



OPEN ACCESS

EDITED BY

Mohammad Akbari,
Amirkabir University of Technology, Iran

REVIEWED BY

Guohua Huang,
Shaoyang University, China
Wasiq Khan,
Liverpool John Moores University,
United Kingdom

*CORRESPONDENCE

Tian Xu
✉ tianxu66@vt.edu

†These authors have contributed equally to this work and share first authorship

RECEIVED 22 January 2023

ACCEPTED 24 July 2023

PUBLISHED 15 August 2023

CITATION

Xu T, Xu Q and Li J (2023) Toward the appropriate interpretation of Alphafold2. *Front. Artif. Intell.* 6:1149748. doi: 10.3389/frai.2023.1149748

COPYRIGHT

© 2023 Xu, Xu and Li. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Toward the appropriate interpretation of Alphafold2

Tian Xu^{1*†}, Qin Xu^{2†} and Jianyong Li¹

¹Department of Biochemistry, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, ²Department of Mathematics, The University of Arizona, Tucson, AZ, United States

In life science, protein is an essential building block for life forms and a crucial catalyst for metabolic reactions in organisms. The structures of protein depend on an infinity of amino acid residues' complex combinations determined by gene expression. Predicting protein folding structures has been a tedious problem in the past seven decades but, due to robust development of artificial intelligence, astonishing progress has been made. Alphafold2, whose key component is Evoformer, is a typical and successful example of such progress. This article attempts to not only isolate and dissect every detail of Evoformer, but also raise some ideas for potential improvement.

KEYWORDS

deep learning, machine learning, Alphafold2, protein, protein folding

1. Introduction

The most common approach to obtain and elucidate protein structures in biological science is x-ray crystallography, by which three-dimensional protein structures are derived based on x-ray diffraction data of individual amino acid residues (Smyth and Martin, 2000). Another useful tool to derive protein structure is nuclear magnetic resonance (NMR) spectroscopy (Wüthrich, 1990), by which one can obtain dynamic changes of the protein structures under specific conditions. Both approaches often lead to accurate structural solutions of proteins of interest. In addition, protein NMR analysis could provide dynamic changes of a given protein under specific solvent conditions. However, obtaining structural information is extremely time consuming and expensive with either method. Issues related to obtaining protein samples with adequate purity also present themselves in protein crystallography and NMR analysis. For example, obtaining crystals that diffract at high resolution once protein samples are obtained is a challenge in protein crystallography. In protein NMR analysis, the size and solubility of given protein samples may lead to additional problems and may require more sophisticated NMR equipment. Some membrane proteins are difficult to crystallize, if not impossible.

Despite the painstaking nature of structural determination (by crystallography or NMR), structural determination is essential, providing structural information for a deep understanding of individual protein functions and numerous reliable models for comparison and contrast. The availability of a substantial number of protein structures is essential, particularly through the crystallographic approach. This requirement is based on the need for a solid foundation in understanding protein functions and interactions at a molecular level. Achieving these structures through crystallography demands a high degree of accuracy during the crystallization process. A known downside of NMR spectroscopy is the huge number of purified samples needed. Managing the purifying process for large amounts of protein is challenging in some scenarios. There is a long history of protein folding prediction via computational methods. In 1995, the Critical Assessment of protein Structure Prediction (CASP) was founded. Held every 2 years, CASP provides an opportunity for scientists and engineers to test their mathematical models. However, despite decades of work, people have still not found a model to predict protein folding that is close to the ground truth.

In 2020, AlphaFold2 (Jumper et al., 2020) won the CASP14 for achieving the most accurate prediction of protein-folding with less than the diameter of one carbon atom confidence error in average (<1Å in average). This was the best result made for that time, and it vastly outperformed other competitors' results. The competitor with the highest scores obtained a summed z score of 244.0217 and an average Z score of 2.6524. In comparison, the closest competitor achieved a lower summed z score of 92.1241 and an average Z score of 1.0013.¹ This result was exhilarating since it was the first-time humans had begun getting truly close to the ground truth. Moreover, scientists have started applying AlphaFold2 in many sub-fields of biology. For example, Ren et al. applied AlphaFold2 in drug discovery. As the first reported small molecule targeting CDK20, out of the seven compounds that were synthesized, ISM042-2-001 exhibited a Kd value of $9.2 \pm 0.5 \mu\text{M}$ ($n = 3$) in the CDK20 kinase binding assay (Ren et al., 2022). In the work by Zhang et al. (2020, 2021), they even extended AlphaFold2 by combining Prod Conn. AlphaFold2, and sequential Monte Carlo (Doucet et al., 2001) to predict engineered unstable protein structures. Their findings indicate that the representations derived from AlphaFold2 models can effectively forecast the stability alterations caused by point mutations. Furthermore, they observed that AlphaFold accurately predicted the ProDCoNN-designed sequences, which exhibited varying root-mean-square deviations (RMSDs) in relation to the target structures. This suggests that certain sequences possess higher foldability compared to others (Zhang et al., 2021). There are more examples in Table 1. Since the major components of Evoformer (Jumper et al., 2020) (row and column attention, triangular self-attention) (Vaswani et al., 2017) are all built upon the transformer model (Vaswani et al., 2017), it is important to have a good grasp of the transformer model (Vaswani et al., 2017). In this article, we will highlight the structure of the Evoformer (Jumper et al., 2020) and how it works from a basic attention mechanism (Vaswani et al., 2017), for instance, the math behind it and what the attention score really is. We will also discuss how Evoformer relates to transformer (Vaswani et al., 2017) and the difference between them, especially when message passing based on graph theory (Gilmer et al., 2020) is blended in. Despite its strengths, AlphaFold2 shows reduced accuracy when predicting very long sequences. The root mean square deviation (RMSD) of longer sequence prediction grows quickly. A potential improvement for this weakness is provided in the discussion section.

2. Methodology

Many machine learning techniques have been used in AlphaFold2, including Evoformer (Jumper et al., 2020) and dataset distillation (Wang et al., 2018). In this article, we will mainly focus on the Evoformer, a modification of the transformer (Vaswani et al., 2017), because Evoformer is the most crucial component of AlphaFold2.

Evoformer is a derivative of the transformer (Vaswani et al., 2017), and it improves attention modules to fit many proteins' specific aspects. In the transformer model (Vaswani et al., 2017)

TABLE 1 Applications of AlphaFold2 in different areas.

	Area of study	Achievement
Novel fold of rotavirus glycan-binding domain predicted by AlphaFold2 and determined by X-ray crystallography (Hu L. et al., 2022)	Structural biology	AlphaFold2 successfully predicts the structure of the VP8* domain (VP8*B) of VP4 and the result is verified by experiment.
AlphaFold accelerates artificial intelligence powered drug discovery: efficient discovery of a novel Cyclin-Dependent Kinase 20 (CDK20) small molecule inhibitor (Ren et al., 2022)	Drug discovery	AlphaFold2 successfully predicts the structure of the chemical compound which demonstrates it is helpful in the early stage of drug discovery.
De novo protein design by inversion of the AlphaFold structure prediction network (Goverde et al., 2023)	Protein design	AlphaFold2 shows the potential capability to solve the challenge in de novo protein design.
CavitySpace: a database of potential ligand binding sites in the human proteome (Wang et al., 2022)	Target prediction	AlphaFold2 is used to create the database CavitySpace which is the first library of human proteome.
Exploring evolution-based and-free protein language models as protein function predictors (Hu M. et al., 2022)	Protein function prediction	Evolution-based evoformer and evolution-free evoformer are compared based on alphaFold2.
Improved prediction of protein-protein interactions using AlphaFold2 (Bryant et al., 2022)	Protein-protein interaction	AlphaFold2 based docking methods perform better than other docking methods.
Identification of a novel substrate motif of yeast separase and deciphering the recognition specificity using AlphaFold2 and molecular dynamics simulation (Liang et al., 2022)	Biological mechanism of action	AlphaFold2 helps scientists with deeper understanding of mechanism of substrate recognition and activation of separase.

for Image Recognition at Scale (ViT) (Dosovitskiy et al., 2020) or Natural Language Processing (NLP), embedding vectors are not correlated to each other. However, from an Evoformer perspective, all the rows and columns are somehow interconnected. So, to solve the interconnections in that scenario, Evoformer introduced row-wise and column-wise attention, which have practical meanings in homology.

One of the inputs of Evoformer is Multiple Sequence Alignment (MSA) (Corpet, 1988), which is composed of human amino acids sequences and amino acid sequences from other animals that are highly similar or identical to humans. Row attention tracks the amino acids' correlations among different species and column attention compares the interconnections among amino acids sequences among homologous species.

Another important input of Evoformer is amino acids pairs, which allow protein-to-protein information transfer (Root-Bernstein, 1982). One of the major problems of sequential data machine learning in a Euclidean space is it does not sustain the relational connectivity for neighborhood elements, especially when we assume relational connectivity does not have connection, but it does. The consequence of this is the model will lose tons of

1 https://predictioncenter.org/casp14/zscores_final.cgi

information to induce the result. To maintain the neighborhood, Evoformer aggregates information across amino acids pairs by using Triangular multiplicative updates and Triangular self-attention in analogy to passing information in GNN (Gilmer et al., 2020). In this article, we will mainly focus on the structure analysis of Evoformer and the comparison between the traditional transformer (Vaswani et al., 2017) and Evoformer.

2.1. Linear projection

In linear algebra, a linear projection P is a linear function to a vector space that, when applied to itself, elicits the same result. This definition of linear projection is confusing because it is too abstract to imagine. In machine learning, linear projection is used to project a target vector onto higher or lower dimensional subspace and simply align different vectors' dimensions. Perceptron (Marsalli, 2008) is a typical linear projection usage.

In transformer models (Vaswani et al., 2017), including Evoformer, embedding vectors are projected into query, key, and value vectors. Assume input $X = \mathbb{R}^{m \times n}$, $W_K, W_Q = \mathbb{R}^{n \times d_k}$ and $W_V = \mathbb{R}^{n \times d_l}$

$$Q = X \cdot W_Q \tag{1}$$

$$K = X \cdot W_K \tag{2}$$

$$V = X \cdot W_V \tag{3}$$

where $Q, K \in \mathbb{R}^{m \times d_k}$ and $V \in \mathbb{R}^{m \times d_l}$. Q, K, and V stand for query, key and value. $W_Q, W_K,$ and W_V are learnable parameters. The intuition of why input X is projected to three different outputs is from a retrieval system. A query is like your search input in Google; Google finds your expected value based on the key words of your search.

2.2. Attention

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K}{\sqrt{d_k}}\right) \cdot V \tag{4}$$

Attention mechanisms can be analogous to humans capturing crucial information in sentences. The transformer model (Vaswani et al., 2017) utilizes dot-product attention instead of additive attention for efficient storage and faster performance (Figure 1). To understand why a dot product works in attention, we need to look inside the dot product.

Let's assume two vectors u and v in \mathbb{R}^n :

$$u \cdot v = ||u|| \cdot ||v|| \cdot \cos \theta_{uv} = \sum_{i=1}^n u_i \cdot v_i \tag{5}$$

The Euclidean distance between vectors u and v :

$$||u - v|| = \sqrt{||u||^2 + ||v||^2 - 2 \cdot ||u|| \cdot ||v|| \cdot \cos \theta_{uv}} = \sqrt{||u||^2 + ||v||^2 - 2 \cdot u \cdot v} \tag{6}$$

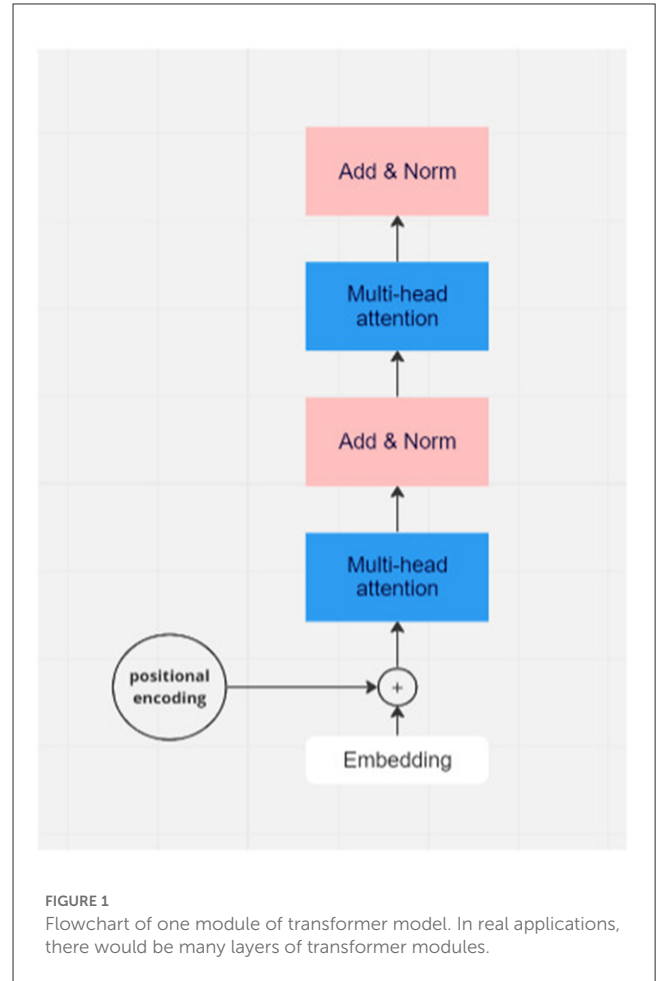
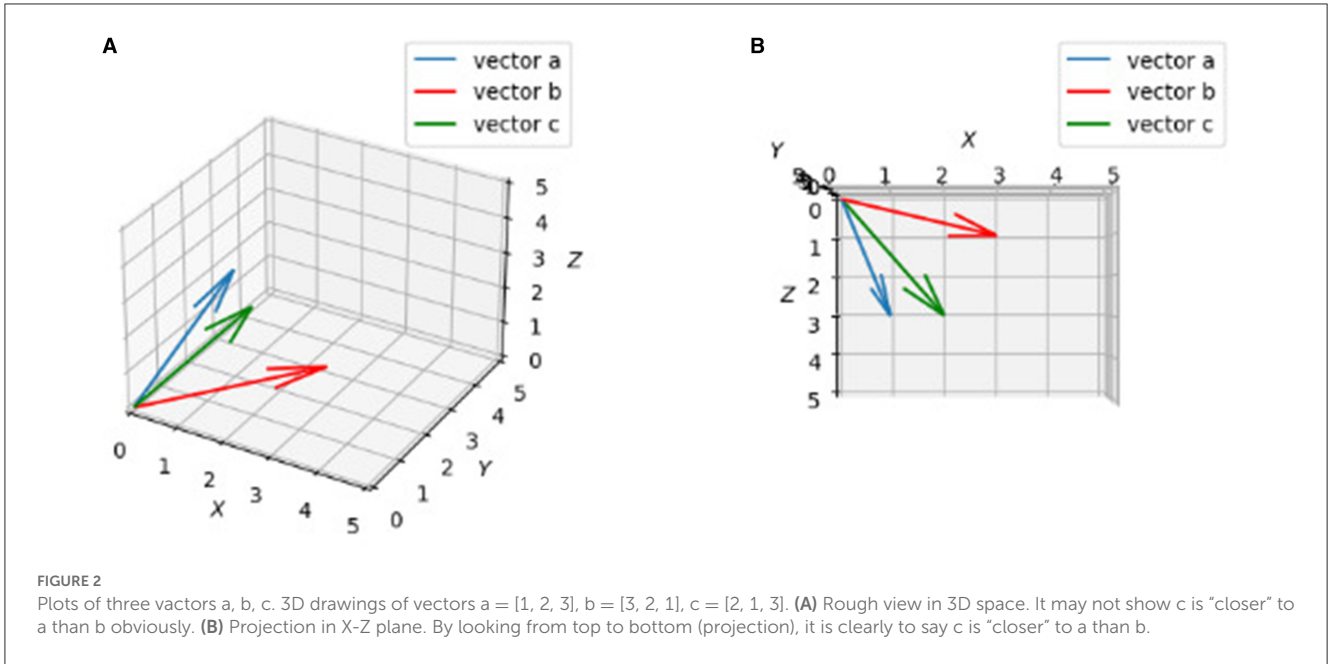


FIGURE 1 Flowchart of one module of transformer model. In real applications, there would be many layers of transformer modules.

In general, if two vectors are close, their Euclidean distance needs to be small according to equation 6, which means the value from equation 5 needs to be big enough. In this manner, one can say the bigger the dot product value, the more similar the two vectors are. Although we cannot draw diagrams in dimensions higher than three, we can show what vectors in 3d looks like in terms of similarity (Figure 2):

$$\begin{aligned} a &= [1 \ 2 \ 3], \quad b = [3 \ 2 \ 1], \quad c = [2 \ 1 \ 3] \\ ||a|| &= ||b|| = ||c|| = \sqrt{14} \\ a^T \cdot b &= 1 \cdot 3 + 2 \cdot 2 + 3 \cdot 1 = 10 \\ a^T \cdot c &= 1 \cdot 2 + 2 \cdot 1 + 3 \cdot 3 = 13 \\ ||a - b|| &= \sqrt{28 - 2 \cdot 10} = \sqrt{8} \\ ||a - c|| &= \sqrt{28 - 2 \cdot 13} = \sqrt{2} \\ ||a - b|| &> ||a - c|| \end{aligned}$$

This is exactly how self-attention works. The scalar value calculated from the dot product of two vectors is represented as scores of similarities. In equation 4, Q is a matrix and K is also a matrix. To calculate dot products of each row between matrix Q and K, matrix multiplication is required:



$$Q \cdot K^T = q_1 \cdot k_1 + q_2 \cdot k_2 + \dots + q_i \cdot k_i = \sum_{i=1}^m q_i \cdot k_i$$

The problem of dot-product attention is it may get either too large or too small depending on the d_k . To see that, assume all q_i and k_i are random independent variables and have a distribution with mean 0 and variance 1

$$\begin{aligned} Cov(q_i, k_i) &= E[(q_i - E[q_i])(k_i - E[k_i])] \\ &= E[q_i \cdot k_i] - E[q_i] \cdot E[k_i] = 0 \end{aligned} \tag{7}$$

$$\begin{aligned} Var(q_i) &= E[q_i^2] - E[q_i]^2 \\ &= E[q_i^2] \\ &= 1 \end{aligned} \tag{8}$$

$$Var(k_i) = 1 \tag{9}$$

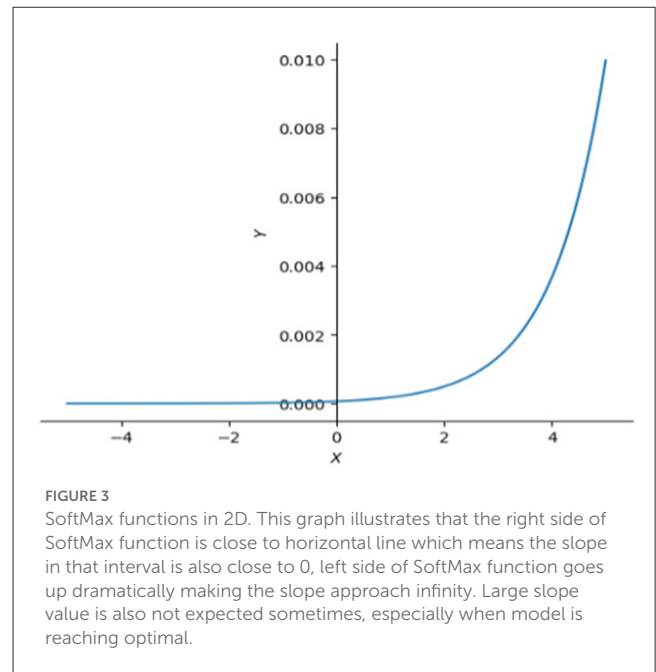
So, based on equations 8 and 9, we have:

$$\begin{aligned} Var(q_i k_i) &= E[(q_i k_i)^2] - E[q_i k_i]^2 \\ &= E[q_i^2] \cdot E[k_i^2] - (E[q_i] \cdot E[k_i])^2 = 1 \end{aligned} \tag{10}$$

and k_i is a random independent variable and based on equation 7, 10:

$$E[\mathbf{q} \cdot \mathbf{k}] = \sum_{i=1}^{d_k} E[q_i \cdot k_i] = 0 \tag{11}$$

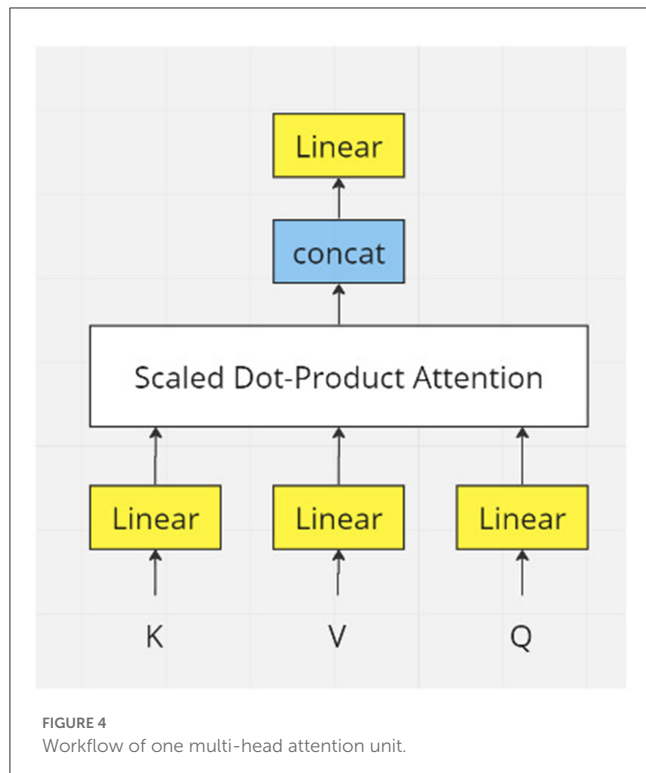
$$Var(\mathbf{q} \cdot \mathbf{k}) = \sum_{i=1}^{d_k} Var(q_i \cdot k_i) = d_k \tag{12}$$



$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{13}$$

If d_k is large enough, \mathbf{qk} may be much larger or smaller than mean 0.

This will make the SoftMax (equation 13) activation function in equation 4 have an unstable or tiny slope (Figure 3), and make the gradient at the backpropagation stage explode or vanish. To avoid that happening, we need to clip it by scaling down the scores by their standard deviation ($\sqrt{d_k}$) (equation 12).



$$a_i = \text{Softmax}\left(\frac{q_i \cdot k_i}{\sqrt{d_k}}\right) \tag{14}$$

$$\text{attention}(q_i, k_i, v_i) = a_i \cdot v_i \tag{15}$$

2.3. Multi-head

In the transformer model (Vaswani et al., 2017), linear projection and attention have been done h ($h = 8$ in transformer model (Vaswani et al., 2017) times (Figure 4). So, equations 1, 2, and 3 become:

$$\begin{aligned} Q_i &= X \cdot W_{Q_i} \\ K_i &= X \cdot W_{K_i} \\ V_i &= X \cdot W_{V_i} \\ i &= 1, 2, 3, \dots, h \end{aligned} \tag{16}$$

equation 4 becomes:

$$\begin{aligned} \text{Head}_i &= \text{Attention}(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i \cdot K_i}{\sqrt{d_k}}\right) \cdot V_i \\ i &= 1, 2, 3, \dots, h \end{aligned} \tag{17}$$

To get all attention, we concatenate all the heads by:

$$\text{Heads} = \text{Concat}(\text{Head}_1, \text{Head}_2, \dots, \text{Head}_h) \tag{18}$$

Recalling equations 1, 2, and 3, the input dimension is $\mathbb{R}^{m \times n}$. The result of equation 18 has the dimension $\mathbb{R}^{m \times nh}$. To keep all attention layers consistent, we use a matrix $W_o = \mathbb{R}^{nh \times n}$ to learn a representation back to dimension $\mathbb{R}^{m \times n}$.

$$\text{MultiheadAttention}(Q, K, V) = \text{Heads} \cdot W_o \tag{19}$$

2.4. Residual

Theoretically, a deeper network should learn more representations than a shallow network. In practice, this is not true since gradients may vanish in deep layers. Intuitively, we want deep layers to perform at least as well as shallow layers, not worse. Resnet (He et al., 2016) introduced a way to solve that problem called ‘‘Identity shortcut connection.’’ If a model is already optimized, we have

$$F(x) := H(x) - x = 0 \tag{20}$$

Equation 20 means $H(x)$ is an identical mapping of x and optimal layers learn nothing. So, it makes sure that the shallow layers receive useful information(x) at least.

2.5. Layer normalization

$$x_i = \frac{x_i - x}{\delta} \tag{21}$$

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n \frac{x_i}{n} = \mu \text{ (law of large numbers)} \tag{22}$$

Layer normalization (Ba et al., 2016) is a substituent of batch normalization (Ioffe, 2015). It provides great help in two areas of batch normalization: small batch sizes of samples and different input lengths in dynamic networks. Both areas mean batch normalization cannot represent the real distribution of data (equation 21 and 22). Instead of normalizing across the batch axis, layer normalization normalizes the channel or embedding axis to avoid such problems (Figure 5).

3. Evoformer

With previous knowledge of the transformer model (Vaswani et al., 2017), we can continue with Evoformer, which takes some advantages from the transformer model, but also provides improvement to fit specific protein attributes. In this section, we give a brief introduction of how Evoformer works and is related to the transformer model.

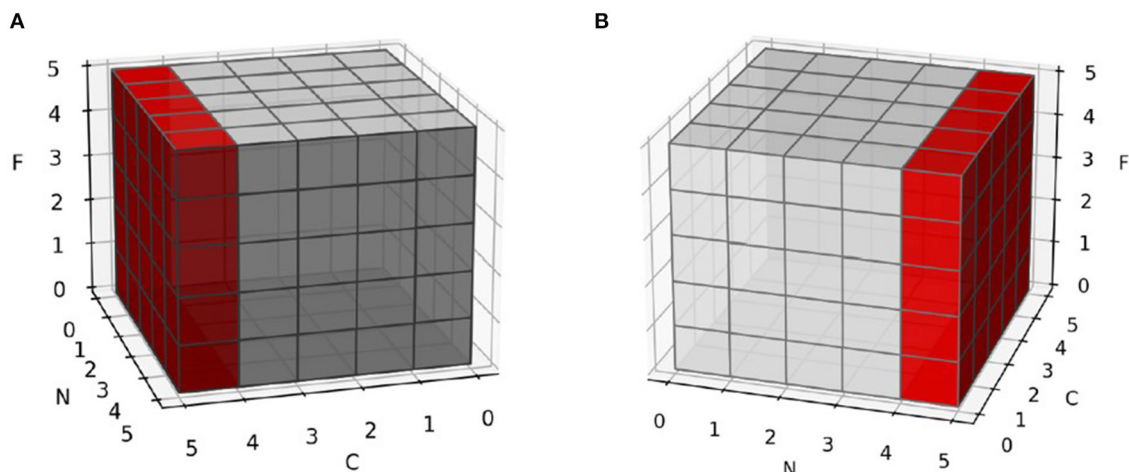


FIGURE 5
 Normalization in different axis. C is channel or embedding axis, N is batch axis and F is feature axis. (A) Normalize samples in batch axis. Each feature is not normalized in sample-wise, but in batch-wise. (B) Normalize samples in their channel or embedding axis. Each feature is not normalized in batch-wise, but in sample-wise.

3.1. Row and column attention

Like equation 1, the transformer model (Vaswani et al., 2017) is often used in NLP missions. The batched inputs of NLP are word-token sequences where they are not related at all if random sampling is performed. For instance, the first sentence is “AlphaFold is a great tool,” and the next sentence is “Panda is one of the rare animals,” but that is not the case in Evoformer. The “word-tokens” from input layers are homologous to each other among rows and columns. Row-wise sequence alignment could represent the similarity or difference of gene expression across different species and column-wise sequence alignment embeds the information of how gene expression evolves (Zhang et al., 2022). For such reasons, Evoformer is used for how to learn weight distribution across rows and columns. Supplementary Figure 2 from the original paper (Jumper et al., 2020) shows all the steps of row-wise self-attention and looks like a traditional multi head self-attention in transformer model with several modifications. One can notice that queries (r_q, h, c) , keys (r_v, h, c) , values (r_v, h, c) , and gate (r_q, h, c) are all linearly transformed from a specific specie (row-wise). To calculate attention weight, pair bias is added at the end before SoftMax activation is performed. Moreover, gate g is used to filter value representation.

For regular self-attention (equation 4) in the transformer model, there is no bias terms since it only has one source of input (e.g., sentences, pictures). However, Evoformer has two totally distinct types of input: MSA and amino acids pairs. Just like a linear function, for instance $y = x \cdot w + b$, according to algorithm 7 from Supplementary material of the paper (Jumper et al., 2020), if we treat $\frac{q_{st}^{h^T} \cdot k_{sj}^h}{\sqrt{c}}$ as xw and b_{ij}^h as b , then b_{ij}^h shifts the attention score to fit the actual score better by utilizing the information in pairs. The sub-index of each tensor is a little bit tricky. So be cautious when aligning the proper b_{ij}^h to $\frac{q_{st}^{h^T} \cdot k_{sj}^h}{\sqrt{c}}$.

As mentioned above, column-wise attention is a new technique for specific protein problems. It captures attention across MSA columns. Columns in MSA represent the same gene expression among different species. Supplementary Figure 3 (Jumper et al., 2020) from the original paper shows the details of column-wise attention. Queries (s_q, h, c) , keys (s_v, h, c) , values (s_v, h, c) , and gate (s_q, h, c) are all linearly transformed from a specific gene expression (column-wise). There is no pair bias anymore so that column-wise attention looks almost the same as the transformer’s self-attention except it has a gate.

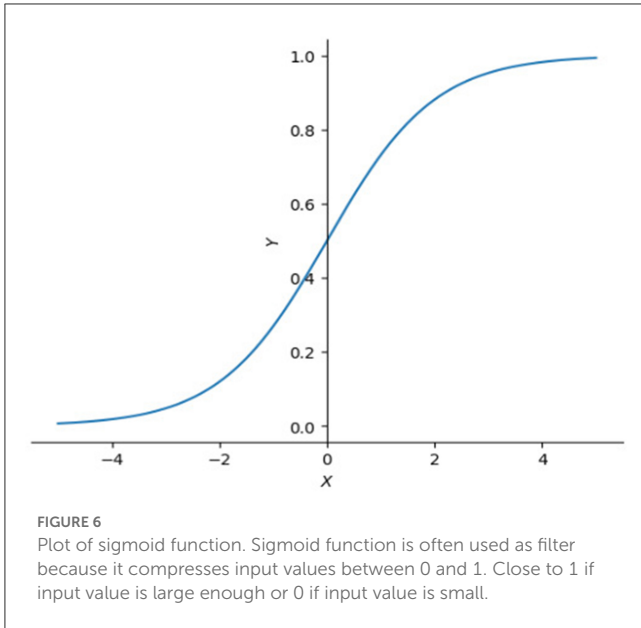
The idea of a gate has a relatively long history. Long short-term memory (LSTM) (Hochreiter, 1997) introduced a few kinds of gate cell to improve RNN (Medsker, 2001). A gate cell can be seen as a filter which only allows values with high confidence to pass because of the nature of sigmoid function (Figure 6).

3.2. MSA transition

MSA transition is equivalent to a feed forward layer in the transformer model (Vaswani et al., 2017) which has most of the trainable parameters and key-value memories (Geva et al., 2020).

3.3. Outer product mean

As in Figure 5A of the paper (Jumper et al., 2020), the result coming from the transition layer needs to be added back to pair representation. However, MSA has a shape (s, r, c_m) and pairing representation has a shape (r, r, c_z) . To add them up, MSA must be reshaped to the same shape as the pairing representation [Supplementary Figure 5 of the paper (Jumper et al., 2020)]. The outer product is a vector version of the Kronecker product of matrices. The most straightforward effect of the outer product is



that the outer product of two vectors is a matrix. In other words, an extra dimension is produced. Here is an example:

Assume vectors \mathbf{u} and $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{u} \otimes \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} \otimes [v_1 \ v_2 \ v_3 \ \cdots \ v_n] = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & u_2 v_3 & \cdots & u_2 v_n \\ u_3 v_1 & u_3 v_2 & u_3 v_3 & \cdots & u_3 v_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_n v_1 & u_n v_2 & u_n v_3 & \cdots & u_n v_n \end{bmatrix} \quad (23)$$

To update pair representation p_{ij} , it calculates the outer product of i th and j th columns from MSA representation, then takes the mean value along the new axis (first dimension s) and finally maps to pair representation p_{ij} with a linear transformation.

There are multiple ways to expand dimensions in this case. For example, we can perform a matrix tile operation on the third dimension. The reason to choose the outer product here is it condenses all the information in MSA in order to reconstruct pair representation.

3.4. Triangular multiplicative update

The triangular multiplicative update model is analogous to nodes in graph theory. To update the information contained at position $\{i, j\}$ by receiving all information, its neighbor nodes $\{i, k\}$ and $\{j, k\}$ pass according to Supplementary Figure 6 of the paper (Jumper et al., 2020). There are two symmetric versions for updating z_{ij} (Jumper et al., 2020). One refers to the “outgoing” edge

(row wise) and another is the “incoming” edge (column wise). In the “outgoing” edge version, every z_{ij} is updated by the sum of all columns of the i th row and j th row. In the “incoming” edge version, every z_{ij} gets updated by the sum of all rows of the i th column and j th column. The algorithm looks like row- or column-wise attention, but the dot product is replaced by elementwise multiplication for a cheaper computational cost (equation 24).

3.5. Triangular self-attention

Triangular self-attention looks almost exactly the same as row and column attention [algorithm 11 and 12 from the paper (Jumper et al., 2020)] and there are also two symmetric versions (Jumper et al., 2020). In the starting node version, the key \mathbf{k}_{ij} and value \mathbf{v}_{ij} are replaced by \mathbf{k}_{ik} and \mathbf{v}_{ik} (key and value from i th row and k th column), b_{jk} is also created and added to inner product affinities as a bias term. In the ending node version, the key \mathbf{k}_{ij} and value \mathbf{v}_{ij} are replaced by \mathbf{k}_{ki} and \mathbf{v}_{ki} (key and value from i th column and k th row); b_{ki} is also created and added to inner product affinities as a bias term. Now, updating z_{ij} not only depends on the dot product similarities of neighbor gene expression but also depends on the third edge b_{ik} and b_{kj} (Jumper et al., 2020).

4. Discussion

Although Alphafold2 can make highly accurate predictions of protein structures, it is not perfect. There are some things the model ignores.

- (a) Alphafold2 is not designed to model co-factor-based protein folds. Myoglobin or hemoglobin need a heme to fold and zinc-finger domains are not stable without a zinc ion, so Alphafill is developed to address such issues (Hekkelman et al., 2023).
- (b) One major drawback of AF2, which has a significant impact on the field of biomedicine, is the inadequate quality of transmembrane protein models. Although the confidence indices for transmembrane segments can be reliable, the overall predicted topology of these models is incompatible with their insertion into a membrane bilayer (Tourlet et al., 2023).
- (c) An inherent constraint of MSA (Multiple Sequence Alignments)-based methods like AlphaFold2 is their reliance on existing knowledge and datasets. While these methods can make educated estimates between known protein structures and potentially even make predictions around those known structures, they struggle to confidently and precisely predict entirely new configurations. Incorporating a molecular dynamics component is crucial to effectively model these novel configurations (Marcu et al., 2022).

Considering the above limitations, the model should aim to incorporate the effects generated by co-factors. To improve the prediction results of poor-quality transmembrane proteins, it should consider how to enhance the accuracy

and reliability of their modeling. Additionally, the model should focus on generalizing to data not present in the existing databases, in order to improve and refine its predictions.

5. Conclusion

AlphaFold2 is a complicated system. There are more than 60 pages in the supplementary document of AlphaFold2 (Jumper et al., 2020). If we break it down to every step, hundreds of pages will be required. In this article, we only discussed the most important module-Evoformer. However, other modules also made huge contributions to its success, for instance, data preparation, IPA Module, and recycling. The argument is still controversial, but some discussion says AlphaFold2 is the future of protein structure prediction (Al-Janabi, 2022). However, AlphaFold2 provides us with a new way to study protein structure where we can combine experimental efforts with a powerful tool set. The AlphaFold2 model can be used to generate more specific drugs and find more suitable animals to test medicines (Thornton et al., 2021), improve structural coverage of the human proteome (Porta-Pardo et al., 2022), and so on. AlphaFold2 also shows its success in general sequential models other than NLP and image recognition tasks. At the beginning of the self-attention transformer model (Vaswani et al., 2017), most of the studies were based on NLP missions, for instance, “Bert”

(Devlin et al., 2018). Now, the transformer model is also popular in biology.

Author contributions

TX contributed to the design and conception of the study. QX contributed to all the details and mathematics of the study. JL contributed by editing and reviewing the manuscript. All authors contributed to the manuscript revision, read, and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Al-Janabi, A. (2022). Has deepmind’s alphafold solved the protein folding problem? *Future Sci.* 72, 73–76. doi: 10.2144/btn-2022-0007
- Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). *Layer Normalization*. arXiv preprint arXiv.
- Bryant, P., Pozzati, G., and Elofsson, A. (2022). Improved prediction of protein-protein interactions using AlphaFold2. *Nat. Commun.* 13, 1265. doi: 10.1038/s41467-022-28865-w
- Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res.* 22, 10881–10890. doi: 10.1093/nar/16.22.10881
- Devlin, J., Chang, M. -W., Lee, K., and Toutanova, K. (2018). Bert: pretraining of deep bidirectional transformers for language understanding. *arXiv [Preprint]*. arXiv: 1810.04805.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. *arXiv [Preprint]*. arXiv: 2010.11929 doi: 10.48550/arXiv.2010.11929
- Doucet, A., Freitas, N. D., and Gordon, N. (2001). *An Introduction to Sequential Monte Carlo Methods*. Sequential Monte Carlo Methods in Practice: Springer. doi: 10.1007/978-1-4757-3437-9_1
- Geva, M., Schuster, R., Berant, J., and Levy, O. (2020). Transformer feedforward layers are key-value memories. *arXiv preprint arXiv*. doi: 10.18653/v1/2021.emnlp-main.446
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2020). *Message Passing Neural Networks*. Machine learning meets quantum physics. Springer. doi: 10.1007/978-3-030-40245-7_10
- Goverde, C., Wolf, B., Khakzad, H., Rosset, S. and Correia, B. E. (2023). De novo protein design by inversion of the AlphaFold structure prediction network. *Protein Sci.* 32, e4653. doi: 10.1002/pro.4653
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision*. (Las Vegas, NV). doi: 10.1109/CVPR.2016.90
- Hekkelman, M. L., De Vries, I., Joosten, R. P., and Perrakis, A. (2023). AlphaFill: enriching AlphaFold models with ligands and cofactors. *Nat. Methods* 20, 205–213. doi: 10.1038/s41592-022-01685-y
- Hochreiter, S. A. S. (1997). Long short-term. *Memory.* 8, 1735–1780. doi: 10.1162/neco.1997.9.8.1735
- Hu, L., Salmen, W., Sankaran, B., Lasanajak, Y., Smith, D. F., Crawford, S. E., et al. (2022). Novel fold of rotavirus glycan-binding domain predicted by AlphaFold2 and determined by X-ray crystallography. *Commun. Biol.* 5, 419. doi: 10.1038/s42003-022-03357-1
- Hu, M., Yuan, F., Yang, K. K., Ju, F., Su, J., Wang, H., et al. (2022). Exploring evolution-based &-free protein language models as protein function predictors. *arXiv [Preprint]*. arXiv: 2206.06583.
- Ioffe, S. A. S. C. (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning* (New York, NY).
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Tunyasuvunakool, K., et al. (2020). *AlphaFold 2. Fourteenth Critical Assessment of Techniques for Protein Structure Prediction*. London: DeepMind.
- Liang, M., Chen, X., Zhu, C., Liang, X., Gao, Z., and Luo, S. (2022). Identification of a novel substrate motif of yeast separase and deciphering the recognition specificity using AlphaFold2 and molecular dynamics simulation. *Biochem. Biophys. Res. Commun.* 620, 173–179.
- Marcu, S. -B., Tăbircă, S., and Tangney, M. (2022). An overview of AlphaFold’s breakthrough. *Front. Artif. Intell.* 5:875587. doi: 10.3389/frai.2022.875587
- Marsalli, M. (2008). “McCulloch-pitts neurons,” in *The 2008 Annual Meeting of the consortium on cognitive science instruction (ccsi)* (Washington, DC), 1161, 1162.
- Mesdker, L. R. A. J. L. (2001). Recurrent neural networks. *Design Appl.* 64–67.
- Porta-Pardo, E., Ruiz-Serra, V., Valentini, S., and Valencia, A. (2022). The structural coverage of the human proteome before and after AlphaFold. *PLoS Comput. Biol.* 1:e1009818. doi: 10.1371/journal.pcbi.1009818

- Ren, F., Ding, X., Zheng, M., Korzinkin, M., Cai, X., Zhu, W., et al. (2022). AlphaFold accelerates artificial intelligence powered drug discovery: efficient discovery of a novel cyclin-dependent kinase 20 (CDK20) small molecule inhibitor. *arXiv preprint arXiv*. doi: 10.1039/D2SC05709C
- Root-Bernstein, R. S. (1982). Amino acid pairing. *J. Theor. Biol.* 94, 885–894. doi: 10.1016/0022-5193(82)90083-2
- Smyth, M. S., and Martin, J. H. J. (2000). x ray crystallography. *Mol. Pathol.* 1, 8. doi: 10.1136/mp.53.1.8
- Thornton, J. M., Laskowski, R. A., and Borkakoti, N. (2021). AlphaFold heralds a data-driven revolution in biology and medicine. *Nat. Med.* 10, 1666–1669. doi: 10.1038/s41591-021-01533-0
- Tourlet, S., Radjasandirane, R., Diharce, J., and De Brevern, A. G. (2023). AlphaFold2 Update and perspectives. *BioMedInform.* 3, 378–390. doi: 10.3390/biomedinformatics3020025
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. 30. doi: 10.48550/arXiv.1706.03762
- Wang, S., Lin, H., Huang, Z., He, Y., Deng, X., Xu, Y., et al. (2022). CavitySpace: a database of potential ligand binding sites in the human proteome. *Biomolecules*. 12, 967. doi: 10.3390/biom12070967
- Wang, T., Zhu, J. -Y., Torralba, A., and Efros, A. A. (2018). Dataset distillation. *arXiv [Preprint]*. arXiv:1811.10959. doi: 10.48550/arXiv.1811.10959
- Wüthrich, K. (1990). Protein structure determination in solution by NMR. *J. Biol. Chem.* 265, 22059–22062. doi: 10.1016/S0021-9258(18)45665-7
- Zhang, Y., Chen, Y., Wang, C., Lao, C. -C., Liu, X., Wu, W., et al. (2020). ProDCoNN: Protein design using a convolutional neural network. *Proteins: Struct. Funct. Bioinform.* 7, 819–829. doi: 10.1002/prot.25868
- Zhang, Y., Li, P., Pan, F., Liu, H., Hong, P., Liu, X. and Zhang, J. (2021). Applications of AlphaFold beyond protein structure prediction. *bioRxiv [Preprint]*. doi: 10.1101/2021.11.03.467194v1
- Zhang, Y., Zhang, Q., Zhou, J., and Zou, Q. (2022). A survey on the algorithm and development of multiple sequence alignment. *Brief. Bioinform.* 23, bbac069. doi: 10.1093/bib/bbac069