



Primary sequence based protein–protein interaction binder generation with transformers

Junzheng Wu¹ · Eric Paquet^{1,2} · Herna L. Viktor¹ · Wojtek Michalowski³

Received: 28 September 2022 / Accepted: 3 September 2023
© The Author(s) 2023

Abstract

The design of binder proteins for specific target proteins using deep learning is a challenging task that has a wide range of applications in both designing therapeutic antibodies and creating new drugs. Machine learning-based solutions, as opposed to laboratory design, streamline the design process and enable the design of new proteins that may be required to address new and orphan diseases. Most techniques proposed in the literature necessitate either domain knowledge or some appraisal of the target protein's 3-D structure. This paper proposes an approach for designing binder proteins based solely on the amino acid sequence of the target protein and without recourse to domain knowledge or structural information. The sequences of the binders are generated with two new transformers, namely the AppendFormer and MergeFormer architectures. Because, in general, there is more than one binder for a given target protein, these transformers employ a binding score and a prior on the sequence of the binder to obtain a unique targeted solution. Our experimental evaluation confirms the strengths of this novel approach. The performance of the models was determined with 5-fold cross-validation and clearly indicates that our architectures lead to highly accurate results. In addition, scores of up to 0.98 were achieved in terms of Needleman-Wunsch and Smith-Waterman similarity metrics, which indicates that our solutions significantly outperform a seq2seq baseline model.

Keywords Protein–protein interaction · Deep learning · Transformer architectures · Protein design

Introduction

Protein–protein interactions (PPIs) are physical contacts of high specificity established between two or more proteins

Junzheng Wu, Eric Paquet, Herna L. Viktor and Wojtek Michalowski have contributed equally to this work.

✉ Herna L. Viktor
hviktor@uottawa.ca

Junzheng Wu
jwu220@uottawa.ca

Eric Paquet
eric.paquet@nrc-cnrc.gc.ca

Wojtek Michalowski
wojtek@telfer.uottawa.ca

- ¹ School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, ON K1N 6N5, Canada
- ² National Research Council, 1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
- ³ Telfer School of Management, University of Ottawa, 55 Laurier Avenue East, Ottawa, ON K1N 6N5, Canada

as a result of biochemical events steered by interactions that include electrostatic forces, hydrogen bonding, and the hydrophobic effect. These physical contacts are molecular associations between chains that occur in a cell or living organism in a specific biomolecular context. PPIs play a key role in biological processes [1, 2]. The knowledge of PPIs allows the deciphering of diseases and design of new drugs and therapeutic antibodies [3]. A typical PPI involves an interaction between a target protein and one or more binding proteins. For instance, the target protein may be part of a pathogen, such as the SARS-CoV-2 (Severe Acute Respiratory Syndrome-CoV-2) spike glycoprotein, while the binder may be a neutralizing antibody, such as the S309 antibody Fab fragment [4]. The resulting complex is illustrated in Fig. 1, where the antibody (binder) is depicted in green, while the spike (target protein) is shown in pink.

Proteins are large biomolecules that comprise one or more long chains of amino acid residues. They perform a vast array of functions within organisms, including catalyzing metabolic reactions, replicating DNA, responding to stimuli, providing structure to cells and organisms, and transporting molecules from one location to another. Proteins differ from

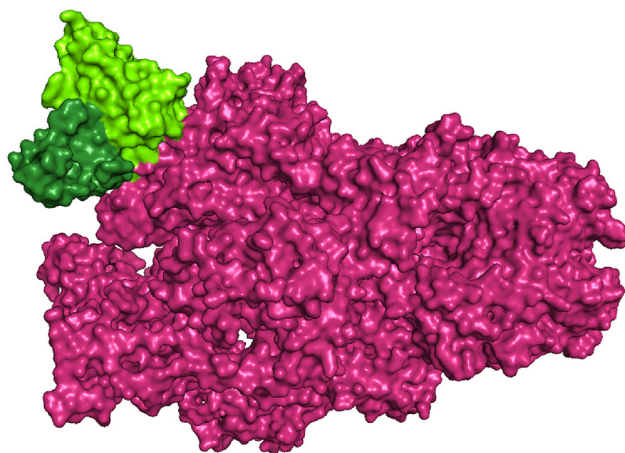


Fig. 1 Structure of the SARS-CoV-2 spike glycoprotein (in pink) in complex with the S309 neutralizing antibody Fab fragment (in green)

one another primarily in their amino acid sequence, which is dictated by the nucleotide sequence of the corresponding genes and usually causes protein folding into a specific 3-D structure that determines its activity [5].

The existence of interactions among proteins may be determined by techniques such as proteome chips [3], microarrays [6], and Strep-tag® [7]. Although outstanding progress has been made, these *in vitro* methods remain both time-consuming and expensive when a large number of proteins are involved [8, 9]. On the other hand, computational or *in silico* approaches, based on machine learning techniques, allow for the scanning of a large number of proteins in a limited amount of time [10–18]. Therefore, given a target protein, it is possible to rapidly screen a large number of binders with a PPI predictor to determine the most suitable candidate for binding. Nevertheless, the screening process is limited to known proteins and may not necessarily be suitable for all applications, e.g., the design of a therapeutic antibody for a new or orphan disease [19]. As a result, it is not possible to directly generate new proteins with a screening method. Therefore, there is a great need for solutions, especially those based on deep learning, that can predict non-existent binders in addition to known ones.

This study addresses this problem. More specifically, it is proposed, given the amino acid sequence of a target protein, to determine the amino acid sequence of the corresponding binder without recourse to domain knowledge such as structural information. In our work, we employ a novel transformer-based deep learning architecture to realize this objective.

As one of the most prominent deep learning algorithms, the transformer has achieved success in fields including natural language processing (NLP) [20], computer vision (CV) [21]. The transformer proposed is a Seq2seq algorithm for machine translation [22, 23]. In the English-to-German and

English-to-French translation challenges, the transformer outperforms previous state-of-art approaches. [23]. According to Lin et al. [24], transformers have the following advantages: 1. Transformers can achieve any pair of positions in the sequence in one step. Compared to recurrent neural networks, this property provides transformers a better capability for handling long-range dependencies. 2. As for convolutional neural networks (CNN), a multi-layered stack is needed to obtain a global receptive field, whereas transformers achieve this in only a single self-attention layer. 3. Even though fully connected layers also possess the above ability, transformers are more efficient in terms of parameters. 4. Transformers have a constant number of sequential operations, whereas RNNs depend on the length of the sequence, so transformers are more parallelizable. With the above motivations, we employ transformers as the backbone model of our algorithm.

The reason for focusing our attention on amino acid sequences is that they may be experimentally determined rapidly and cheaply with high-throughput systems [25], in contrast to 3-D structures, which must be determined with slow and expensive methods such as X-ray crystallography [26].

Amino acid sequences may be assimilated to pseudo-time series, i.e., the time corresponding to the position of the amino acid in the chain, because they are oriented: the beginning of a chain is marked by an amino-terminus (N-terminus), while the end is marked with a carboxyl-terminus (C-terminus) [27]. As such, deep learning techniques related to time series and natural language processing (NLP) may be transposed to predict amino acid sequences. Therefore, it is proposed to employ a transformer [23] for predicting the binders, a transformer being a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data to predict the output.

Another problem specific to proteins is that there is usually more than one possible binder for a given target protein [28]. As a result, the problem is multivalued (having more than one solution) and some additional constraints must be applied to be able to learn a particular solution (binder). Two constraints are employed in our deep learning solution: the binding score, which is related to the strength of the interaction, and a prior about the amino acid sequence of the binder. The score is either appended to the amino acid sequence of the target protein (input of the encoder) or injected directly into the latent space of the encoder. In order to further constrain the problem, a prior is applied to the output of the decoder, i.e., to the binder amino acid. This prior consists of the first n amino acids of the binder to ensure the uniqueness of the solution.

In this study, we proposed a primary sequence-based approach to design PPIs with two different transformers. We also address the multivalued problem in PPI design by intro-

ducing the binding score as well as a prior. Given 10% of the binder as a prior, both of the transformers are able to achieve a Needleman-Wunsch score of 0.86.

Our main contributions include (1) a novel deep-learning model that generates unique binders given a target protein, a binding score, and a prior on the binder, (2) two innovative transformer architectures that allow the binder generation to be conditioned by previous knowledge (conditional transformer), and (3) a new search strategy named Beam Search with Prior (BSP), which considerably reduces the computational complexity and error accumulation.

This paper is organized as follows: related work is covered in the following section, while the new transformer-based deep learning architectures employed for binder prediction are described in the next section. To the best of our knowledge, this is the first work that employs transformers for PPI design. The dataset and the preprocessing techniques are presented in the subsequent section and the transformer training procedure is described in “[Network training](#)”. The inference, the evaluation techniques, and the prior are addressed in “[Inference, evaluation, and prior](#)”, while the experimental results are reported in the subsequent section. Conclusion concludes the paper.

Related work

Protein design is the rational design of new protein molecules to design novel activity, behavior, or purpose and advance the basic understanding of protein function. Proteins may be designed from scratch (de novo design) or by making calculated variants of a known protein structure and its sequence (protein redesign). The most common methods for redesign are residue interactions clustering followed by high-affinity binding protein searching [29, 30], searching for an existing binding protein with high affinity that is subsequently optimized [31–33], and incorporating metal-binding sites while optimizing their periphery [34]. As for de novo design, the most common approaches are optimizing existing binding sites [35, 36] and grafting either one [37–39] or two sides [40] of an existing site. All these techniques rely on domain knowledge such as the ternary structures (3-D shape) and information about residue interactions and complexes. For instance, metal-binding sites are restricted to nonpolar proteins [34]. Frequently, information such as the ternary structure is unavailable, thereby considerably reducing the effectiveness and generalizability of these methods.

There are relatively few studies about protein design based on deep learning. Based on the criterion that if the study be related to sequence-based protein design, we select a few studies. Among the most important are signal peptide generation by attention-based neural networks [41], antibody design using a long short-term memory (LSTM)-based deep

generative model from a phage display library for affinity maturation [42], sequence-based deep learning antibody design for in silico antibody affinity maturation [43, 44], improving the accuracy of neural network-based computational protein sequence design with DenseNet [45] using backbone structures, computational protein design with deep learning neural networks [46], and the generation of functional protein variants with variational autoencoders [47]. Wu et al. approach lacks the ability to solve multivalued problems, and all these methods require either knowledge of the 3-D structure [45] or domain-specific knowledge [43, 44].

An approach for solving the antibody-antigen binding problem is proposed in [43]. Given an antibody-antigen pair, a one-hot embedded matrix and an adjacent matrix are first evaluated. Then, the two matrices are fed to a neighborhood aggregation layer and a graph neural network (GNN) to obtain a binary classification. Being confined to binary classification, the Kang et al. method is unsuitable for determining the sequences of binding proteins.

A single-domain antibody (sdAb), also known as a nanobody, is an antibody fragment consisting of a single monomeric variable antibody domain. Like a whole antibody, it can bind selectively to a specific antigen. Shin et al. employ an autoregressive dilated convolutional model for predicting nanobody sequences [44]. As the model is autoregressive, the amino acids are predicted sequentially, with the prediction of a new amino acid being based on the previous ones. Wang et al. propose a framework, consisting of three cooperating deep learning models, that predicts the amino acid sequence of the binder based on the backbone structure (3-D) of the target protein [46]. The applicability of this framework is limited to target proteins for which the backbone structure is known.

In one study, a variational autoencoder is employed to generate new proteins [47]. In this work, by modulating the latent representation, it is possible to generate new proteins. Unfortunately, the autoencoder does not account for the target protein, making it unsuitable for interacting proteins.

The approach by [45] is based on DenseNet [48], which is a convolutional neural network with several parallel skip connections that was originally introduced in computer vision for diagnostics and identification [49, 50] and more recently was used to diagnose COVID-19 infections from computed tomography scans [51]. In the proposed architecture, DenseNet is extended to volumetric data [45]. The latter consists of regular volumetric grids in which each volume element is related to the atomic density, while each grid corresponds to a particular atomic type, thus focusing on the 3-D structure. The amino acid sequence of the target residues is predicted from the distribution of backbone atoms. Qi et al. achieved an accuracy of 53.2% with a 5-fold cross-validation, which constitutes an improvement of more than 10% compared to the state of the art. Unfortunately, because of the

one-to-one correspondence between each amino acid and the backbone, the target protein and the binder must have the same number of amino acids, which strongly limits the applicability of this approach. Indeed, most of the time, the binder and the target protein do not have the same number of amino acids [52].

This limitation may be circumvented by employing an autoregressive model [53] in which the new amino acid is predicted recursively from the previous one. The recursive prediction process ends when an end-of-sequence (EOS) token is predicted. Therefore, the model can predict sequences with variable lengths, thus allowing for binders and targets not having the same number of amino acids, which is by far the most common scenario. Such a strategy was followed by Sake et al. [42] for antibody design against kynurenine, a metabolite used in the production of niacin. The autoregressive model consisted of an LSTM [54]. An extended dataset was obtained with the phage display laboratory technique [55] and an antigen-binding (Fab) library [56]. Sake et al. model can predict sequences with arbitrary lengths. Nonetheless, the predictions are based on a specific probability distribution, the one associated with the maturation of antibodies against kynurenine, which means that the approach cannot be easily generalized. A similar problem occurs in [44].

The study by Wu et al. is the one closest to our work [41]. The authors employ an autoregressive model to predict binders of variable length. Their training dataset consist of signal peptides (short amino acid sequences) and their corresponding target proteins, and their architecture is based on a seq2seq transformer. Recall that a transformer is a deep learning model that adopts the mechanism of self-attention [23]. As such, seq2seq transforms the target protein amino acid sequence into the binder amino acid sequence. The main components are the encoder and the decoder. The encoder converts each input sequence into a more informative and compact latent (hidden) representation. The decoder reverses this process, converting the latent representation into an output sequence using the previous output as the input context. The network includes two optimization mechanisms: attention and beam search. The input to the decoder is a single vector that stores the entire context. Attention allows the decoder to look at the input sequence selectively. As for beam search, instead of picking a single output (amino acid) as the output, multiple highly probable choices are retained while being represented as a data structure. For testing, de novo signal peptides were generated with the trained model. It was determined experimentally that 48% of the peptides bound effectively. These peptides shared on average 73% of their sequence with the corresponding closest known signal peptides.

Yet, there are some limitations to the method of Wu et al. [41]. First, it is essentially limited to small molecules, whereas our main concern is PPIs. Second, as opposed to

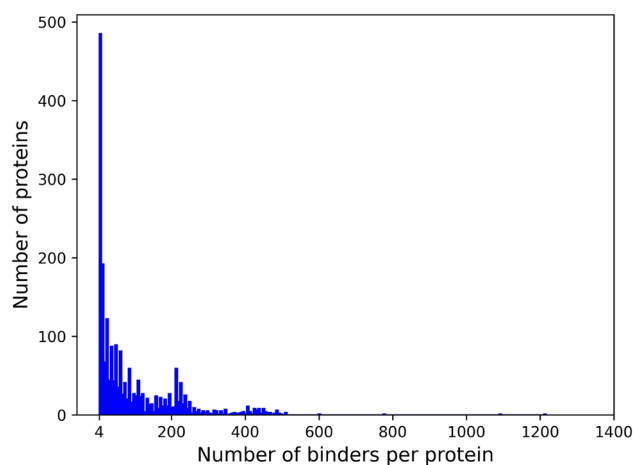


Fig. 2 Distribution of the number of interactions per protein in the STRING dataset

signal peptides, PPIs are less specific. As a result, a given target protein may interact with multiple binders. Indeed, only 0.13% of proteins studied by Wu et al. interacted with more than one binder. The situation is completely different for PPIs. For instance, in the Search Tool for the Retrieval of Interacting Genes/Proteins (STRING) dataset [57], most proteins interact with at least four binders. The distribution of the number of interactions per protein for the STRING dataset is reported in Fig. 2. Therefore, protein design is made more complex by the fact that there are multiple solutions for a given target protein (multivalued problem). This is to be contrasted with signal peptides, for which the solution is essentially unique, which means that the problem is more constrained.

As indicated earlier, all state-of-the-art methods either require knowledge of the 3-D structure of the target protein or domain-specific knowledge. Therefore, they may encounter difficulties when applied to new problems, meaning that their generalizability is rather limited. In addition, very often, the 3-D structure is not readily available. On the other hand, amino acid sequences may be easily obtained. For these reasons, we introduce a solution for designing binder proteins that is based on amino acid sequences. Our approach aims at designing target proteins solely from amino acid sequences while addressing the multivalued problem inherent to PPIs. The architectures of our models are presented in “[Transformer for binding protein prediction given a known target protein](#)”.

Dataset and preprocessing

The dataset employed for training and testing is the STRING dataset that was originally proposed by Szklarczyk et al. in 2019 [57]. For the species *Homo sapiens*, the dataset con-

sists of 11,759,455 interacting protein pairs, each of which is characterized by a binding score. Amino acid sequences consisting of more than 256 amino acids are excluded because the computational requirements for the self-attention mechanism grow quadratically with the number of amino acids [58]. In addition, since we are interested in binding proteins with high affinities, only proteins for which the binding score is between 0.9 and 1.0 are retained [59–62].

Recall that a transformer architecture is asymmetric in the sense that the binder may become the target protein and vice versa. Therefore, the dataset was extended by inverting the binder and the target protein. As a result, 96,744 interacting pairs involving 2091 distinct proteins were obtained in conjunction with their binding scores. Since a seq2seq model is employed, the amino acids are assimilated to tokens, while the amino acid sequences are assimilated to sentences. As in NLP, each sequence starts with a BOS token and ends with an EOS token. The BOS token is employed to initialize the recursive prediction at the inference stage, while the EOS token marks the end of the sequence. Therefore, a sequence of arbitrary length may be predicted. Sequences with fewer than 256 amino acids are padded with padding tokens to be able to predict smaller amino acid sequences.

For AppendFormer, the binding score is discretized into 100 bins ranging from 0.9 to 1.0, with each bin corresponding to a particular binding score. The number of bins was determined through experimentation. As a result, the vocabulary needed to be increased by 100 tokens in addition to those associated with the amino acids. For MergeFormer, since the binding score is injected directly at the level of the latent space, no additional tokens are required. Network training and the priors are addressed in “[Network training](#)”.

Transformer for binding protein prediction given a known target protein

Recall that a transformer is a deep learning architecture that transforms one sequence into another and that it is based on seq2seq. It is an autoregressive model with an attention mechanism that was first proposed for neural machine translation [23]. As opposed to recurrent neural networks [63], no sequential treatment of the input vector is required. In addition, the scaled dot product attention mechanism densely connects the input and output. Therefore, the context vector is linked to the entire input, thus overcoming the problem associated with long-term dependencies [64]. As stated earlier, PPI is a multivalued problem since more than one binder may be associated to a target protein, as reported in Fig. 2. To obtain a particular solution, some prior knowledge must be introduced in order to constrain the solution to a particular binder. To achieve a unique solu-

tion, it is proposed to employ the binding scores between target proteins and binders in addition to prior knowledge of the amino acid sequence of the binder. Here, the prior refers to the first amino acids of the target protein that are known.

In our study, two transformer architectures are introduced: one is optimized for short priors while the other is optimized for longer priors. The first AppendFormer architecture extends the original transformer as introduced by Vaswani et al. [23] with the objective of learning in the presence of short priors. The input AppendFormer transformer consists of the amino acid sequence of the target protein and the discretized binding score that is appended in between the padding and the EOS tokens (Fig. 3). The EOS token is employed to determine the localization of the discretized binding score within the input sequence. The architecture of the AppendFormer is reported in Fig. 4. The transformer consists of an encoder and a decoder, where the encoder transforms the target protein amino acid sequence into a latent representation, while the decoder generates the amino acid sequence of the corresponding binder protein from the latent representation. The decoder has a recurrent architecture, which means that its output is reinjected into its input as illustrated in Fig. 4. The output of the decoder is initialized with a beginning-of-sequence (BOS) token as well as with a prior about the binder amino acid sequence that consists of its first n amino acids. The Bayesian prior is employed to constrain the solution to a unique binder. Indeed, as stated earlier, because PPI is essentially a multivalued problem, it is necessary to introduce some constraints to obtain a unique solution. For both the encoder and the decoder, the amino acid sequences are processed with an embedding layer [23] that ensures that both sequences have a continuous representation while mapping amino acids with similar properties to the same region. As the attention mechanism does not account for the order in which the amino acids appear in the sequences, a positional encoder [23] is added after each embedding layer, thus allowing the transformer to account for the order in which the amino acids appear without jeopardizing parallel processing. The positional layer is followed by a multi-head self-attention layer. The aim of this layer is to capture the interrelations among the various amino acids such as self-folding and self-interaction either for the target protein or the binder, as demonstrated in [65]. Then, the latent representation is generated by a multi-head attention layer. For this layer, the query keys are associated with the decoder, while the memory and value keys are associated with the encoder, thus allowing the capture of the interrelations between the amino acids of the target protein (receptor) and those of the binders. The multi-head attention layer is followed by a classifier that consists of a feed-forward neural network, a linear, and a softmax activation function. The classes correspond to the twenty standard amino acids as well as to a potentially

unknown amino acid (multi-class problem). During the training phase, the classifier allows for determining if the amino acid generated by the decoder is the right one. Various skip connections [66] are added along the way to reduce the complexity of the loss function landscape, therefore facilitating its optimization [67].

All the amino acid sequences employed in the experiments come from the Protein Data Bank. As such, they were determined experimentally and curated. When sequencing the amino acids, if the procedure is inconclusive for a particular amino acid, the latter is considered unknown and labelled as X. These unknown amino acids constitute the noise, or better said, the uncertainty associated with the data. Therefore, the unknown amino acids are also employed in addition to the twenty standard amino acids during training. This allows the transformer not only to handle unknown data, but also to assess if the experimental determination of an amino acid may be potentially challenging when predicting new sequences.

We also introduce the MergeFormer architecture, designed for longer priors, i.e., a scenario in which a larger portion of the target protein is known. In the second architecture, a merge layer is introduced between the encoder and the decoder to inject the binding score deeper into the encoder just after the feed-forward neural network. Therefore, the binding score is decoupled from the amino acid sequence as opposed to the first architecture. The MergeFormer architecture is described in Fig. 5. The input of the encoder consists of the amino acid sequence of the target protein. The network is designed in such a way that the effect of the binding score may be learned independently for each amino acid. Therefore, a constant vector is created, the constant being the binding score, which has the same dimension as the latent or hidden representation. This vector is concatenated to the latent representation, the resulting vector is transposed, and a linear transformation is applied, thus allowing for the learning of the interrelations between the binding score and the amino acids. Finally, the linearly transformed representation is transposed to restore the original topology. The main advantages of MergeFormer are:

- No vocabulary is required for the binding affinity, thus reducing the number of tokens that must be learned by the network.
- For AppendFormer, the discretized binding score is appended at the end of the protein sequence. Since the amino acid sequence may have an arbitrary length, the position of the binding score is unknown and therefore must be learned by the network. In the case of MergeFormer, the binding score is injected directly into the latent space while its relation to each amino acid is learned.

Table 1 Hyperparameters associated with the transformers: $n_{encoder}$ and $n_{decoder}$ represent the number of layers in the encoder and the decoder, respectively, d_{model} and d_{ffn} are the dimensions of the embedding and feed-forward neural network, respectively, and P_{drop} is the dropout probability. The number of heads in the multi-head attention mechanism is given by n_{head} , while the size of the input and output vocabulary is represented by v_{input} and v_{output} , respectively. For MergeFormer, these are equal as they correspond to the sum of the number of amino acids, the number of special amino acids, and the number of special tokens. For AppendFormer, v_{input} is larger by the number of tokens

Model	MergeF	AppenF-s	AppenF-m	AppenF-l
$n_{encoder}$	3	3	3	3
$n_{decoder}$	3	3	3	3
P_{drop}	0.1	0.2	0.2	0.2
d_{ffn}	1024	1024	1280	1792
d_{model}	256	256	256	256
n_{head}	8	8	8	8
v_{input}	31	131	131	131
v_{output}	31	31	31	31

- The fact that the binding score is independent of the amino acid sequence allows for initializing the encoder weights with techniques such as masked language modeling (MLM), which has been shown to substantially improve performances in NLP when employed in conjunction with the bidirectional encoder representations from transformers (BERT) [68].

The hyperparameters employed for both transformers are reported in Table 1. The table consists of four transformers, namely one MergeFormer architecture and three AppendFormer architectures, with the difference in the three AppendFormer architectures being the number of neurons in the feed-forward neural network.

To further evaluate the proposed transformers, their performances are compared against a seq2seq baseline with LSTM layers [22]. We employ this model as a baseline since it famously surpassed the state of the art for machine translation in the Conference on Machine Translation (WMT' 14) English to French Challenge [69]. To date, it remains one of the most widely used seq2seq models and has been successful in a wide range of applications [70–72]. Specifically, it has also been successfully adopted in protein design applications [73–75].

Its architecture is described in Fig. 6. As for AppendFormer and MergeFormer, baseline model consists of an encoder and a decoder. The key difference is that the attention layers are replaced by LSTM layers. The number of neurons in the LSTM layers were set to 512 by inspection. The baseline model in AppendFormer and MergeFormer possesses embedding layers that convert the amino acid sequences into

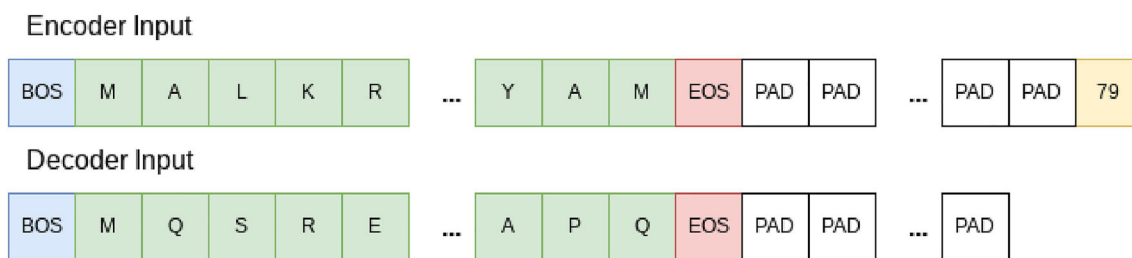
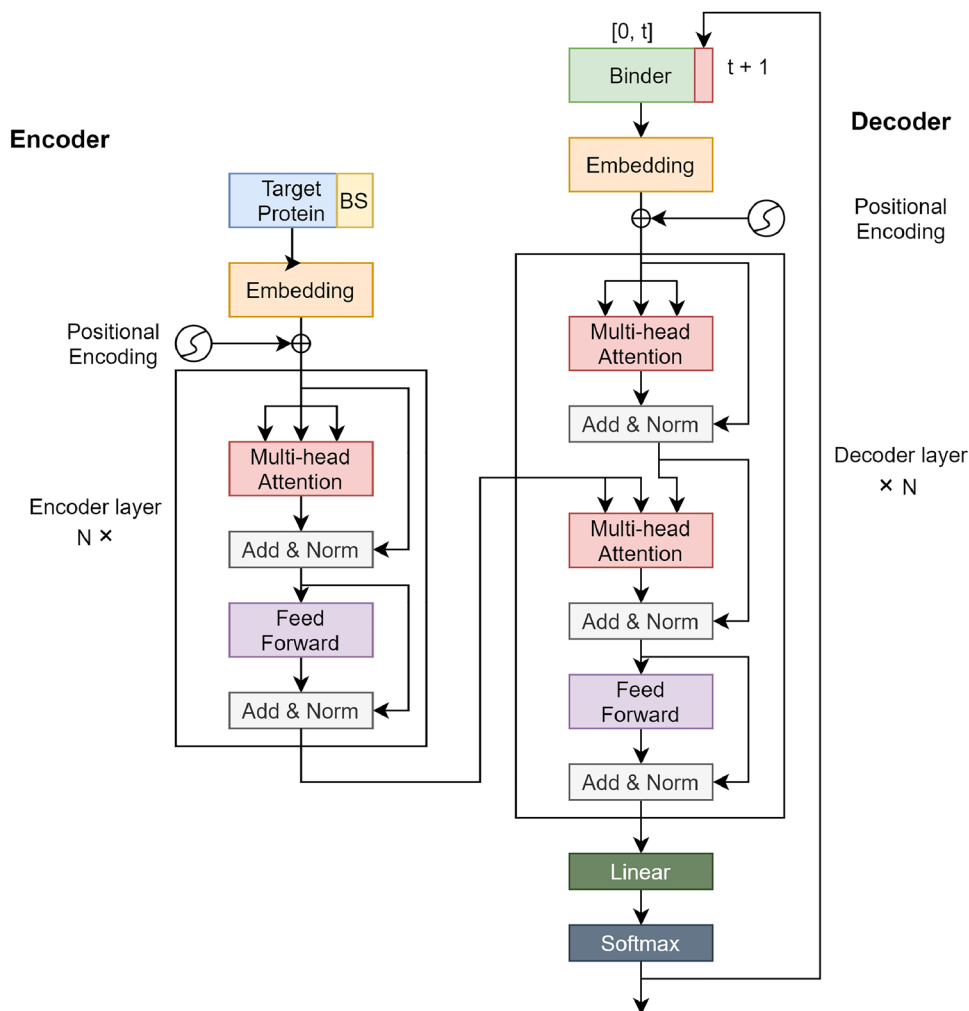


Fig. 3 Example input for encoder and decoder in AppendFormer

Fig. 4 Architecture of AppendFormer. The digitized binding score BS is inserted at the end of the sequence. The number N of encoder layers was 3 in this study

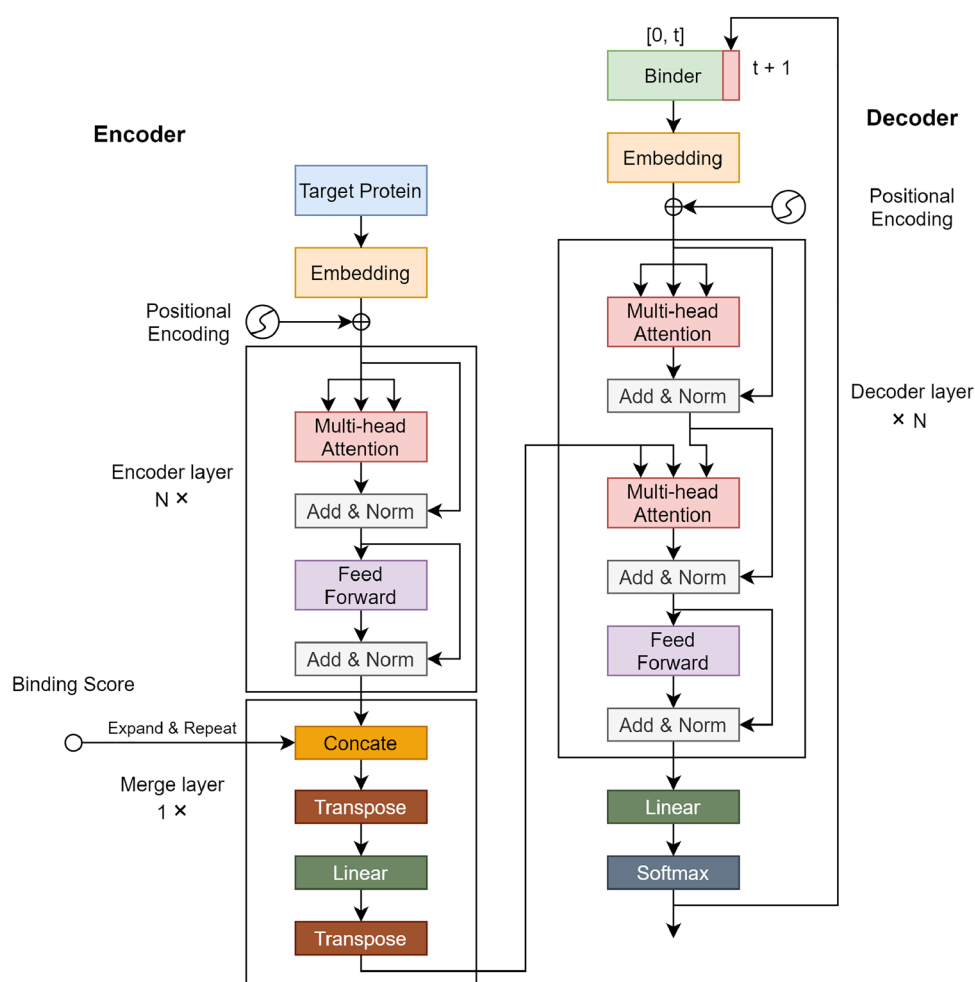


hidden or latent representations that are subsequently processed by the LSTM layers. The LSTM layer associated with the decoder is initialized from the internal states of the encoder’s LSTM layer. These internal states consist of the cell state and the hidden state. The decoder’s LSTM autoregressively generates the amino acid sequence for the binder, given the target protein. In AppendFormer and MergeFormer, the amino acids are predicted with a linear layer employing a softmax activation function.

Network training

Teacher forcing, which is an essential part of many seq2seq architectures [76], is employed at the training stage. Being an autoregressive model, a transformer predicts the next amino acid from the latent representation of the input sequence (target protein) and the previous predictions of the decoder. In teacher forcing, during training, the predicted value at position t is replaced by the true value if the prediction is erroneous. As indicated earlier, given a target protein, the

Fig. 5 Architecture of MergeFormer. The input is the amino acid sequence of the target protein, while the binding score is concatenated with the latent representation. $N = 3$ in this study



solution in terms of binders is multivalued. Therefore, due to the multiplicity of solutions, an error may be made when predicting an amino acid. The model being autoregressive, this error may jeopardize subsequent predictions in addition to causing error accumulation. Teacher forcing avoids the accumulation of errors by providing the decoder with the true input, resulting in a faster convergence during the training phase.

With teacher forcing, the training may be drastically accelerated. Without teacher forcing, the predictions have to be made recursively because the decoder's output is reinjected into the decoder's input. This means that the prediction for the current position is always based on the previous one. The recursive process eliminates any possibility for parallel training. However, with teacher forcing, the decoder is provided with ground truth labels rather than the model's previous predictions. Therefore, the sequence prediction may be partitioned into multiple independent sub-tasks (parallelization). These sub-tasks can be distributed to many nodes and calculated concurrently, thus accelerating the training process.

The loss function is optimized with the Adaptive Moment Estimation (ADAM) stochastic optimizer technique [77].

For the loss function, with teacher forcing training, each amino acid prediction on the chain may be performed independently, therefore resulting in independent multiclass classification problems, with the classes corresponding to the amino acid types. It follows that the loss function is a sparse categorical cross entropy [78] that is described by Eq. 1. w is the weights of the model. The output of the model and the ground-truth label are noted as \hat{y} and y respectively. As in [23], the padding tokens are ignored during training.

$$J(w) = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) \quad (1)$$

As for the original transformer, a dynamic learning rate with warm-up and decay is employed as it improves convergence during training. We use the same for our AppendFormer model since it is similar to the original transformer architecture as shown in Eq. 2. For our MergeFormer model, we design a dynamic learning rate with spiking followed by

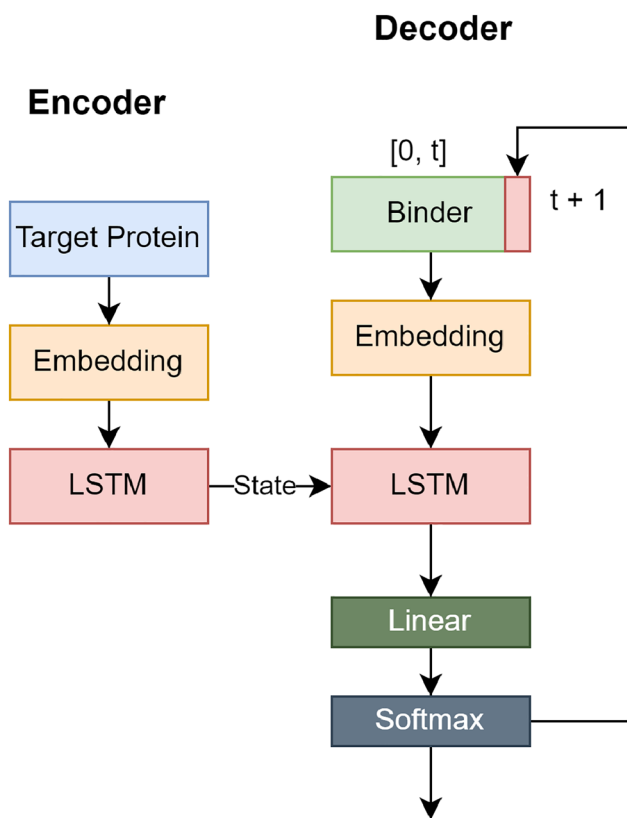


Fig. 6 Architecture of the seq2seq baseline model

a constant learning rate, as described by Eq. 3.

$$l_{\text{appe}} = d_{\text{model}}^{-0.5} \cdot \min(n_s^{-0.5}, n_s \cdot n_{\text{ws}}^{-1.5}) \tag{2}$$

$$l_{\text{merg}} = \begin{cases} l_{\text{init}} \cdot \frac{n_s}{n_{\text{ws}}} & n_s \leq n_{\text{ws}} \\ \max(l_{\text{init}} \cdot \frac{1}{1 + \alpha \cdot (n_s - n_{\text{ws}})}, \epsilon) & n_s > n_{\text{ws}} \end{cases} \tag{3}$$

where n_s represents the number of training steps, n_{ws} the number of warming steps, ϵ the lower boundary, α the learning rate decay speed, and l_{init} the initial learning rate.

The learning rate for MergeFormer is slightly different. To avoid local minima, both the lower boundary and the initial learning rate must be smaller [79]. The learning rates for both AppendFormer and MergeFormer are reported in Fig. 7.

Inference, evaluation, and prior

Recall that both AppendFormer and MergeFormer are autoregressive models, which means that the prediction of a new amino acid is based on the previous one. The encoder generates a latent representation from the target amino acid sequence and the binding score, the latter being appended to the sequence in AppendFormer or injected into the latent space in the case of MergeFormer. The latent representation

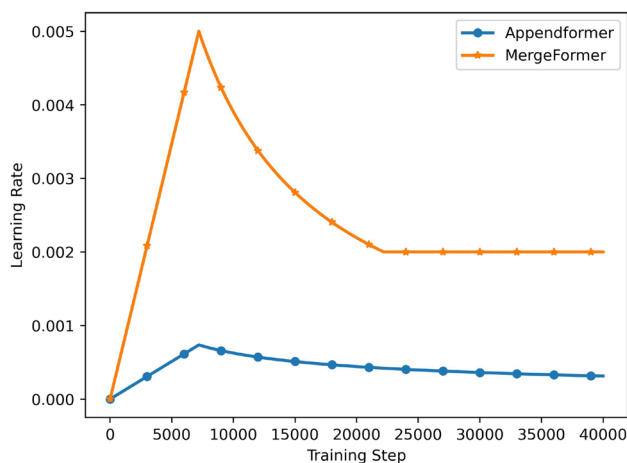


Fig. 7 Dynamic learning rates for AppendFormer and MergeFormer

acts as a key and a value for the multi-head attention mechanism associated with the decoder.

Initially, when the amino acid sequence of the binder is completely unknown, the sequence is initialized with a BOS. Then, the decoder predicts the first amino acid. Being a multiclass classifier, the decoder predicts an occurrence probability for each of the twenty amino acid types or classes. The amino acid with the highest probability is selected with a greedy search strategy [80]. The output sequence is updated with the outcome of the greedy search, and the recursion is repeated until an EOS is generated by the discriminator, allowing for the prediction of sequences of arbitrary length.

The greedy search algorithm only retains the amino acid with the highest classification probability. Because there is more than one binder for a given target, many correct solutions are possible, at least at the beginning of the prediction process. The real and unique solution may only be inferred when more amino acids are predicted along the way. This problem may be overcome with beam search [81]. Instead of considering only the first b_w predicted amino acid, beam search considers the first predicted amino acid, with the last one being conditional on the first $b_w - 1$. For instance, for $b_w = 2$, the probabilities associated with both amino acids (bi-gram) are multiplied to obtain the conditional probability of the bi-gram, with the probability of the second amino acid being conditional upon the first one. Then, the bi-gram with the highest conditional probability is selected and the process is repeated for each new prediction. As a result, the results are improved, and the issues associated with the multivalued problem are mitigated.

To further constrain the multivalued problem, a prior is attributed to the binder. Such priors are common in protein design, especially when mutation-based techniques are employed [35–40]. Their application is based on the observation that, without the prior, the transformer would not have enough information, or constraints, to converge onto a par-

ticular solution [82]. In our work, a prior consists of the first n amino acids forming the binder for both AppendFormer and MergeFormer. In the first case, the prior is appended to the input of the AppendFormer, while, in the second case, the prior is injected directly into the latent space between the encoder and the decoder. Depending on its length, the prior may be assimilated to a peptide or a polypeptide. The prior may be assimilated as a form of teacher forcing. Indeed, for the first n amino acids of the binder, the amino acids generated by the decoder are replaced by the corresponding amino acids of the prior. From the $n + 1$ prediction, the amino acid is directly predicted via the decoder with beam search. Additionally, during the predicting stage, teacher forcing is only applied within the section of prior. During training, in contrast, teacher forcing is engaged in the entire process.

Because the ground truth is known for the prior, it is not involved in the evaluation. Therefore, all the evaluation metrics ignore the prior and only focus on the amino acid directly generated by the decoder, thus avoiding the introduction of bias in the evaluation procedure. Five metrics are employed for the evaluation, namely the token matching rate (TMR), bilingual evaluation understudy (BLEU), Needleman-Wunsch, Smith-Waterman metrics, and the Metric for Evaluation of Translation with Explicit Ordering (METEOR).

- **Token matching rate:** The token matching rate is evaluated over the whole test set. It corresponds to the ratio between the total number of tokens (amino acids) correctly predicted to the total number of tokens. The metric ranges between 0 and 1.

$$TMR = \frac{n_{\text{token_match}}}{n_{\text{token}}} \quad (4)$$

- **BLEU metric:** The BLEU score [83] is a metric that was originally developed for evaluating the quality of texts that have been machine-translated from one natural language to another. In this study, the BLEU score is the logarithm of the geometric mean of the modified n -gram precisions for the amino acid prediction, with an n -gram corresponding to a contiguous sequence of amino acids of length n . Short n -grams are penalized with a brevity penalty score [83]. Here, the value of n ranges between 1 and 4, while the BLEU score lies between 0 and 1. The metric depends only on the n -grams and not on their positions within the sequence. Therefore, the BLEU score determines the similarity between the true and predicted sequence in terms of their n -grams. The BLEU score is also related to the protein structure quality. Indeed, the number of amino acid substitutions is significantly correlated with neighborhood preference among amino acids

[84]. As reported by [85], the neighborhood preference may be employed to assess structural quality.

- **Needleman-Wunsch metric:** The Needleman-Wunsch algorithm determines the best global alignment and similarity measure between two protein sequences with dynamic programming [86]. Given an amino acid sequence, each correctly predicted amino acid is attributed a reward of 1, while a penalty of 0 is attributed otherwise. The similarity measure is the ratio between the sum of the rewards and penalties and the total number of amino acids. This is a number between 0 and 1 whose value is independent of the number of amino acids forming the sequence.
- **Smith-Waterman metric:** The Smith-Waterman algorithm [87] is based on local alignment, which compares amino acid segments of all possible lengths and optimizes the similarity measure with dynamic programming. The measure itself is evaluated in the same way as for the Needleman-Wunsch metric. It is a number between 0 and 1 whose value is independent of the number of amino acids forming the sequence.
- **METEOR:** Like the other metrics, METEOR measures the similarity between the generated and the targeted sequences. The main difference with the other metrics is that METEOR first aligns the two sequences and then calculates the harmonic mean of unigram precision and recall while putting more emphasis on recall [88]. From an information retrieval point of view, it enables stemming and synonymy matching, this is impossible with the other metrics [88].

To further ascertain the performances of the transformers, the classification accuracy and the top-5 classification accuracy are evaluated. Absorbed from classification tasks, Top-5 accuracy means that the predicted amino acid is among the five best results in terms of prediction probability.

The hyperparameters of the transformers were determined by evaluating and comparing the performances of the models for various values of the hyperparameters. As stated earlier, the sequences are evaluated recursively, meaning that the training process cannot be parallelized making the evaluation of the hyperparameters computationally prohibitive. For this reason, as explained in “[Network training](#)”, teacher forcing was employed to train each model. Indeed, teacher forcing allows the prediction of each amino acid independently of the others, thus allowing for parallelization of the calculations which reduces the time required for training [89].

Our experimental results are presented in the next section.

Experimental results

The best models and hyperparameters were determined from the accuracy and the top-5 accuracy as obtained by 5-fold

Table 2 Number of trainable parameters, 5-fold cross-validation accuracy and top-5 accuracy with teacher forcing

Model	Trainable parameters	Accuracy	Top-5 accuracy
AppendFormer-s	5,579,039	93.07% ± 0.20%	98.00% ± 0.04%
AppendFormer-m	6,367,007	94.03% ± 0.11%	98.51% ± 0.01%
AppendFormer-l	7,942,942	95.81% ± 0.42%	99.02% ± 0.09%
Baseline	3,173,663	96.53% ± 0.29%	98.41% ± 0.16%
MergeFormer	5,620,519	96.41% ± 0.48%	98.38% ± 0.31%

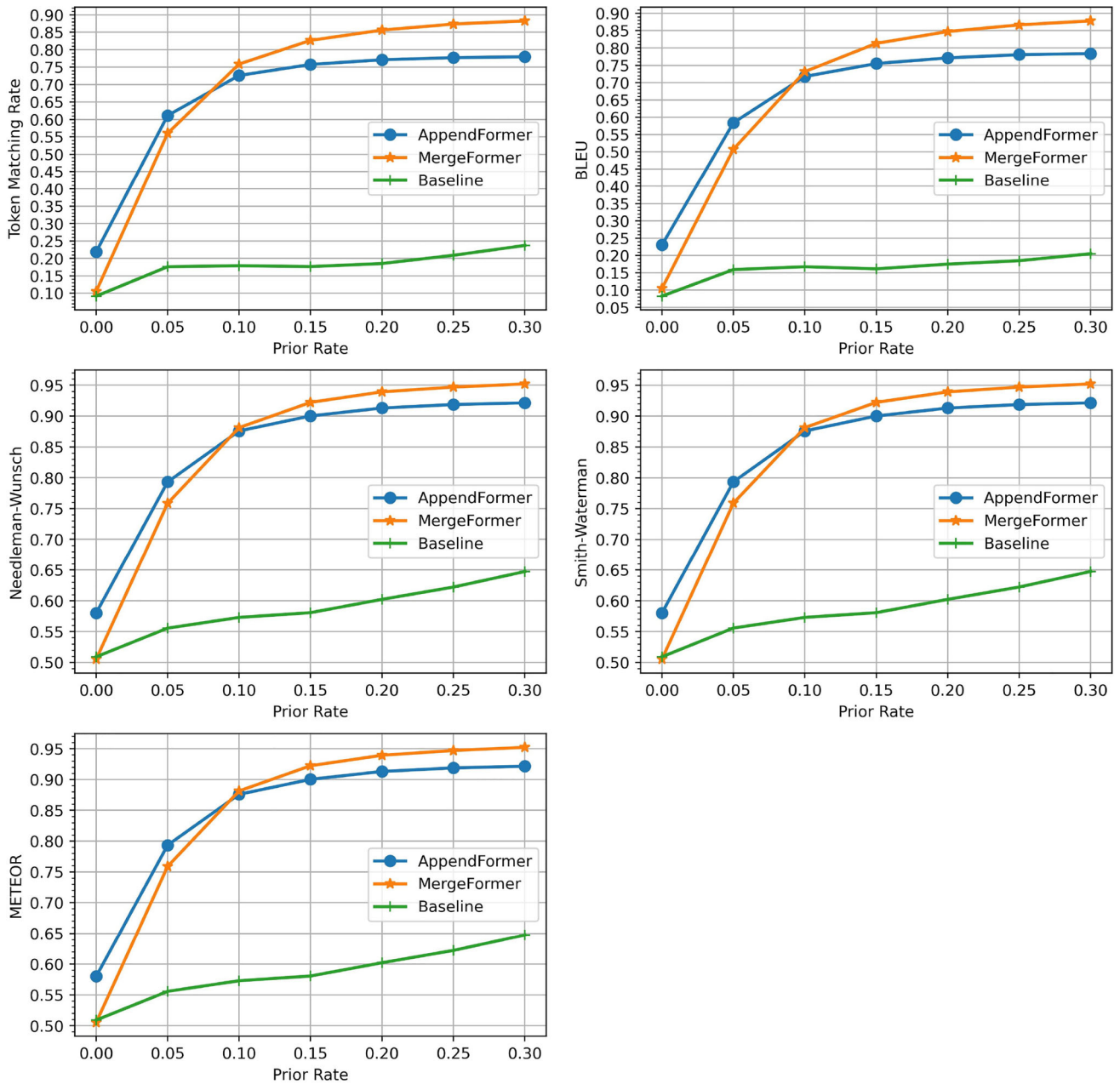


Fig. 8 Evaluation of the performances of AppendFormer, MergeFormer, and the seq2seq baseline model according to the importance of the prior for five metrics: the token matching rate, BLEU, Needleman-Wunsch, Smith-Waterman, and METEOR

cross-validation with teacher forcing. Recall that teacher forcing was employed to parallelize amino acid predictions, thus allowing for faster convergence. The best performances for AppendFormer (AppendFormer-1) were obtained with a feed-forward neural network consisting of 1792 neurons. The number of parameters, the accuracy and the top-5 accuracy for each model are reported in Table 2. While MergeFormer and AppendFormer-1 have comparable performances in terms of accuracy, MergeFormer has much less parameters than AppendFormer-1, thus reducing the risk of overfitting while facilitating convergence during the training phase. AppendFormer-1 has slightly better performance in terms of top-5 accuracy than MergeFormer, while the situation is the opposite for accuracy.

The models that obtain the best-three accuracy-AppendFormer-1 and MergeFormer-and the baseline seq2seq model were evaluated with the token matching rate, the BLEU, Needleman-Wunsch, Smith-Waterman metrics, and METEOR. It should be noted that teacher forcing was not employed during the testing phase as it would bias the results by providing the network with the correct solution. These metrics were evaluated for different priors on the binding proteins. The priors were obtained by providing the decoder with a certain percentage of the first amino acids forming the binder. This number was restricted in between 0% and 30% of the total length. As stated in Sect. 4, this prior is required in order to constrain the multivalued problem as multiple binders may bind to a given target protein. Therefore, the prior ensures that the right binder is found for a particular interaction.

The results for the four metrics are reported in Fig. 8. First, we turn our attention to the Needleman-Wunsch and Smith-Waterman metrics since they are widely used in bioinformatics. The best performances in terms of the Needleman-Wunsch and Smith-Waterman metrics are obtained with AppendFormer-1 when no prior is employed, the metrics being equal to 0.58. This value increases to 0.61 for AppendFormer-1 with a 5% prior, 0.86 with a 10% prior (both for AppendFormer-1 and MergeFormer) and up to 0.95 with a 30% prior with MergeFormer. AppendFormer-1 slightly outperforms MergeFormer when the prior is inferior to 10%, while MergeFormer strongly outperforms AppendFormer-1 otherwise, improving performances by up to 40% when a 30% prior is employed. In comparison, as illustrated by Fig. 8, the seq2seq baseline model has relatively poor performances with the Needleman-Wunsch and Smith-Waterman metrics, barely reaching 0.65 when a 30% prior is employed. These results show that AppendFormer is suited for short priors (less than 10% of the entire sequence), while MergeFormer performs better with longer priors (more than 10%), thus confirming the validity of our architectures. Both transformers outperform the baseline seq2seq model. The performances in terms of Needleman-Wunsch and Smith-Waterman metrics are quite similar, meaning that no local

alignment is required, and thus, that the sequence is properly predicted up to a global alignment. The same general behavior is observed for the token matching rate, BLEU, and METEOR results. Although these three metrics were reported for completeness, as stated above, the Needleman-Wunsch and Smith-Waterman metrics are the most relevant from a bioinformatics perspective. In contrast, researchers familiar with NLP evaluation metrics utilize token matching, BLEU, and METEOR scores more often.

Park and Marcotte [90] and Hamp and Burkhard [91] propose three evaluation procedures- C1, C2, and C3. In C1, it is assumed that some ligands and receptors appear in the training and validation sets; in C2, only some receptors appear in both groups; and in C3, no receptors are present in the validation set. The C1 procedure corresponds to the original experimental setup. Regarding the C2 procedure, we created a new dataset and conducted another round of experiments against it. This dataset was created by removing protein pairs for which the receptor was present in the training and validation sets. As reported in Fig. 9, the obtained results are comparable with those achieved with C1. For instance, for the bioinformatics metrics (Needleman-Wunsch and Smith-Waterman), a value of 0.86 was obtained, assuming a 10% prior on the ligand. In comparison, a value of 0.88 is reached with the C1 counterpart (Fig. 8). Besides, the best results for C1 are achieved with MergeFormer, while AppendFormer performs better with C2. That may be explained by the fact that the learnable alignment procedure associated with MergeFormer is more challenging when none of the ligands is present in both sets. Such an alignment procedure is absent from the AppendFormer model. The C3 evaluation procedure is not an appropriate choice in our experimental setting. In our work, the prior becomes challenging if none of the receptors are present in the training and validation sets. The reader should notice that the need for a prior corresponds to real-world settings. The receptor, which usually corresponds to the disease (e.g., a SARS-CoV-2 spike protein), is known in practice. In contrast, the ligand, which corresponds to a new therapeutic protein, is either unknown or partially known. Therefore, some receptors may appear in both sets without compromising the prediction of new therapeutic proteins.

Indeed, these results further highlight the successes of AppendFormer and MergeFormer.

Conclusions

This paper introduced novel deep learning architectures for therapeutic antibody and drug design based solely on amino acid sequences. A new primary sequence based approach for predicting the amino acid sequence of the corresponding binder from a given target protein amino acid sequence was introduced. As a target protein may interact with more

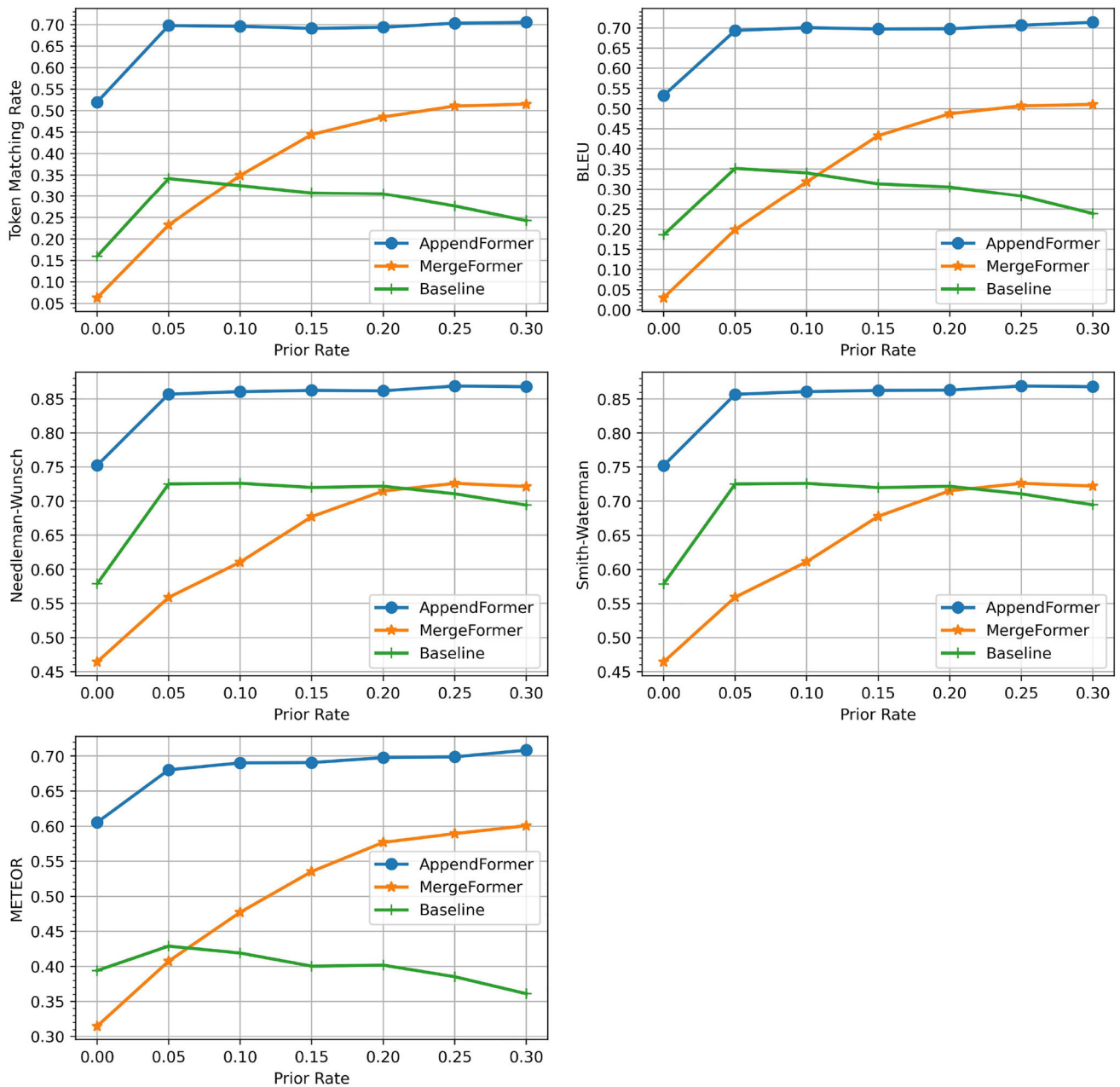


Fig. 9 Evaluation of the performances of AppendFormer, MergeFormer, and the seq2seq baseline model according to the importance of the prior for four metrics, namely the token matching rate, BLEU, Needleman-Wunsch, Smith-Waterman, and METEOR with the C2 evaluation procedure

than one protein, the problem was constrained by employing the binding score and a prior on the binder. The binder was predicted using the transformer deep learning approach. Two architectures were created: a transformer in which the discretized binding score is appended to the target amino acid sequence and a transformer for which the binding score is injected directly into the latent space between the encoder and the decoder, resulting in fewer parameters and a learnable relation between the binding score and the latent space. To improve convergence and parallelize the calculations, train-

ing was performed with teacher forcing. The best results were obtained with AppendFormer-1 when the prior was less than 10.

A limitation of the proposed approach is the reliance on the prior. Indeed, the receptor (target protein) sequence and a portion of the ligand (binder protein) sequence must be known to predict the unknown part. It is often the case in practice but may become more challenging when little information is known. In addition, the prior restricts the number of solutions to one. Sometimes, a plurality of solutions is

desirable, especially when toxicity issues arise with some generated proteins.

In the future, it is proposed to generate binders with a generative adversarial network (GAN) [92] in which the generator would be a transformer and the discriminator would be a pyramid neural network [65]. The discriminator would determine if the target and binder proteins interact or not. In this extended framework, the transformer would not require any prior information. Rather, all the binders would be generated directly by the GAN, thus overcoming the problems associated with multivalued solutions in addition to providing the most possible solutions.

As for most transformer-based architectures, the computational complexity bottleneck comes from the self-attention mechanisms, which grow quadratically with the number of amino acids [58]. This is why the number of amino acids was limited to 256. In future work, it is proposed to employ attention mechanisms with linear complexity, such as those described in [93] and [94], to handle very long sequences.

Acknowledgements The authors would like to thank the National Research Council of Canada for its financial support through the Artificial Intelligence for Design (AI4Design) Challenge Program.

Data availability In the interest of reproducibility, all data required for conducting experiments will be made available upon request with a license that allows free usage for research purposes.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Cooper GM, Hausman RE, Hausman RE (2007) The cell: a molecular approach, vol 4. ASM Press, Washington
- Raman D, Sobolik-Delmaire T, Richmond A (2011) Chemokines in health and disease. *Exp Cell Res* 317(5):575–589. <https://doi.org/10.1016/j.yexcr.2011.01.005>
- Zhu H et al (2001) Global analysis of protein activities using proteome chips. *Science* 293(5537):2101–2105. <https://doi.org/10.1126/science.1062191>
- Pinto D et al (2020) Cross-neutralization of SARS-CoV-2 by a human monoclonal SARS-CoV antibody. *Nature* 583(7815):290–295. <https://doi.org/10.1038/s41586-020-2349-y>
- Murray RK, Granner DK, Mayes PA, Rodwell VW (2006) Harper's illustrated biochemistry (harper's biochemistry), 27th edn. McGraw-Hill Medical
- Neiswinger J et al (2016) Protein microarrays: flexible tools for scientific innovation. *Cold Spring Harb Protoc* 2016:837–839. <https://doi.org/10.1101/pdb.top081471>
- Herzberg C et al (2007) SPINE: a method for the rapid detection and analysis of protein-protein interactions in vivo. *Proteomics* 7(22):4032–4035. <https://doi.org/10.1002/pmic.200700491>
- Li Y et al (2021) Robust and accurate prediction of protein-protein interactions by exploiting evolutionary information. *Sci Rep* 11(1):1–12. <https://doi.org/10.1038/s41598-021-96265-z>
- Hosur R, Xu J, Bienkowska J, Berger B (2011) iWRAP: an interface threading approach with application to prediction of cancer-related protein-protein interactions. *J Mol Biol* 405(5):1295–1310. <https://doi.org/10.1016/j.jmb.2010.11.025>
- Guo Y, Yu L, Wen Z, Li M (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res* 36(9):3025–3030. <https://doi.org/10.1093/nar/gkn159>
- You Z, Ming Z, Niu B, Deng S, Zhu Z (2013) A SVM-based system for predicting protein-protein interactions using a novel representation of protein sequences. *Intelligent Computing Theories*. 629–637
- Shen J et al (2007) Predicting protein-protein interactions based only on sequences information. *Proc Natl Acad Sci* 104(11):4337–4341. <https://doi.org/10.1073/pnas.0607879104>
- Rao VS, Srinivas K, Sujini GN, Kumar GNS (2014) Protein-protein interaction detection: methods and analysis. *Int J Proteomics* 2014:147648. <https://doi.org/10.1155/2014/147648>
- Guo Y et al (2010) PRED_PPI: a server for predicting protein-protein interactions based on sequence data with probability assignment. *BMC Res Notes* 3(1):1–7. <https://doi.org/10.1186/1756-0500-3-145>
- Martin S, Roe D, Faulon J-L (2005) Predicting protein-protein interactions using signature products. *Bioinformatics* 21(2):218–226. <https://doi.org/10.1093/bioinformatics/bth483>
- Hamp T, Rost B (2015) Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinformatics* 31(12):1945–1950. <https://doi.org/10.1093/bioinformatics/btv077>
- Tastan O, Qi Y, Carbonell J, Klein-Seetharaman J (2009) Prediction of interactions between HIV-1 and human proteins by information integration. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. 527: 516–27. https://doi.org/10.1142/9789812836939_0049
- Maetschke SR, Simonsen M, Davis MJ, Ragan MA (2012) Gene ontology-driven inference of protein-protein interactions using inducers. *Bioinformatics* 28(1):69–75. <https://doi.org/10.1093/bioinformatics/btr610>
- Ikemura N et al (2021) SARS-CoV-2 Omicron variant escapes neutralization by vaccinated and convalescent sera and therapeutic monoclonal antibodies. *medRxiv*. <https://doi.org/10.1101/2021.12.13.21267761>
- Wolf T et al. Qun L, David S (eds) (2020) Transformers: state-of-the-art natural language processing. In: Qun L, David S (eds) *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45 (Association for Computational Linguistics, Online, 2020)
- Khan S et al (2022) Transformers in vision: a survey. *ACM Comput Surv (CSUR)*. <https://doi.org/10.1145/3505244>
- Sutskever I, Vinyals O, Le Q et al. (eds) (2014) Sequence to sequence learning with neural networks. In: Z, G., M, W., C, C., N, L., D & K, W., Q (eds) *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*
- Vaswani A et al (2017) Attention is all you need. *Adv Neural Inf Process Syst* 30:6000–6010

24. Lin T, Wang Y, Liu X, Qiu X (2021) A survey of transformers. arXiv preprint [arXiv:2106.04554](https://arxiv.org/abs/2106.04554)
25. Rigaut G et al (1999) A generic protein purification method for protein complex characterization and proteome exploration. *Nat Biotechnol* 17(10):1030–1032. <https://doi.org/10.1038/13732>
26. Tong AHY et al (2002) A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science* 295(5553):321–324. <https://doi.org/10.1126/science.1064987>
27. Krishna MM, Englander SW (2005) The N-terminal to C-terminal motif in protein folding and function. *Proc Natl Acad Sci* 102(4):1053–1058. <https://doi.org/10.1073/pnas.0409114102>
28. De Las Rivas J, Fontanillo C (2010) Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Comput Biol* 6(6):e1000807
29. Fleishman SJ et al (2011) Hotspot-centric de novo design of protein binders. *J Mol Biol* 413(5):1047–1062. <https://doi.org/10.1016/j.jmb.2011.09.001>
30. Fleishman SJ et al (2011) Computational design of proteins targeting the conserved stem region of influenza hemagglutinin. *Science* 332(6031):816–821. <https://doi.org/10.1126/science.1202617>
31. Butz M, Kast P, Hilvert D (2014) Affinity maturation of a computationally designed binding protein affords a functional but disordered polypeptide. *J Struct Biol* 185(2):168–177. <https://doi.org/10.1016/j.jsb.2013.03.008>
32. Jha RK et al (2010) Computational design of a PAK1 binding protein. *J Mol Biol* 400(2):257–270. <https://doi.org/10.1016/j.jmb.2010.05.006>
33. Procko E et al (2013) Computational design of a protein-based enzyme inhibitor. *J Mol Biol* 425(18):3563–3575. <https://doi.org/10.1016/j.jmb.2013.06.035>
34. Der BS et al (2012) Metal-mediated affinity and orientation specificity in a computationally designed protein homodimer. *J Am Chem Soc* 134(1):375–385. <https://doi.org/10.1021/ja208015j>
35. Kosloff M, Travis AM, Bosch DE, Siderovski DP, Arshavsky VY (2011) Integrating energy calculations with functional assays to decipher the specificity of G protein-RGS protein interactions. *Nat Struct Mol Biol* 18(7):846–853. <https://doi.org/10.1038/nsmb.2068>
36. Chen TS, Palacios H, Keating AE (2013) Structure-based redesign of the binding specificity of anti-apoptotic Bcl-x_L. *J Mol Biol* 425(1):171–185. <https://doi.org/10.1016/j.jmb.2012.11.009>
37. Azoitei ML et al (2011) Computation-guided backbone grafting of a discontinuous motif onto a protein scaffold. *Science* 334(6054):373–376. <https://doi.org/10.1126/science.1209368>
38. Azoitei ML et al (2012) Computational design of high-affinity epitope scaffolds by backbone grafting of a linear epitope. *J Mol Biol* 415(1):175–192. <https://doi.org/10.1016/j.jmb.2011.10.003>
39. Liu S et al (2007) Nonnatural protein-protein interaction-pair design by key residues grafting. *Proc Natl Acad Sci* 104(13):5330–5335. <https://doi.org/10.1073/pnas.0606198104>
40. Potapov V et al (2008) Computational redesign of a protein-protein interface for high affinity and binding specificity using modular architecture and naturally occurring template fragments. *J Mol Biol* 384(1):109–119. <https://doi.org/10.1016/j.jmb.2008.08.078>
41. Wu Z et al (2020) Signal peptides generated by attention-based neural networks. *ACS Synth Biol* 9(8):2154–2161. <https://doi.org/10.1021/acssynbio.0c00219>
42. Saka K et al (2021) Antibody design using LSTM based deep generative model from phage display library for affinity maturation. *Sci Rep* 11(1):1–13. <https://doi.org/10.1038/s41598-021-85274-7>
43. Kang Y, Leng D, Guo J, Pan L (2021) Sequence-based deep learning antibody design for in silico antibody affinity maturation. arXiv preprint [arXiv:2103.03724](https://arxiv.org/abs/2103.03724). <https://doi.org/10.48550/arXiv.2103.03724>
44. Shin J-E et al (2021) Protein design and variant prediction using autoregressive generative models. *Nat Commun* 12(1):1–11. <https://doi.org/10.1038/s41467-021-22732-w>
45. Qi Y, Zhang JZ (2020) DenseCPD: improving the accuracy of neural-network-based computational protein sequence design with DenseNet. *J Chem Inf Model* 60(3):1245–1252. <https://doi.org/10.1021/acs.jcim.0c00043>
46. Wang J, Cao H, Zhang JZ, Qi Y (2018) Computational protein design with deep learning neural networks. *Sci Rep* 8(1):1–9. <https://doi.org/10.1038/s41598-018-24760-x>
47. Hawkins-Hooker A et al (2021) Generating functional protein variants with variational autoencoders. *PLoS Comput Biol* 17(2):e1008736. <https://doi.org/10.1371/journal.pcbi.1008736>
48. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Rama, C., Anthony, H. & Zhengyou, Z. (eds) Densely connected convolutional networks. (eds Rama, C., Anthony, H. & Zhengyou, Z.) *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708
49. Huang Z et al (2020) Application of innovative image processing methods and AdaBound-SE-DenseNet to optimize the diagnosis performance of meningiomas and gliomas. *Biomed Signal Process Control* 59:101926. <https://doi.org/10.1016/j.bspc.2020.101926>
50. Rafi AM et al (2019) Sanja, F. & Andrea, V. (eds) *Application of DenseNet in Camera Model Identification and Post-processing Detection*. (eds Sanja, F. & Andrea, V.) *CVPR workshops*, 19–28
51. Hasan N, Bao Y, Shawon A, Huang Y (2021) DenseNet convolutional neural networks application for predicting COVID-19 using CT image. *SN Comput Sci* 2(5):1–11. <https://doi.org/10.1007/s42979-021-00782-7>
52. Sandhya S et al (2009) Length variations amongst protein domain superfamilies and consequences on structure and function. *PLoS One* 4(3):e4981. <https://doi.org/10.1371/journal.pone.0004981>
53. Le Q, Miralles L, Kulkarni S, Su J, Boydell O (2020) An overview of deep learning in industry. Auerbach Publications, pp 65–98
54. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
55. Frenzel A, Schirrmann T, Hust M (2016) Phage display-derived human antibodies in clinical development and therapy. *MAbs* 8(7):1177–1194
56. Larsson L-I (2020) Immunocytochemistry: theory and practice. CRC Press
57. Szklarczyk D et al (2019) STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res* 47(D1):D607–D613. <https://doi.org/10.1093/nar/gky1131>
58. Moritz N, Hori T, Le Roux J (2021) Dimitri, A., Kostas, P. & Zhang, X.-P. (eds) *Capturing Multi-resolution Context by Dilated Self-attention*. (eds Dimitri, A., Kostas, P. & Zhang, X.-P.) *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5869–5873
59. Von Mering C et al (2005) STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res* 33:D433–D437. <https://doi.org/10.1093/nar/gki005>
60. Mazandu GK, Mulder NJ (2011) Scoring protein relationships in functional interaction networks predicted from sequence data. *PLoS One* 6(4):e18607. <https://doi.org/10.1371/journal.pone.0018607>
61. Tran L, Hamp T, Rost B (2018) ProfPPIdb: pairs of physical protein-protein interactions predicted for entire proteomes. *PLoS One* 13(7):e0199988. <https://doi.org/10.1371/journal.pone.0199988>
62. Bozhilova LV, Whitmore AV, Wray J, Reinert G, Deane CM (2019) Measuring rank robustness in scored protein interaction networks.

- BMC Bioinform 20(1):1–14. <https://doi.org/10.1186/s12859-019-3036-6>
63. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. MIT Press, Cambridge, pp 318–362
 64. Qin Y, Ding J, Sun Y, Ding X (2021) Teddy, M., Minh, L., Media, A., Wong, K. W. & Hidayanto, A. N. (eds) *A Transformer-based Model for Low-resource Event Detection*. (eds Teddy, M., Minh, L., Media, A., Wong, K. W. & Hidayanto, A. N.) *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, Proceedings, Part IV 28*, 452–463. Springer (Springer-Verlag, Berlin, Heidelberg, 2021)
 65. Wu J, Paquet E, Viktor HL, Michalowski W (2021) Tianzi, J. & Xin, Y. (eds) *Paying Attention: Using a Siamese Pyramid Network for the Prediction of Protein-Protein Interactions with Folding and Self-Binding Primary Sequences*. (eds Tianzi, J. & Xin, Y.) *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (IEEE, 2021)
 66. He K, Zhang X, Ren S, Sun J (2016) Tinne, T., Fei-Fei, L. & Ruzena, B. (eds) *Deep Residual Learning for Image Recognition*. (eds Tinne, T., Fei-Fei, L. & Ruzena, B.) *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (IEEE, 2016)
 67. Li H, Xu Z, Taylor G, Studer C, Goldstein T (2017) Visualizing the Loss Landscape of Neural Nets. arXiv preprint [arXiv:1712.09913](https://arxiv.org/abs/1712.09913). <https://doi.org/10.48550/arXiv.1712.09913>
 68. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Jill, B., Christy, D. & Thamar, S. (eds) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. (eds Jill, B., Christy, D. & Thamar, S.) *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019)
 69. Bojar O et al. (2014) Christian, B. & Christian, F. (eds) *Findings of the 2014 Workshop on Statistical Machine Translation*. (eds Christian, B. & Christian, F.) *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 12–58
 70. Xiang Z, Yan J, Demir I (2020) A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resour Res*. <https://doi.org/10.1029/2019WR025326>
 71. Cui Y, Yin B, Li R, Du Z, Ding M (2020) Xu, B. (ed.) *Short-time Series Load Forecasting by Seq2seq-LSTM Model*. (ed. Xu, B.) *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Vol. 9, 517–521 (IEEE, 2020)
 72. Li Y, Zhu M, Ma Y, Yang J (2020) Work modes recognition and boundary identification of MFR pulse sequences with a hierarchical seq2seq LSTM. *IET Radar Sonar Navig* 14(9):1343–1353. <https://doi.org/10.1049/iet-rsn.2020.0060>
 73. Karimi M, Wu D, Wang Z, Shen Y (2019) DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* 35(18):3329–3338. <https://doi.org/10.1093/bioinformatics/btz111>
 74. Kawano K, Koide S, Imamura C (2019) Seq2seq fingerprint with byte-pair encoding for predicting changes in protein stability upon single point mutation. *IEEE/ACM Trans Comput Biol Bioinform* 17(5):1762–1772. <https://doi.org/10.1109/TCBB.2019.2908641>
 75. Berman DS, Howser C, Mehoke T, Evans JD (2020) MutaGAN: a Seq2seq GAN framework to predict mutations of evolving protein populations. [arXiv:2008.11790](https://arxiv.org/abs/2008.11790)
 76. Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. *Neural Comput* 1(2):270–280
 77. Kingma DP, Ba J (2015) Bengio, Y. & LeCun, Y. (eds) *Adam: A Method for Stochastic Optimization*. (eds Bengio, Y. & LeCun, Y.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, 1–13
 78. Murphy KP (2013) Machine learning: a probabilistic perspective. (Adaptive Computation and Machine Learning Series). MIT Press, Cambridge
 79. Goodfellow I, Bengio Y, Courville A (2017) Deep learning. MIT Press
 80. Jungnickel D (1999) The greedy algorithm. Springer, Berlin Heidelberg, Heidelberg, pp 129–153
 81. Lowerre B, Reddy R (1976) The harpy speech recognition system: performance with large vocabularies. *J Acoust Soc Am* 60(S1):S10–S11. <https://doi.org/10.1121/1.2003089>
 82. Xia T, Wang Y, Tian Y, Chang Y (2021) Jure, L., Marko, G., Marc, N., Jie, T. & Leila, Z. (eds) *Using Prior Knowledge to Guide BERT's Attention in Semantic Textual Matching Tasks*. (eds Jure, L., Marko, G., Marc, N., Jie, T. & Leila, Z.) *Proceedings of the Web Conference 2021*, 2466–2475 (Association for Computing Machinery, New York, NY, USA, 2021)
 83. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Pierre, I. (ed.) *BLEU: a method for automatic evaluation of machine translation*. (ed. Pierre, I.) *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318
 84. Xia X, Xie Z (2002) Protein structure, neighbor effect, and a new index of amino acid dissimilarities. *Mol Biol Evol* 19(1):58–67. <https://doi.org/10.1093/oxfordjournals.molbev.a003982>
 85. Liu S, Xiang X, Gao X, Liu H (2020) Neighborhood preference of amino acids in protein structures and its applications in protein structure assessment. *Sci Rep* 10(1):1–11. <https://doi.org/10.1038/s41598-020-61205-w>
 86. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 48(3):443–453. [https://doi.org/10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4)
 87. Smith TF, Waterman MS et al (1981) Identification of common molecular subsequences. *J Mol Biol* 147(1):195–197. [https://doi.org/10.1016/0022-2836\(81\)90087-5](https://doi.org/10.1016/0022-2836(81)90087-5)
 88. Banerjee A, Lavie E (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. *In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and Summarization*, 65–72
 89. Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. *Neural Comput* 1(2):270–280. <https://doi.org/10.1162/neco.1989.1.2.270>
 90. Park Y, Marcotte E (2012) Flaws in evaluation schemes for pair-input computational predictions. *Nat Methods* 2:1134–1136
 91. Hamp T, Burkland R (2015) More challenges for machine-learning protein interactions. *Bioinformatics* 31(10):1521–1525
 92. Goodfellow I et al. (2014) Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. (eds) *Generative Adversarial Nets*. (eds Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K.) *Advances in Neural Information Processing Systems*, Vol. 27, 2672–2680 (Curran Associates, Inc., 2014)
 93. Wang S, Li B, Khabsa M, Fang H, Ma H (2020) Linformer: Self-attention with linear complexity. [arXiv:2006.04768](https://arxiv.org/abs/2006.04768), 1–12
 94. Liu J, Pan Z, He H, Cai J, Zhuang B (2022) EcoFormer: energy-saving attention with linear complexity. *In Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS)*, 1–14

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.