Bobby Becker, Rafael Djamous, Marco Carbullido
Intro to Deep Learning
12/12/2024

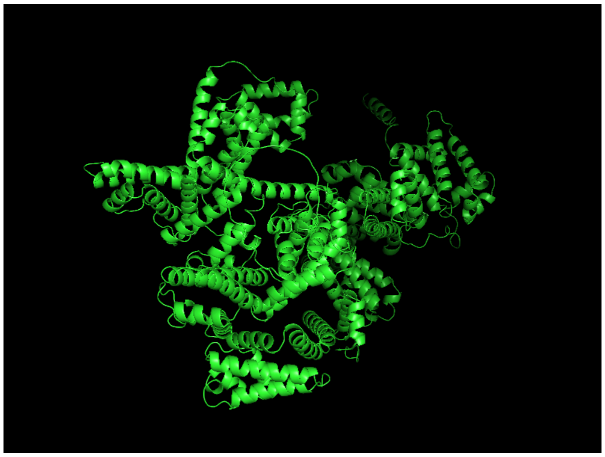
Predicting B-Factor of Amino Acids through Deep Learning

Abstract

In this project, we aimed to predict the B-Factor, a measure of atomic displacement, of amino acids within a protein sequence using deep learning approaches. We tested various parameters and architectures of linear models, recurrent neural networks, long-short term models, and transformers in order to systematically measure the performance of these approaches for this prediction task. For each model, we measured the performance of models using amino acid sequences raw, as well as the embeddings of sequence positions using ProBERT. We found that, across all models, ProBERT embeddings produce a significant improvement in their predictive power. We then scaled up our experiments, implementing a customized LSTM and Transformer model to predict B-Factors on a training dataset of 60,000 protein sequences. On this larger scale, we were able to match state-of-the-art performance with an 81% Pearson Correlation Coefficient, and found that positional embeddings allowed our models to make predictions with higher accuracy and more efficiency.

Background & Intro

In recent years, there has been massive progress made in deep learning approaches applied to protein sequences. Most notably, Google Deepmind has shown the ability to render the 3D structure of proteins via sequences alone, with an accuracy exceeding physics based approaches:

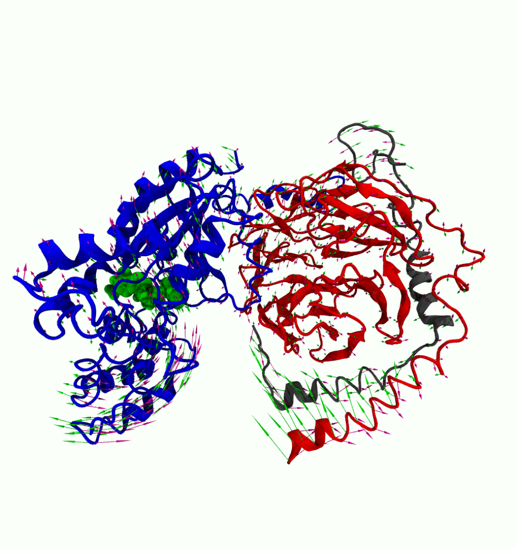


A Protein in our dataset, 2BXO, rendered in Alphafold3

This research has an incredible potential to advance medical science—revealing how protein mutations affect structure helps us understand disease mechanisms, predicting how proteins will interact with potential drug molecules, understanding how protein misfolding diseases like Alzheimer's and Parkison's works, among many other use cases.

Furthermore, Generative AI approaches have shown immense promise. Start-ups and research labs are investing capital and talent toward GPT models of protein sequences. Similar to LLMs, these models are trained on a sequence of data to be able to iteratively predict the next token (amino acid). Many researchers believe that this technology could be the key to designing customized proteins for drug discovery or gene editing.

One important piece of this analysis—outside of merely predicting the positions of amino acids in a protein structure—is predicting the amount in which different amino acids move. The protein structure which AlphaFold generates shows the 'average' position that amino acids are most likely to take, but in reality, proteins move and vary from this position:



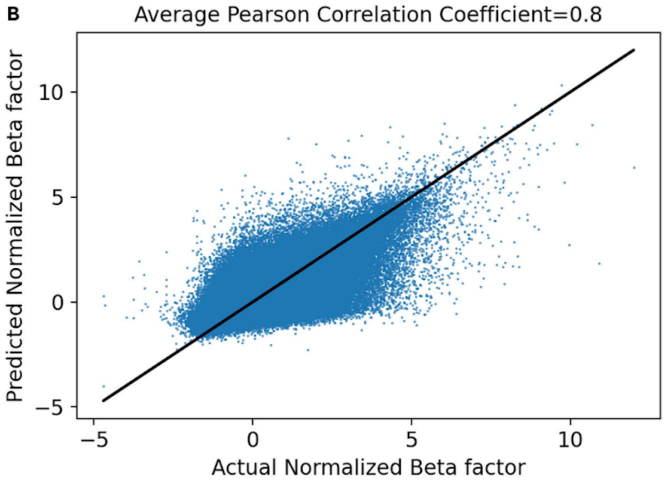
Green and red arrows show common ways in which parts of the protein fluctuate in position over time

Some approximate measures for this are Radius of Gyration (ROG) or Root Mean Square Fluctuation (RMSF). For our purposes, we focused on B-Factor, a measure of amino acid displacement empirically obtained through X-ray crystallography. This type of measurement is expensive, which is why researchers would benefit from the ability to use existing measurements to predict B-Factors of proteins to give an approximate measurement of B-Factors.

The ‘B-Factor’ is, most precisely, a measure for each individual atom’s mobility within an amino acid molecule. However, in practice, the base carbon atom is most often used to represent the mobility of the entire amino acid, as it has the most effect. Researchers also sometimes use the backbone atoms (N, C-alpha, C, O) which better represent peptide backbone mobility or the average of all atoms in the residue to obtain an even more detailed picture. Our data contained all three of these measures, and we were able to achieve comparable predictions across all of them.

Methods

Neighboring atoms have a large influence on each other’s B-Factors, which makes this type of measurement dependent and derivable from sequence data. Because of this, LSTMs have previously shown to achieve SOTA performance in B-Factor prediction due to their ability to better handle vanishing gradients faced by RNNs for long sequences. To benchmark our evaluation, we reimplemented this model and validated its performance:



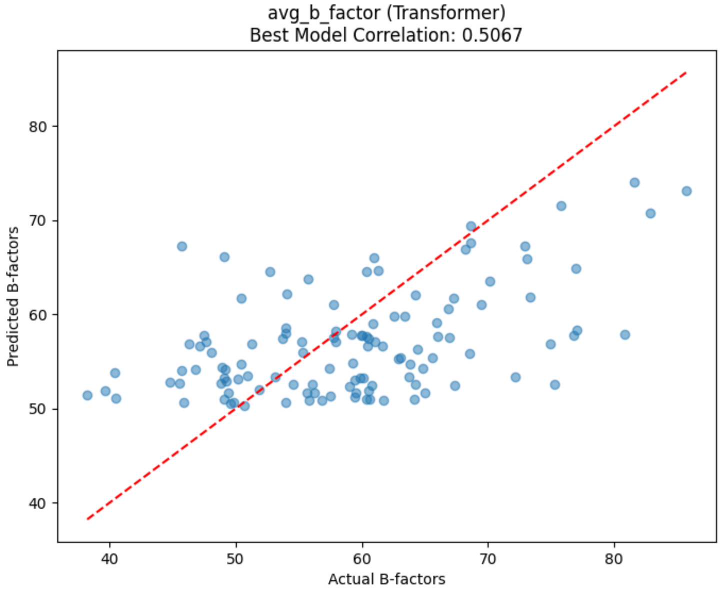
(As is shown here, the performance of these tasks is generally represented by the correlation between predicted B-Factor and actual B-Factor.)

Although the LSTM architecture performs better for longer sequences than the vanilla RNN, it struggles with long sequences. In our most successful approach, each sequence of input tokens (i.e., amino acid embeddings) is projected into a set of keys, queries and values through learned linear transformations.

In our highest performing trial, they were projected to a dimension of 128. While the value vectors represent the information contributed by each token, the query and key vectors are used to weight that contribution. The sum of the value vectors weighted by the dot product of a given token's query vector and the key vectors imbues that token with contextual information from the entire sequence. The dot products are generally normalized by dividing by the square root of the key dimension before being passed to the softmax function to ensure they sum to one. Previous research has shown that this approach significantly improves the ability to model intricate biological sequences due to its capacity to learn from both local and global contexts.

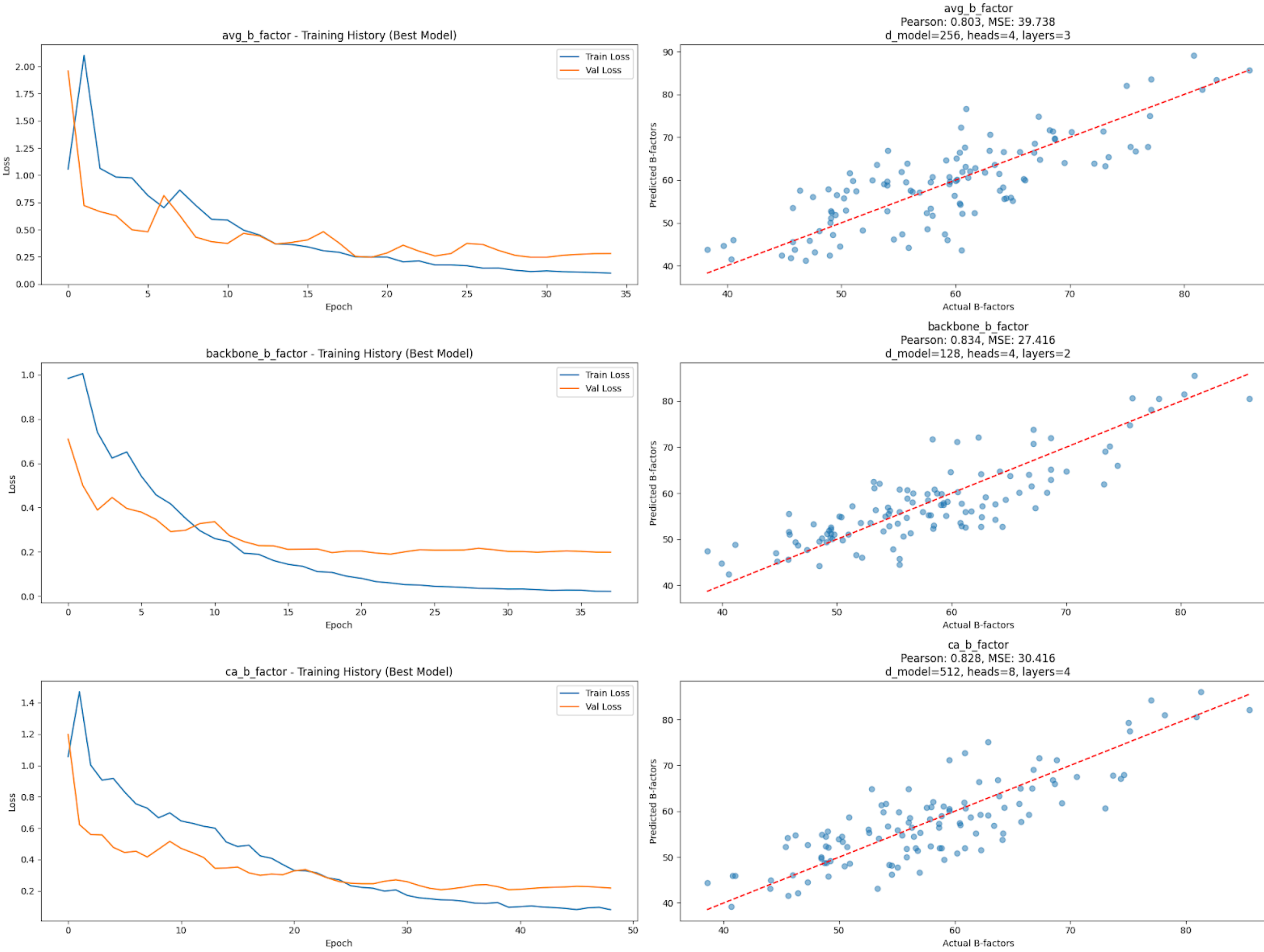
Experiments 1: Small-Subset

On a small subset of our data, we systematically tested linear models, RNNs, LSTMs, and Transformers to further evaluate their performance more quickly and through less expensive computational resources. On this smaller scale, we found that Transformers were able the best performance, originally with a Pearson Correlation Coefficient of 0.50:

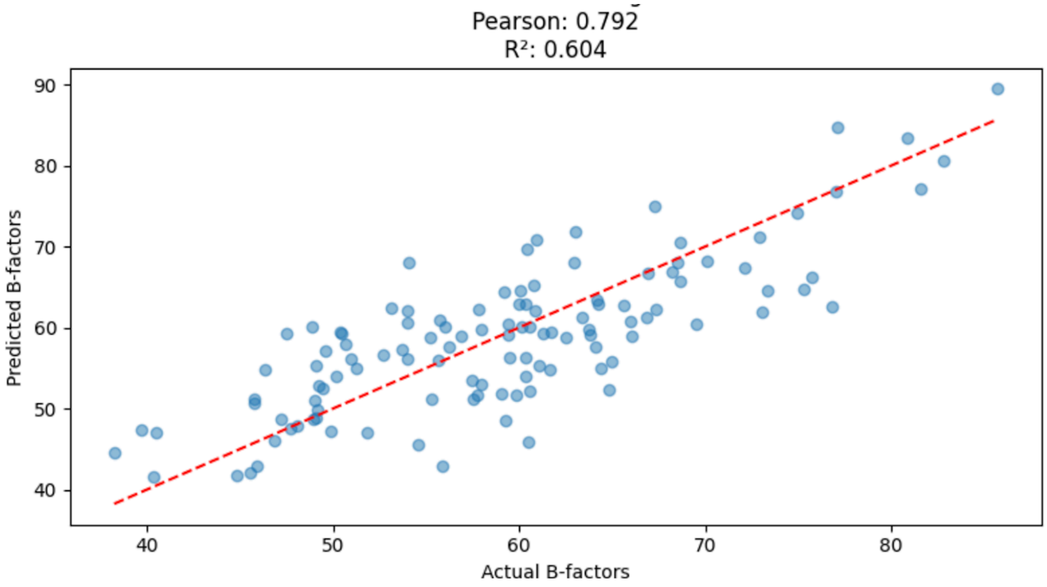


Linear models, RNNs, and LSTMs all achieved a Pearson Correlation Coefficient of around 0.40. While this was all substantially lower than SOTA performances (which was to be expected on the extremely small subset of data which we were working with), we did achieve a similar hierarchy of performance across the model architectures to what was reported in the research. For each model, we iteratively tested ten permutations regarding its configurations & parameters to find the best performance of each one in order to ensure we were getting an accurate picture of their capabilities in this task.

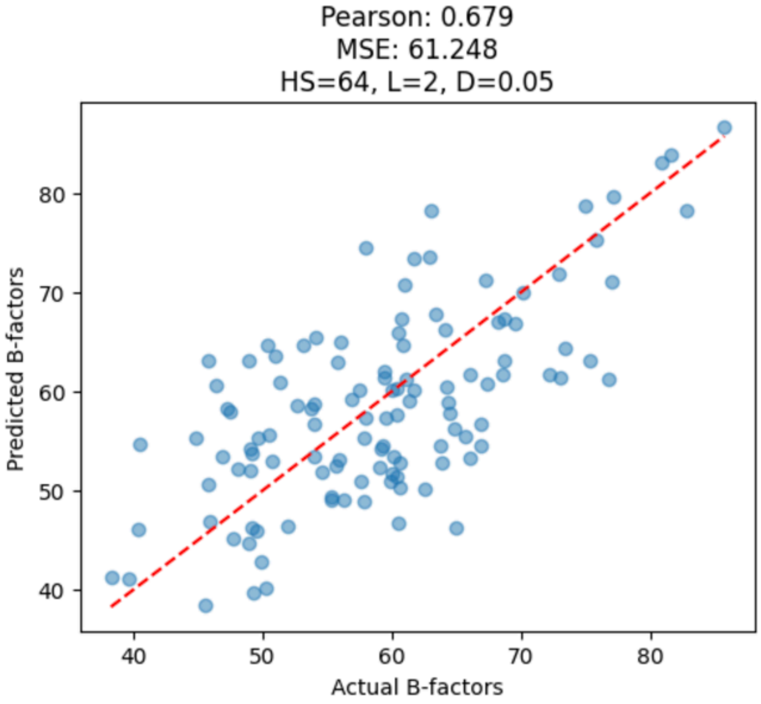
We then created embedding representations of our data through a BERT model designed to handle protein sequences. After modifying our models to take in these embedding representations as inputs, we tested the models again. Here, we found, once again, that Transformers achieved the best performance across all measures of B-Factor:



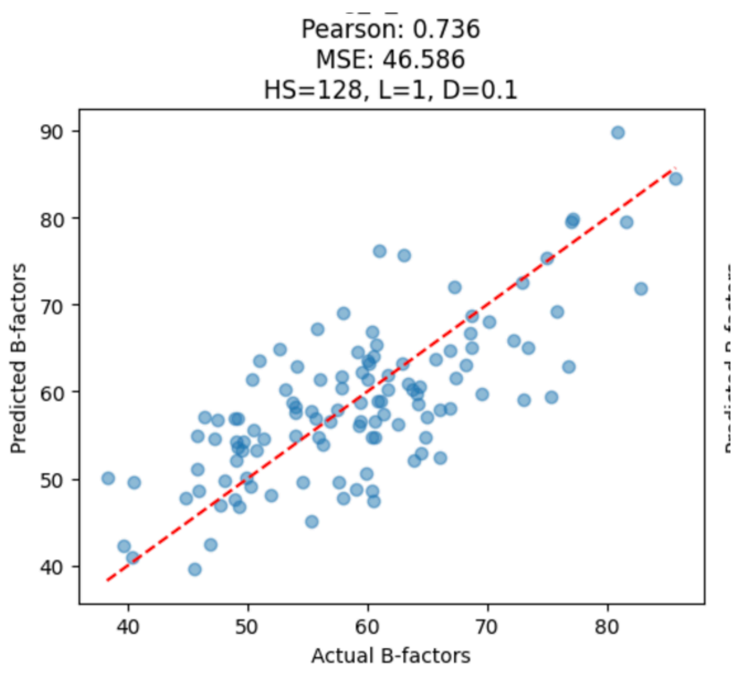
Interestingly, we also found that linear models achieved better results than RNNs and LSTMs, potentially suggesting that these models struggled in regression tasks with embeddings:



Best Linear Model Performance: Ridge Regression



Best RNN Performance

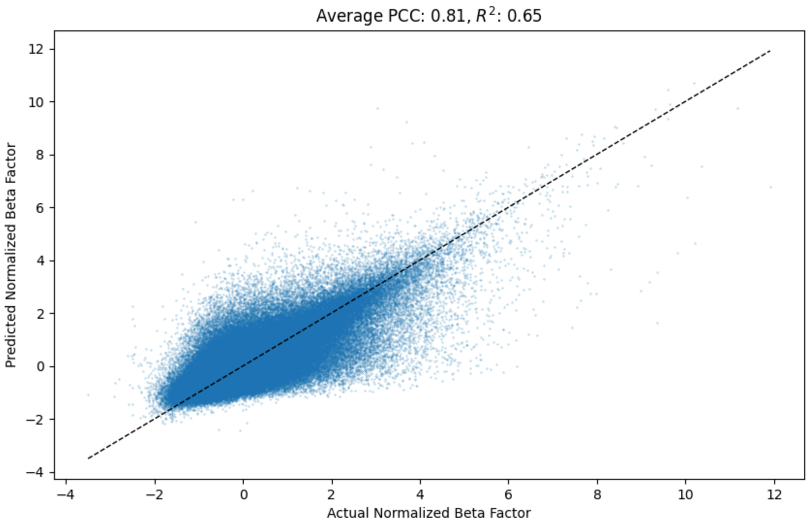


Best LSTM Performance

Regardless, across all models, we found substantial evidence for ProBERT embedding's ability to increase the accuracy of predictions, especially for small datasets. Since our ProBERT model was trained on a much larger dataset of protein sequences, it appeared to have the ability to represent sequences with the necessary semantic context for regressive models to make accurate predictions with.

Experiment 2: 60,000 Protein Sequences

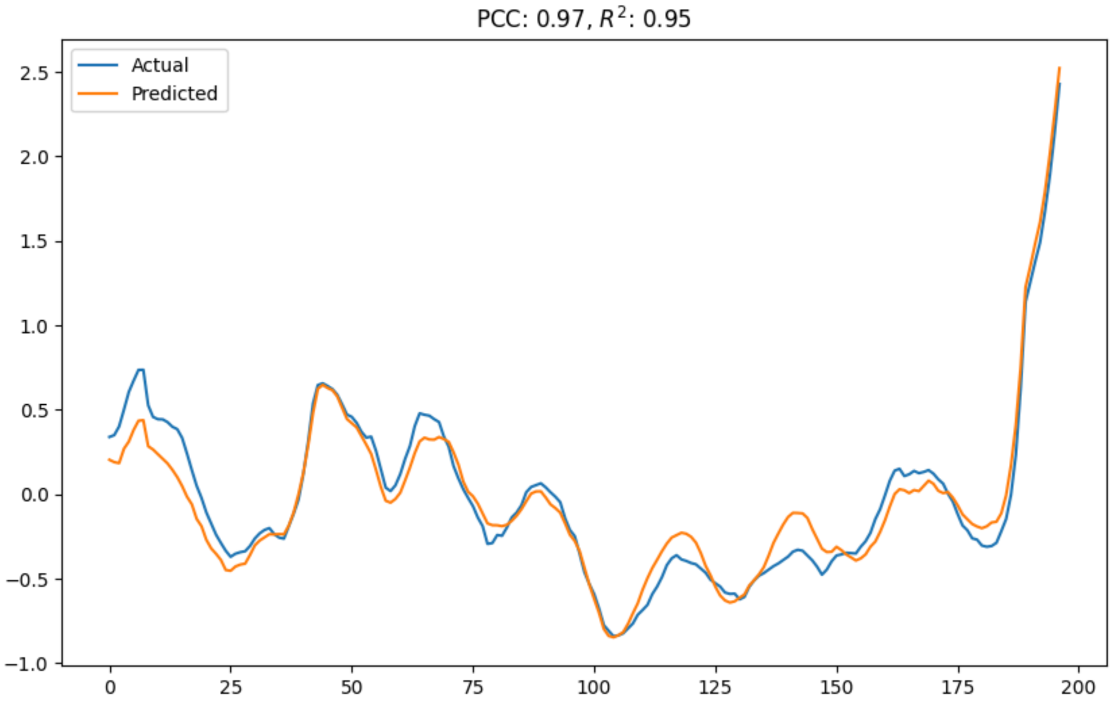
Our second experiment involved implementing a customized transformer to run on a much larger amount of data. We used every protein in the Protein Data Bank with less than 500 amino acids, which gave us a total of 60,000 sequences. Our best model achieved a performance of 0.81, better than the recent LSTM SOTA model set last Spring and on par with the most recent SOTA model achieved in July:



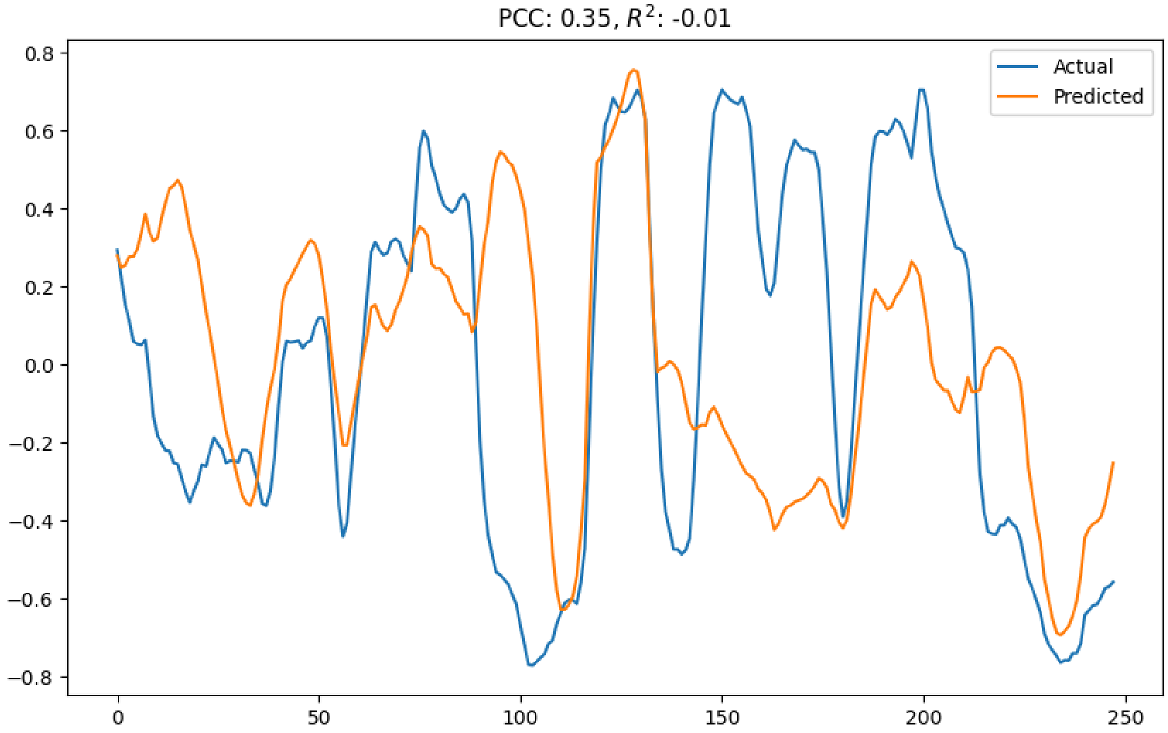
To learn effective representations of the amino acid sequences and their associated structural information, we implemented an auto-encoding architecture as a preliminary step before passing the data to the transformer model. The autoencoder consisted of an encoder and a decoder, enabling it to learn compressed representations of the input data by minimizing the reconstruction error between the original and reconstructed sequences. The encoder works by processing the input sequences of 500 x 28 and then outputting a latent representation that captures essential features while retaining the necessary details to reconstruct the input. It then reconstructs the original input from this latent representation, further refining the learned features through backpropagation. By training the autoencoder on the input sequences, we ensure that the encoded representation captures relevant structural and contextual information, facilitating the transformer model in learning richer relationships within the data. This dual-architecture approach harnesses the strengths of both self-attention and auto-encoding, yielding a robust representation conducive for downstream tasks:

Discussion

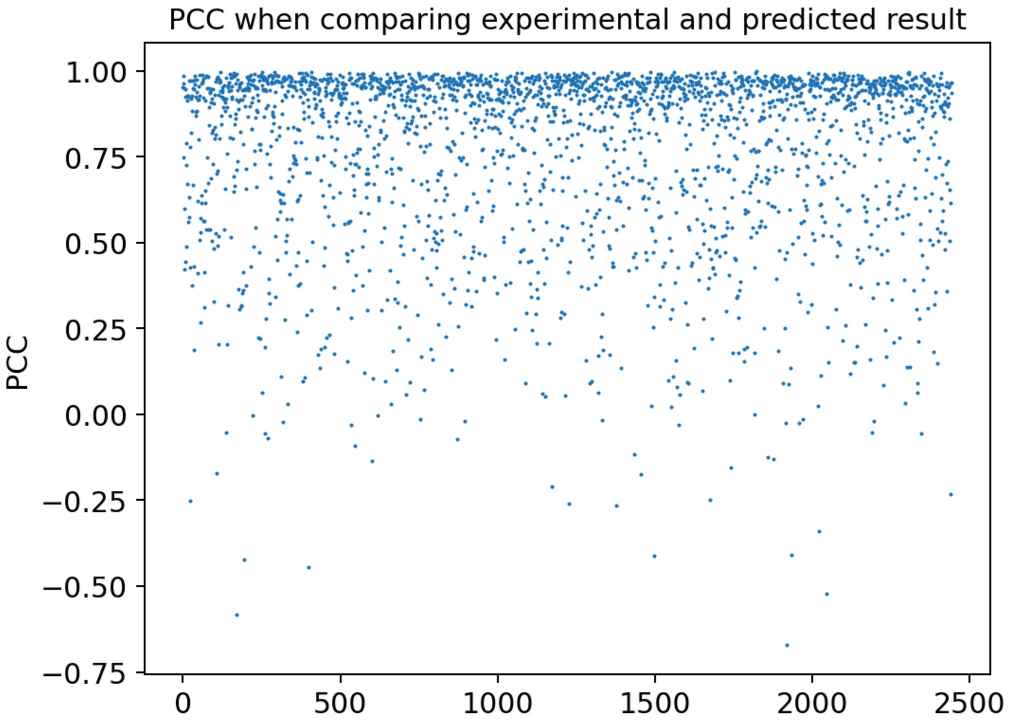
Most notably, we found that our Transformer model's performance varied immensely based on the protein it was predicting:



Our model's high performance of prediction on a protein



Our model's poor performance on different performance



This data indicates that our accuracy is weighed down by some percentage of proteins in which performance is poor, while in many others the performance is close to perfect

This observation leads us to a few promising future areas of research: first, since our model performs near perfect results on many proteins, we could train a specialized model made to predict proteins in which its performance is high. Furthermore, if there is some commonality between the proteins in which performance is poor—such as inaccurate data or data structured in a different way which is throwing off the model—this could be observed and addressed. Further research is necessary to see if these hypotheses could be true.

Moreover, this also leads us to believe that a mixture-of-experts (or an analogous approach) could yield higher accuracy in this domain. If we could design models to make predictions with high accuracy on certain subsets of proteins, we could take in an input sequence, select a model that would yield the highest performance, and then make a prediction with the selected model. We could find which proteins are most similar to an input protein through a cosine similarity of their embeddings, and then feed the input into a model which makes the most optimal prediction for that protein.

Conclusion

We demonstrate significant progress in predicting protein B-Factors using deep learning approaches through the implementation of transformer architectures and protein-specific embeddings. Our experiments, conducted across both small-scale and large-scale datasets confirmed that transformer models consistently outperform traditional architectures like RNNs and LSTMs in this prediction task. We also demonstrated the importance of the encoding architecture in improving prediction accuracy.

Our ultimate model achieved a Pearson Correlation Coefficient of 0.81%, on par with SOTA models. While some proteins showed near-perfect prediction accuracy, others demonstrated significantly poorer results, suggesting that protein-specific characteristics play an important role in regressive predictive tasks in this domain. This leads us to hypothesize how specialized models for specific protein types and mixture-of-experts approaches could optimize predictive accuracy.

References

Brandes, N., Ofer, D., et al. (2022). ProteinBERT: A universal deep-learning model of protein sequence and function. Bioinformatics, 38(8), 2102-2110.

Chandra, A., Tünnermann, L., et al. (2023). Transformer-based deep learning for predicting protein properties in the life sciences. eLife, 12, e82819.

Jumper, J., Evans, R., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.

Pandey, A., Liu, E., et al. (2023). B-factor prediction in proteins using a sequence-based deep learning model. Patterns, 100805.

Smyth, M. S., Martin, J. H. J. (2000). X Ray crystallography. Journal of Clinical Pathology: Molecular Pathology, 53(1), 8-14.

Xu, G., Yang, Y., et al. (2024). OPUS-BFactor: Predicting protein B-factor with sequence and structure information.