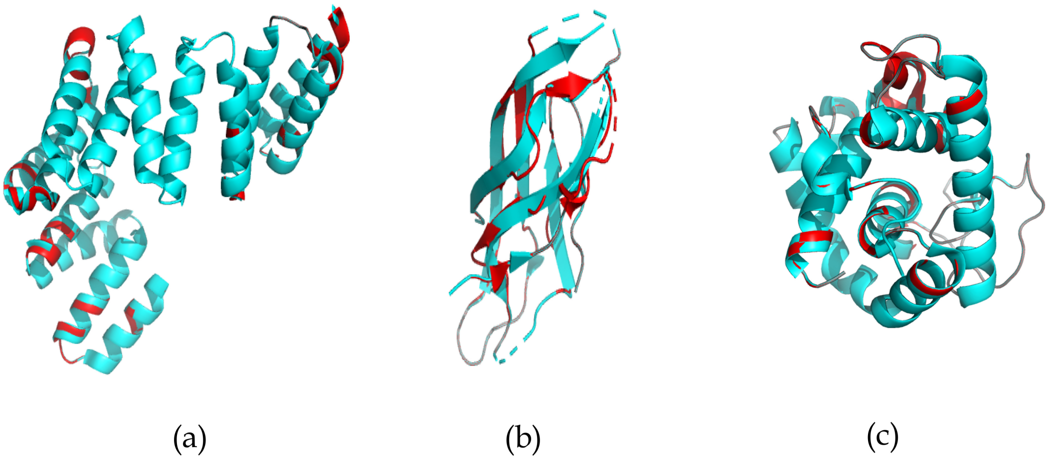

Figure 5: **Superpositions of the structures predicted by SAINT (cyan) with the structures obtained from PDB (red) for three representative proteins from CASP11 and CB513.** (a) T0821-D1:20-274 in CASP11 (PDB ID: 4r7s), (b) T0840-D2:546-661 in CASP11 (PDB ID: 4qt8), and (c) a representative protein from CB513 (PDB ID: 154L). The images are generated in Pymol [72]. Since Pymol does not differentiate between all the 8 distinct states, we translated the 8-state structure to 3-state structure according to the Rost and Sander scheme [10].

with LSTM based encoder-decoder only achieved 68.4% accuracy on CB513 dataset which is significantly worse than the MUFOLD-SS. In this study, we have used the self-attention mechanism in a unique way and proposed a novel attention augmented 3I module (2A3I module) and achieved notable success. We have used the self-attention mechanism to retrieve the relation between vectors that lay far from each other in a sequence. As self-attention mechanism looks at a single vector and measures its similarity or relationship with all other vectors in the same sequence, it does not need to encode all the information in a sequence into a single vector like recurrent neural networks. This reduces the loss of contextual information for long sequences.

SAINT contributes towards simultaneously capturing the short- and long-range dependencies among the amino acid residues. Unlike some of the existing deep learning methods, SAINT can capture the long-range dependencies without using computationally expensive recurrent networks or convolution networks with large window sizes. SAINT was assessed for its performance against the state-of-the-art 8-state SS prediction methods on a collection of widely used benchmark dataset. Our experimental results suggest that SAINT outperforms the best existing methods across all the dataset. In particular, it achieved 70.91% accuracy on the most widely used benchmark dataset CB513, whereas the previous best results achieved by MUFOLD-SS was 70.5%. Given the difficulties and the slow progress rate of the Q8 prediction accuracy, this demonstrated improvement of SAINT over the previous best results is remarkable.

One of the most significant conclusions from the demonstrated experimental results is that appropriate use of self-attention mechanism can significantly boost the performance of deep neural networks and is capable of producing results which rank SAINT at the very top of the current SS prediction methods. Thus, the idea of applying self-attention mech-

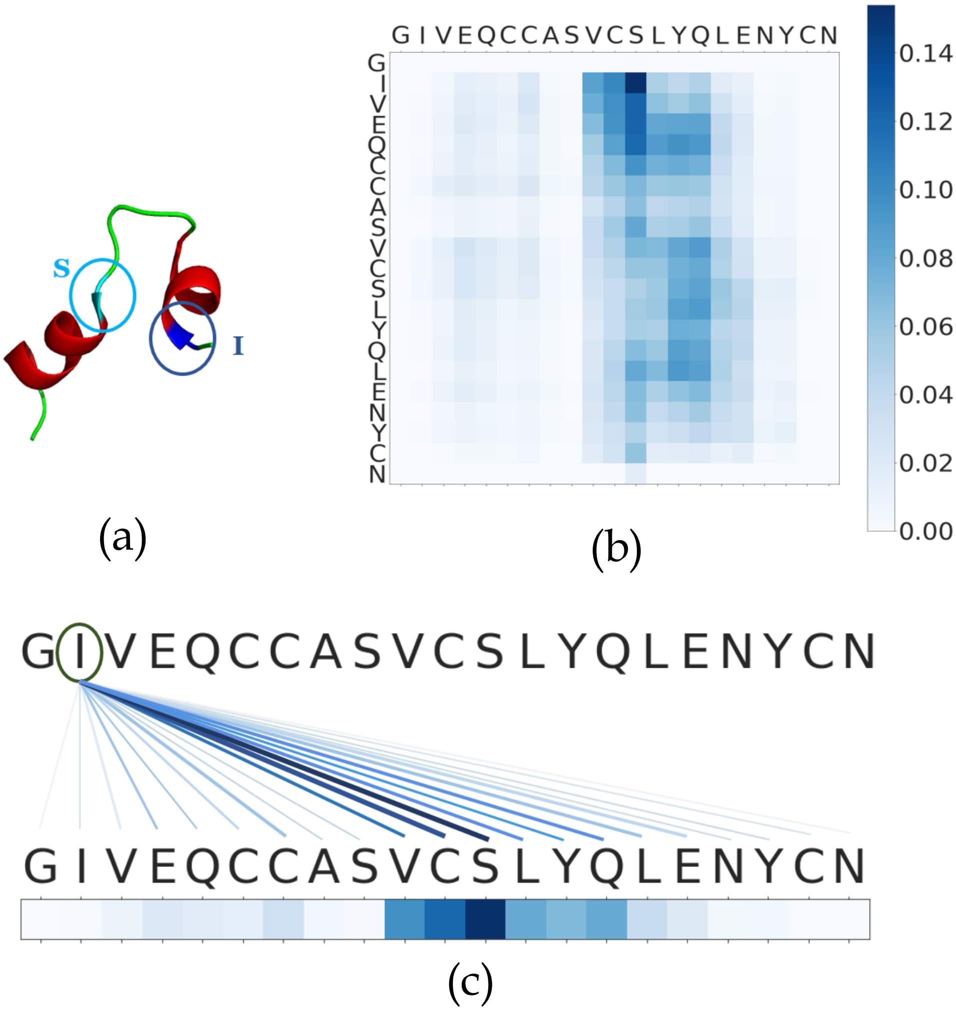

Figure 6: **Demonstration of the interpretability of SAINT using the attention map.** (a) 3D structure of 5MIZ chain A obtained from PDB, (b) self-attention alignment matrix generated by SAINT on 5MIZ, and (c) self-attention alignment scores indicating the interactions of a specific amino acid residue (I) with others in 5MIZ. Deeper hue and thicker lines indicate higher levels of interactions.

anism can be applied to predicting various other protein attributes (e.g., torsion angles, turns, etc. [90]) as well. SAINT also contributes towards generating interpretable deep neural network methods by leveraging the attention map to explain how different residues interact with each other and their roles in the structure of a protein. These insights will be useful to understand the complex relationship between the primary sequence and various structural and functional properties of proteins. Therefore, we believe SAINT represents significant advances, and will be a useful tool for predicting the secondary structures of proteins.

References

[1] Qian Jiang, Xin Jin, Shin-Jye Lee, and Shaowen Yao. Protein secondary structure prediction: A survey of the state of the art. *Journal of Molecular Graphics and Modelling*, 76:379–402, 2017.

[2] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[3] David Baker and Andrej Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[4] Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37:289–316, 2008.

[5] Philip Bradley, Kira MS Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, 2005.

[6] Linus Pauling, Robert B Corey, and Herman R Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

[7] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

[8] Ning Qian and Terrence J Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202(4):865–884, 1988.

[9] L Howard Holley and Martin Karplus. Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences*, 86(1):152–156, 1989.

[10] Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232(2):584–599, 1993.

[11] Markéta J Zvelebil, Geoffrey J Barton, William R Taylor, and Michael JE Sternberg. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, 195(4):957–961, 1987.

[12] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999.

[13] Hyunsoo Kim and Haesun Park. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, 16(8):553–560, 2003.

[14] Jonathan J Ward, Liam J McGuffin, Bernard F. Buxton, and David T. Jones. Secondary structure prediction with support vector machines. *Bioinformatics*, 19(13):1650–1655, 2003.

[15] Jian Guo, Hu Chen, Zhirong Sun, and Yuanlie Lin. A novel method for protein secondary structure prediction using dual-layer svm and profiles. *PROTEINS: Structure, Function, and Bioinformatics*, 54(4):738–743, 2004.

[16] Wei Chu, Zoubin Ghahramani, and David L Wild. A graphical model for protein secondary structure prediction. In *Proceedings of the twenty-first international conference on Machine learning*, page 21. ACM, 2004.

[17] Kiyoshi Asai, Satoru Hayamizu, and Ken'ichi Handa. Prediction of protein secondary structure by the hidden markov model. *Bioinformatics*, 9(2):141–146, 1993.

[18] Zafer Aydin, Yucel Altunbasak, and Mark Borodovsky. Protein secondary structure prediction for a single-sequence using hidden semi-markov models. *BMC Bioinformatics*, 7(1):178, 2006.

[19] Gianluca Pollastri, Darisz Przybylski, Burkhard Rost, and Pierre Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235, 2002.

[20] Jinmiao Chen and Narendra Chaudhari. Cascaded bidirectional recurrent neural networks for protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(4):572–582, 2007.

[21] Claudio Mirabello and Gianluca Pollastri. Porter, paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*, 29(16):2056–2058, 2013.

[22] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*, 33(18):2842–2849, 2017.

[23] Matt Spencer, Jesse Eickholt, and Jianlin Cheng. A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 12(1):103–112, 2015.

[24] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, 19(3):482–494, 2016.

[25] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6:18962, 2016.

[26] Jian Zhou and Olga G. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages 745–753, 2014.

[27] Zhen Li and Yizhou Yu. Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2560–2567. AAAI Press, 2016.

[28] Akosua Busia and Navdeep Jaitly. Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. *arXiv preprint arXiv:1702.03865*, 2017.

[29] Jiyun Zhou, Hongpeng Wang, Zhishan Zhao, Ruifeng Xu, and Qin Lu. Cnnh_pss: protein 8-class secondary structure prediction by convolutional neural network with highway. *BMC Bioinformatics*, 19(4):60, 2018.

[30] Chao Fang, Yi Shang, and Dong Xu. Mufold-ss: New deep inception-inside-inception networks for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 86(5):592–598, 2018.

[31] Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology*, 308(2):397–407, 2001.

[32] Scott C Schmidler, Jun S Liu, and Douglas L Brutlag. Bayesian segmentation of protein secondary structure. *Journal of Computational Biology*, 7(1-2):233–248, 2000.

[33] Laurens Van Der Maaten, Max Welling, and Lawrence Saul. Hidden-unit conditional random fields. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 479–488, 2011.

[34] Pierre Baldi, Søren Brunak, Paolo Frasconi, Giovanni Soda, and Gianluca Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

[35] Christophe N Magnan and Pierre Baldi. Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 2014.

[36] Zhiyong Wang, Feng Zhao, Jian Peng, and Jinbo Xu. Protein 8-class secondary structure prediction using conditional neural fields. In *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 109–114. IEEE, 2010.

[37] James A Cuff and Geoffrey J Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[38] Yanbu Guo, Weihua Li, Bingyi Wang, Huiqing Liu, and Dongming Zhou. Deepaclstm: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction. *BMC Bioinformatics*, 20(1):341, 2019.

[39] Buzhong Zhang, Jinyan Li, and Qiang Lü. Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC bioinformatics*, 19(1):293, 2018.

[40] Jack Hanson, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. *Bioinformatics*, 33(5):685–692, 2016.

[41] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2, 2012.

[42] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[43] Sekitoshi Kanai, Yasuhiro Fujiwara, and Sotetsu Iwamura. Preventing gradient explosions in gated recurrent units. In *Advances in neural information processing systems*, pages 435–444, 2017.

[44] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2016.

[45] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks and Learning Systems*, 5(2):157–166, 1994.

[46] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[47] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M Rao, et al. Interpretability of deep learning models: a survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, pages 1–6. IEEE, 2017.

[48] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[49] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[51] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[52] UniProt Consortium. The universal protein resource (uniprot). *Nucleic Acids Research*, 36(suppl_1):D190–D195, 2007.

[53] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[54] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[55] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016.

[56] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[57] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[58] Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*, 2016.

[59] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 448–456. JMLR. org, 2015.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[61] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *Stat*, 1050:21, 2018.

[62] Søren Kaae Sønderby and Ole Winther. Protein secondary structure prediction with long short term memory networks. *arXiv preprint arXiv:1412.7828*, 2014.

[63] Yanbu Guo, Bingyi Wang, Weihua Li, and Bei Yang. Protein secondary structure prediction improved by recurrent neural networks integrated with two-dimensional convolutional neural networks. *Journal of Bioinformatics and Computational Biology*, 16(05):1850021, 2018.

[64] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[65] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[66] Guoli Wang and Roland L Dunbrack Jr. Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[67] Patrice Koehl and Michael Levitt. A brighter future for protein structure prediction. *Nature Structural Biology*, 6(2):108, 1999.

[68] Iddo Drori, Isht Dwivedi, Pranav Shrestha, Jeffrey Wan, Yueqi Wang, Yunchu He, Anthony Mazza, Hugh Krogh-Freeman, Dimitri Leggas, Kendal Sandridge, et al. High quality prediction of protein q8 secondary structure by diverse neural network architectures. *arXiv preprint arXiv:1811.07143*, 2018.

[69] Yutaka Sasaki et al. The truth of the f-measure. *Teach Tutor Mater*, 1(5):1–5, 2007.

[70] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[71] Jan Ludwiczak, Aleksander Winski, Antonio Marinho da Silva Neto, Krzysztof Szczepaniak, Vikram Alva, and Stanislaw Dunin-Horkawicz. Pipred–a deep-learning method for prediction of π-helices in protein sequences. *Scientific Reports*, 9(1):6888, 2019.

[72] Warren L DeLano et al. Pymol: An open-source molecular graphics tool. *CCP4 Newsletter on Protein Crystallography*, 40(1):82–92, 2002.

[73] Reza Ghaeini, Xiaoli Fern, and Prasad Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4952–4957, 2018.

[74] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126, 2017.

[75] Tamer Alkhouli and Hermann Ney. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, 2017.

[76] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[77] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 2019.

[78] Chen Chen, Jie Hou, Xiaowen Shi, Hua Yang, James A Birchler, and Jianlin Cheng. Interpretable attention model in transcription factor binding site prediction with deep neural networks. *bioRxiv*, page 648691, 2019.

[79] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.

[80] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

[81] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, 2016.

[82] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406, 2016.

[83] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[84] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

[85] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[86] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[87] Nikolaos Pappas and Andrei Popescu-Belis. Multilingual hierarchical attention networks for document classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1015–1025, 2017.

[88] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3298–3307, 2016.

[89] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[90] Chao Fang, Zhaoyu Li, Dong Xu, and Yi Shang. Mufold-ssw: A new web server for predicting protein secondary structures, torsion angles, and turns. *Bioinformatics*, 2019.