# ENCODING STRUCTURE: GENERATING A STRUCTURE-AWARE EMBEDDING FOR PROTEIN SEQUENCES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Machine learning models, such as AlphaFold, have recently demonstrated remarkable accuracy in predicting the structures of protein sequences. This capability enables their use as oracles for providing structure-based information to aid other learning tasks. In this study, we investigate the use of deep learning embeddings and explore the feasibility of developing a structure-aware protein sequence embedding. To achieve this, we use a neural network model (S4PRED), which accurately predicts protein secondary structures directly from single sequences, as an oracle to generate structure-aware embeddings using an autoencoder. We then compare this approach to purely sequence-based embeddings in a Protein-Protein Interaction (PPI)-prediction task. Our findings highlight the potential advantages of employing structure-aware embeddings and provide grounds for future research directions.

## 1 INTRODUCTION

Deep learning models are extensively used to predict and generate information about proteins. To do this, these models use an embedding, which converts the protein sequence into a numerical form that can be used for training. The most classical of these embeddings are one-hot encoding, where each of the 20 amino acids are encoded as a 20-dimensional vector with a single bit set to 1, BLOSUM62, based on the 20-dimensional BLOSUM Matrix which is derived from clustering of aligned segments in blocks (Steven Henikoff, 1992), and VHSE8, an 8-dimensional matrix that contains hydrophobic, steric, and electronic properties (Hu Mei, 2005). While all of these embedding matrices are useful, none of them provide any higher-level structural information about a protein. This means that for deep learning tasks in which protein structure is important, the network needs to first learn about the structure of the protein before even starting to become more specialized. In this work, we begin exploring this issue using an autoencoder network to generate a structure-aware embeddings for protein sequences. We do so by using a relatively simple (compared to AlphaFold) secondary structure-prediction model called S4PRED, which is able to predict the protein secondary structures directly from sequence, as our oracle (Lewis Moffat, 2021). We show that this concept is feasible and worth pursuing by comparing our structure-aware embedding to the classical embeddings on a deep learning task in PPI prediction, a task that relies on structural knowledge of proteins (ElAbd H., 2020).

## 2 METHODS

We trained two types of embeddings: one structure-aware, and one purely sequence aware. The dataset used to train the autoencoder consisted of 242,971 proteins of length 600 amino acids or smaller from the Protein Database (PDB); see Appendix section 5.2 for full dataset details.

### 2.1 STRUCTURE AWARE EMBEDDING

The structure-aware embedding was trained using an autoencoder model in the Pytorch ML framework. The encoder part of the model is 6 fully connected linear layers with LeakyReLU's between each of its layers. The embedding size of the encoder was somewhat arbitrarily set to 5000 because it is roughly less than half the size of the one-hot encoding size of 12000. The decoder part of the

Table 1: Comparison of Encodings on PPI Prediction

| Encoding | Train Acc | Train AUC | Test Acc | Test AUC | Flattened Size |
|---|---|---|---|---|---|
| One Hot (20 dimensions) | 0.86 | 0.93 | 0.83 | 0.88 | 12000 |
| BLOSUM (20 dimensions) | 0.99 | 0.99 | 0.80 | 0.86 | 12000 |
| VHSE8 (8 dimensions) | 0.81 | 0.89 | 0.80 | 0.86 | 4800 |
| Structure-Aware | 0.66 | 0.77 | 0.65 | 0.74 | 5000 |
| Pure Sequence | 0.85 | 0.91 | 0.84 | 0.81 | 5000 |
| Pure Sequence | 0.82 | 0.85 | 0.76 | 0.83 | 500 |

model is also 6 fully connected linear layers with LeakyReLU's between each of the layers and a Sigmoid activation function on the final layer. For the loss function, we used a slightly modified version of S4PRED to predict the secondary structures for both the input and output protein sequence. Both sequences, returned as one-hot encoded sequences of $[C|E|H] * n$, were then passed into the mean squared error function to compute the loss. The gradient was thus computed and back-propagated through the S4PRED model, the decoder, and the encoder. We froze the layers of the S4PRED model so as to not alter the state of the oracle. Stochastic Gradient Descent was used as the Optimizer. Training of the autoencoder was carried out using a bootstrapping inspired technique using batches of 100000 sequences drawn uniformly at random. We trained this model using 6 GPU's. Refer to the diagram in the Appendix 5.1 for a visualization of this model structure.

### 2.2 Pure Sequence Embedding

The pure sequence-based embedding was generated using the same autoencoder model used for the structure-aware embedding. The difference being that instead of calculating the loss based on the S4PRED model output, we used a binary cross entropy function on both the input and output one-hot encoding. Using this model, we created embeddings of size 5000 and 500.

## 3 Results

Table 1 shows results of the PPI prediction task for all of the encodings that were benchmarked. Acc is the accuracy of the PPI model and AUC is the area under the ROC curve of the model. The flattened size of each encoding is the number of values needed to represent the protein sequence. Refer to the Appendix section 6.2 for more details on how these encodings were benchmarked. Overall, the data indicates that this apporach to structure-aware encoding was not able to provide accuracy comparable to classical encodings. Interestingly, the pure sequence-based encodings performed similar to, or in some cases better than, all of the classical encodings in both accuracy, AUC, and memory (embedded dimension size). We note that re-training the structure-aware model using cross-entropy loss, as opposed to mean squared error, could boost model performance.

## 4 Discussion

The modest accuracy achieved by the structure-aware encoding is highly promising, especially considering that the model utilized only the secondary structure of the protein for PPI prediction. This underscores the potential for improving the accuracy of structure-aware encoding through a hybrid approach, combining sequence and structural information. By incorporating oracles like S4PRED or AlphaFold, we could train the encoding to learn from both sequence and structural features, thus enhancing the model's accuracy and performance. Indeed, the hybrid approach is likely to be successful, given the clear advantage of a pure sequence encoding compared to other encodings in terms of accuracy, AUC, and memory footprint. Alternatively, task-specific structure-aware encodings could be generated to optimize the encoding for specific tasks. In conclusion, this work on structure-aware encoding serves as a solid foundation for future research and development in this area, with promising avenues for enhancing the accuracy of the encoding through a hybrid approach and task-specific design.

### URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

## REFERENCES

Hoarfrost A. ElAbd H., Bromberg Y. Amino acid encoding for deep learning applications. *BMC Bioinformatics*, 21:235, 2020.

Yuan Zhou Shengshi Z Li Hu Mei, Zhi H Liao. A new set of amino acid descriptors and its application in peptide qsars. *Biopolymers*, 80:775–786, 2005.

David T Jones Lewis Moffat. Increasing the accuracy of single sequence prediction methods using a deep semi-supervised learning framework. *Bioinformatics*, 37:3744–3751, 2021.

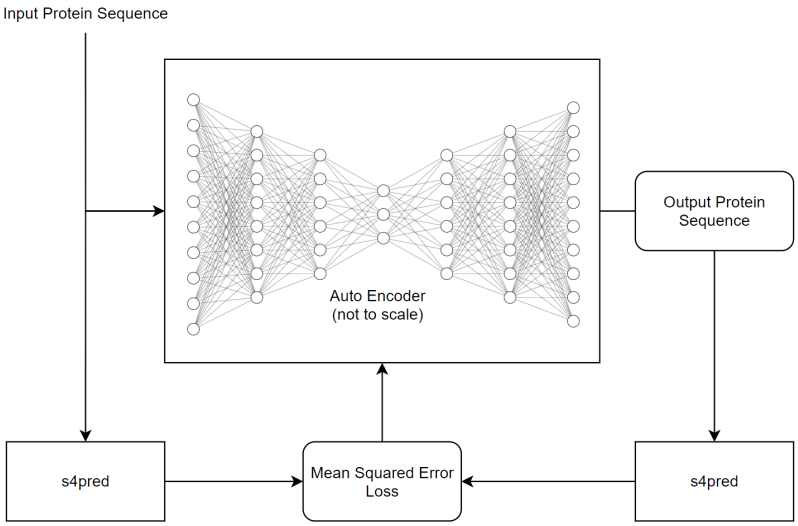
Aly A Khan Jinbo Xu Somaye Hashemifar, Behnam Neyshabur. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34:i802–i820, 2018.

Jorja Henikoff Steven Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89:10915–10919, 1992.

## 5 APPENDIX

The following sections contain more details on how the encodings were trained and benchmarked.

### 5.1 DIAGRAM OF STRUCTURE-AWARE ENCODER




### 5.2 DATASET

The dataset used to train the autoencoder were sequences from the Protein Database (PDB). We chose this dataset because it contained proteins with defined secondary structures, which is not a guarantee for proteins in sequence-only databases such as Uniref. We then filtered out any proteins with sequences of length 600 or higher and padded every sequence with zeroes to a length of 600. Finally, we filtered out all of the sequences with unknown regions (determined by an $X$ in the sequence). As a result, we ended up with 242,971 proteins in the dataset.

### 5.3 BENCHMARKING

Using Pytorch, we re-implemented an architecture inspired by (Somaye Hashemifar, 2018) for identifying PPIs similar to the module used by (ElAbd H., 2020). This model is composed of two main parts. The first part is a convolution neural network consisting of 4 modules with a convolutional layer, followed by a rectified linear unit (ReLU), a batch normalization layer, and an average pooling layer in all except the final layer which using a pooling layer for the Global average. Every forward pass takes a pair of proteins and generates a pair of vectors that represent each of these vectors using the same weights. The second part of this mode is a feed-forward multilayer perceptron which receives the pair of vectors and outputs the probability that the pair of proteins interact, thereby predicting PPI. Like (ElAbd H., 2020), we also make the optimization of using 4 modules instead of 5 to speed up training time and allow for faster experimentation. Note that this model uses an Adam optimizer and a sigmoid function for the activation function for each of the prediction layers. Finally, binary cross-entropy was used as a loss function.