

GLOBAL-CONTEXT AWARE GENERATIVE PROTEIN DESIGN

Cheng Tan^{1,2,3†} Zhangyang Gao^{1,2,3†} Jun Xia^{1,2,3} Bozhen Hu^{1,2,3} Stan Z. Li^{1,2*}

¹ Zhejiang University. ² AI Lab, School of Engineering, Westlake University.

³ Institute of Advanced Technology, Westlake Institute for Advanced Study

ABSTRACT

The linear sequence of amino acids determines protein structure and function. Protein design, known as the inverse of protein structure prediction, aims to obtain a novel protein sequence that will fold into the defined structure. Recent works on computational protein design have studied designing sequences for the desired backbone structure with local positional information and achieved competitive performance. However, similar local environments in different backbone structures may result in different amino acids, which indicates the global context of protein structure matters. Thus, we propose the **Global-Context Aware** generative de novo protein design method (GCA), consisting of *local* modules and *global* modules. While *local* modules focus on relationships between neighbor amino acids, *global* modules explicitly capture non-local contexts. Experimental results demonstrate that the proposed GCA method achieves state-of-the-art performance on structure-based protein design. Our code and pretrained model have been released on Github¹.

Index Terms— Bio signal processing, computational biology, structural biology, protein design, deep learning

1. INTRODUCTION

Computational protein design, which aims to invent protein molecules with desired structures and functions automatically, has a wide range of applications in therapeutics and pharmacology [1, 2, 3]. Recent years have witnessed remarkable advancements in this field with increased computation power, in which many of them are led by deep learning techniques [4, 5, 6]. While classical protein design approaches depend on composite energy functions of protein physics and sampling algorithms for exploring both sequence and structure spaces, data-driven approaches take advantage of deep neural networks to generate protein sequences with less complex prior knowledge.

Designing a protein sequence for a given structure remains challenging, as the difficulty in mapping the 3D space

of structures to the vast-size sequence space. Current data-driven protein design methods [7, 8, 9, 10] agree on the assumption based on biology and physics prior knowledge that, for each amino acid, its neighborhoods have the most immediate and vital effects on itself. The majority of such methods represent protein structures as graphs with hand-crafted features and aggregate local messages in hidden layers. The computational protein design process is formulated to learn valuable features from 3D structures with the local message passing mechanism. However, *the similar local environment in different proteins may correspond to different amino acids*. Local neighbors do matter [11], but it is not enough to obtain high-quality protein sequences.

To fully explore the non-local information, we propose the **Global-Context Aware** generative de novo protein design method (GCA) with both local and global modules. While local modules are built upon graph attention networks that aggregate local messages gained from neighbors with different weights, global modules extend local graph attention to global self-attention neural networks in the form of Transformer [12] architectures. As shown in Fig. 1, the local module focuses on adjacent structure information though distant nodes can deliver information implicitly; the global module explicitly gathers information from distant nodes in a self-attention mechanism. By composing multiple blocks of local modules and global modules, our proposed GCA can capture high-order dependencies between protein sequences and protein structures in both neighbor-level and overall-level.

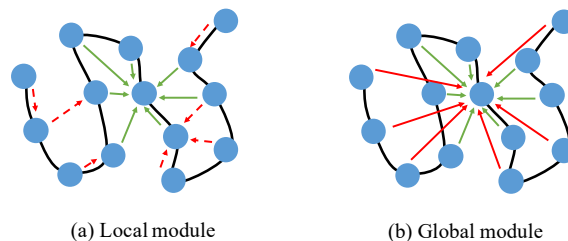


Fig. 1. The comparison between the local module and global module. Information flows from adjacent nodes, and distant nodes are denoted as green arrows and red arrows, respectively. The red dashed arrows indicate the implicit information flow from distant nodes.

*Corresponding author

†Equal contribution

¹github.com/chengtang9907/gca-generative-protein-design

This paper is organized as follows: we introduce the details of our proposed GCA generative de novo protein design method in section 2. Experimental results are reported in section 3, and we conclude in section 4.

2. PROPOSED METHOD

2.1. Preliminaries

Protein primary structure is the linear sequence of amino acids, typically notated as a string of letters. A protein sequence $\mathcal{S}^N = \{(a)^N | a \in \{A, R, N, ..V\}\}$ has N amino acids while each of them is represented by a letter of twenty possible letters such as A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, and V. A protein sequence will fold into a protein tertiary structure $\mathcal{X}^N = \{\mathbf{x}_i^\omega \in \mathbb{R}^3 : 1 \leq i \leq N, \omega \in \{C\alpha, C, N, O\}\}$, where N is the number of amino acids and ω indicates the chain in protein. As shown in Fig. 2, protein design task predicts the protein sequence of a given protein structure, while structure prediction task is the opposite.

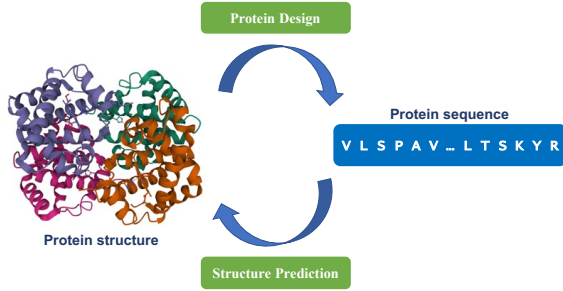


Fig. 2. The comparison of two important tasks in protein modeling: structure prediction and protein design. The 3D visualization is created by Mol* Viewer [13].

2.2. Represent protein as a graph

The structure of a protein is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where node feature $\mathbf{v}_i \in \mathcal{V}$ corresponds to an amino acid while edge feature $\mathcal{E} = \{\mathbf{e}_{ij}\}_{j \in \mathcal{N}_i}$ suggests the rotation-invariant and translation-invariant relationships between each pair of nodes \mathbf{v}_i and \mathbf{v}_j . In particular, \mathcal{N}_i denotes the K -nearest neighbors of node i calculated by Euclidean distances of the backbone.

For node features, we construct three dihedral angles $\{\phi_i, \psi_i, \omega_i\}$ of the protein backbone from $C_{i-1}, N_i, C\alpha_i, C_i$, and N_{i+1} . Then these dihedral angles are embedded on the 3-torus as $\mathbf{v}_i = \{\sin, \cos\} \times \{\phi_i, \psi_i, \omega_i\}$.

For edge features, we focus on describing relative spatial relationships between amino acids that satisfy rotation-invariant and translation-invariant properties. To simplify the computation, we only consider the position $\mathbf{x}_i^{C\alpha}$ of the alpha carbon $C\alpha$ as it's the central carbon atom in each amino acid. The distance $\|\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}\|_2, \forall i \neq j$ is encoded

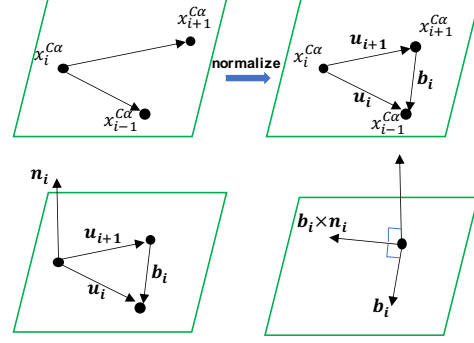


Fig. 3. A general view of how the local coordinate system is built. The final coordinate satisfies $\mathbf{b}_i \perp \mathbf{n}_i, \mathbf{b}_i \times \mathbf{n}_i \perp \mathbf{n}_i, \mathbf{b}_i \times \mathbf{n}_i \perp \mathbf{b}_i$.

by Gaussian radial basis functions $\mathbf{r}(\cdot)$. Then, as shown in Fig. 3, the direction is encoded by $\mathbf{O}_i^T \frac{\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}}{\|\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}\|}$ while $\mathbf{O}_i = [\mathbf{b}_i \quad \mathbf{n}_i \quad \mathbf{b}_i \times \mathbf{n}_i]$ defines a local coordinate system for each amino acid by:

$$\mathbf{u}_i = \frac{\mathbf{x}_i^{C\alpha} - \mathbf{x}_{i-1}^{C\alpha}}{\|\mathbf{x}_i^{C\alpha} - \mathbf{x}_{i-1}^{C\alpha}\|}, \mathbf{b}_i = \frac{\mathbf{u}_i - \mathbf{u}_{i+1}}{\|\mathbf{u}_i - \mathbf{u}_{i+1}\|}, \mathbf{n}_i = \frac{\mathbf{u}_i \times \mathbf{u}_{i+1}}{\|\mathbf{u}_i \times \mathbf{u}_{i+1}\|}. \quad (1)$$

The orientation is encoded by the common-used quaternion representation of rotation matrix $\mathbf{q}(\mathbf{O}_i^T \mathbf{O}_j)$. Thus, the edge feature \mathbf{e}_{ij} is the concatenation of the distance, direction and orientation encodings as:

$$\mathbf{e}_{ij} = \left(\mathbf{r}(\|\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}\|_2), \mathbf{O}_i^T \frac{\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}}{\|\mathbf{x}_j^{C\alpha} - \mathbf{x}_i^{C\alpha}\|}, \mathbf{q}(\mathbf{O}_i^T \mathbf{O}_j) \right). \quad (2)$$

2.3. Network architecture

2.3.1. Local module

The local module is a graph neural network (GNN) that aggregates both node embeddings and local edge embeddings and updates the node embedding for further sequence generations. Considering a L -layer GNN, the key operations *aggregating* and *updating* can be formulated as follows:

$$\mathbf{h}_{\mathcal{N}_i}^{(l)} = \text{aggregating}^{(l)}(\{\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, \mathbf{h}_{\mathbf{e}_{ij}}\} : j \in \mathcal{N}_i), \quad (3)$$

$$\mathbf{h}_i^{(l)} = \text{updating}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_{\mathcal{N}_i}^{(l)}), \quad (4)$$

where $\mathbf{h}_i^{(l)} \in \mathbb{R}^D$ denotes the embedding of node i on the l -th layer, $\mathbf{h}_{\mathcal{N}_i}^{(l)} \in \mathbb{R}^{K \times D}$ denotes the local edge embedding of node i 's neighbors on the l -th layer, K is the number of local neighbors, and D is the dimensions of the embedding.

3. EXPERIMENTS

3.1. Experimental settings

3.1.1. Dataset

We use the CATH 4.2 dataset collected by [7] to evaluate the ability of our method to generalize across different protein folds. This dataset obtains full chains up to length 500, and structures have been partitioned with 40% non-redundancy by their CATH (Class, Architecture, Topology, Homologous) for all domains. As the evaluation set and the test set have minor similarities to the training set, we consider this dataset is approximate to the real-world scenarios that require the design of novel structures. With no CAT overlap between sets, there are 18024 chains in the training set, 608 chains in the validation set, and 1120 chains in the test set, respectively. Two subsets of the entire test set are evaluated simultaneously: a 'Short' subset containing chains up to length 100 and a 'Single chain' subset for comparing with baselines that only use the single chain. We also consider a smaller dataset TS50, which is the standard benchmark introduced by [14]. Though the model is still trained on the CATH 4.2 dataset, we filter the training and validation sets to ensure there is no overlap with TS50.

3.1.2. Measurement

Perplexity Following [7, 15], we define the perplexity that evaluates the predicted protein sequences from natural language perspective:

$$\text{PERP}(\mathcal{S}^N, \mathcal{X}^N) = \exp\left(-\frac{1}{N} \sum_{i=1}^N \mathcal{S}_i^N \log p(\mathcal{S}_i^N | \mathcal{X}_i^N)\right), \quad (12)$$

where $(\mathcal{S}^N, \mathcal{X}^N)$ is the sequence-structure pair of a protein with N amino acids. $\mathcal{S}_i^N, \mathcal{X}_i^N$ denote the i -th amino acid in sequence and structure respectively. $p(\mathcal{S}_i^N | \mathcal{X}_i^N)$ is the output probability from the model.

Recovery To evaluate the predicting accuracy of the protein sequence at per-residue level, we consider the recovery:

$$\text{REC}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathcal{X}^N, \mathcal{S}^N) \in \mathcal{D}} \frac{1}{N} \sum_{i=1}^N \mathbb{1}[S_i^N = \arg \max p(S_i^N | \mathcal{X}_i^N)], \quad (13)$$

where \mathcal{D} denotes the whole dataset.

3.1.3. Model architecture and optimization

In all experiments, GCA model is built by three blocks of both local modules and global modules for the encoder and decoder with the hidden dimension of 128. The Adam optimizer with learning rate of $1e^{-3}$ is employed. Models are

In particular, $\mathbf{h}_i^{(0)} \in \mathbb{R}^D$ is the embedding of \mathbf{v}_i , and $\mathbf{h}_{e_{ij}} \in \mathbb{R}^D$ is the embedding of the edge feature e_{ij} . The local edge information flows into node embeddings at each layer, while distant edge information flows through high-level layers.

In order to capture the relationships in local neighborhoods, we generalize graph attention scheme that take advantage of attention coefficients $\alpha \in \mathbb{R}^{N \times K}$ as strong relational inductive bias. Specifically, the attention coefficients are calculated as follows:

$$\alpha_{ij} = \frac{\exp(c_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(c_{ik})}, \forall j \in \mathcal{N}_i, \quad (5)$$

where c_{ij} is expressed as:

$$c_{ij} = \sigma\left(a^T [\mathbf{W}\mathbf{h}_i^{(l-1)} \parallel \mathbf{W}\mathbf{h}_j^{(l-1)} \parallel \mathbf{W}\mathbf{h}_{e_{ij}}]\right), \forall j \in \mathcal{N}_i, \quad (6)$$

and $\mathbf{W} \in \mathbb{R}^{D \times D}$, $a \in \mathbb{R}^{3D}$ are learnable parameters, σ is the activation function, \parallel is the concatenation operation.

Thus, the *aggregating* operation is adopted as:

$$\mathbf{h}_{\mathcal{N}_i}^{(l)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_r [\mathbf{h}_i^{(l-1)} \parallel \mathbf{h}_j^{(l-1)} \parallel \mathbf{h}_{e_{ij}}], \quad (7)$$

where $\mathbf{W}_r \in \mathbb{R}^{D \times 3D}$ encodes the relation between i and j . The *updating* operation is simply renovating hidden layers by their local neighbors: $\mathbf{h}_i^{(l)} = \mathbf{h}_{\mathcal{N}_i}^{(l)}$.

2.3.2. Global module

The global module is the fully self-attention network that generalizes Transformer [12] to protein graph. Specifically, the attention coefficients are calculated as follows:

$$\alpha_{ij} = \frac{\exp(c_{ij})}{\sum_{k \in \mathcal{V}} \exp(c_{ik})}, \quad (8)$$

where c_{ij} is expressed as:

$$c_{ij} = \frac{1}{\sqrt{d}} \left(\mathbf{W}_q \mathbf{h}_i^{(l-1)} \right)^T \left(\mathbf{W}_k [\mathbf{h}_i^{(l-1)} \parallel \mathbf{h}_j^{(l-1)} \parallel \mathbf{h}_{e_{ij}}] \right), \quad (9)$$

where $\mathbf{W}_q \in \mathbb{R}^{D \times D}$, $\mathbf{W}_k \in \mathbb{R}^{D \times 3D}$ are parameter matrices for the query and key, and d is a scale factor.

Then, the *aggregating* operation is formulated as:

$$\mathbf{h}_{\mathcal{N}_i}^{(l)} = \sum_{j \in \mathcal{V}} \alpha_{ij} \mathbf{W}_r \mathbf{h}_j^{(l-1)}, \quad (10)$$

the *updating* operation is defined by employing layer normalization (LayerNorm), dropout (DropOut) and fully connected networks (FFN):

$$\mathbf{h}_i^{(l)} = \text{LayerNorm}(\mathbf{h}_{\mathcal{N}_i}^{(l)} + \text{DropOut}(\text{FFN}(\mathbf{h}_{\mathcal{N}_i}^{(l)}))). \quad (11)$$

The overall architecture with stacked local modules and global modules is shown in Fig. 4.

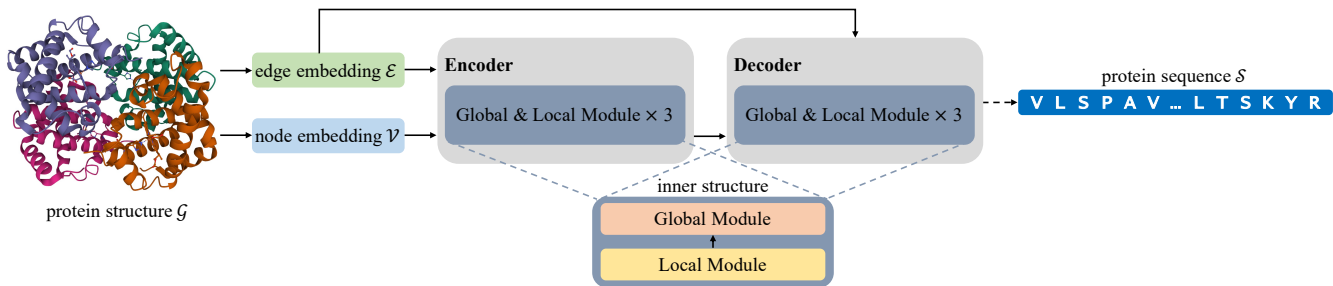


Fig. 4. The architecture of our proposed method.

trained for 100 epochs while the sequence of each batch contains up to 2,500 characters.

3.2. Experimental results

We first present the median of PERP in Table 1. While the structure-free language model LSTMs produce confusing protein sequences, structure-based models obtain less-perplex protein sequences, indicating the importance of structural features. GCA outperforms other structure-based models as global contexts of protein structures are taken into account.

Methods	Short	Single chain	All
Language models			
LSTM ($h = 128$)	16.06	16.38	17.13
LSTM ($h = 256$)	16.08	16.37	17.12
LSTM ($h = 512$)	15.98	16.38	17.13
SPIN2	12.11	12.61	-
Structure-based models			
StructTrans	8.56	8.97	7.14
StructGNN	8.40	8.84	6.69
GCA	7.68	8.09	6.44

Table 1. Performance of different methods on CATH 4.2 dataset assessed by PERP (lower is better).

Though PERP matters from the perspective of natural language, REC that evaluates the ability of models in inferring sequences given determined structures is also crucial. We compare GCA with other structure-based models in Table 2.

Methods	Short	Single chain	All
StructTrans	31.59	30.35	33.90
StructGNN	30.90	30.85	35.25
GCA	33.25	33.04	36.11

Table 2. Performance of different methods on CATH 4.2 dataset assessed by REC (higher is better).

GCA obtains the highest REC on all three sets among these structure-based methods. Moreover, the recovery of StructGNN and StructTrans drops significantly in 'Short' and 'Single chain' sets, which suggests they are overfitting on

long sequences and multiple chains, while GCA performs consistently well on them. As few structural features can be explored in short sequence and single chain, the prediction is relatively difficult. However, the global information in GCA makes up for the deficiency of structural features of short chains, making performance significantly improved.

To compare with other methods, we conduct experiments on the standard TS50 dataset and show the results in Table 3. The methods for comparison include the CNN-based ProDCoNN [16], the distance-map-based SPROF [17], the graph-based GVP [8] the sequential representation method SPIN [14] and SPIN2 [18], the constraint satisfaction method ProteinSolver [9], and the popular method Rosetta. GCA achieves remarkable performance and outperforms other methods by a large margin.

Methods	REC
Rosetta	30.0
SPIN	30.3
ProteinSolver	30.8
SPIN2	33.6
StructTrans	36.1
StructGNN	38.0
SPROF	39.2
ProDCoNN	40.7
GVP	44.1
GCA	47.0

Table 3. Performance of different methods on TS50 dataset assessed by REC (higher is better).

4. CONCLUSION

We introduce the consideration of global information and propose the global-context aware generative de novo protein design method, consisting of local modules and global modules. The local module propagates neighborhood messages across layers, and the global module emphasizes long-term dependencies. Experimental results show that GCA outperforms state-of-the-art methods on benchmark datasets. In 'Short' and 'Single chain' sets, the global-context aware mechanism significantly improves the performance, indicating the potentials to promote structure-based protein design.

5. REFERENCES

- [1] TJ Brunette, Fabio Parmeggiani, Po-Ssu Huang, Gira Bhabha, Damian C Ekiert, Susan E Tsutakawa, Greg L Hura, John A Tainer, and David Baker, “Exploring the repeat protein universe through computational protein design,” *Nature*, vol. 528, no. 7583, pp. 580–584, 2015.
- [2] Po-Ssu Huang, Scott E Boyken, and David Baker, “The coming of age of de novo protein design,” *Nature*, vol. 537, no. 7620, pp. 320–327, 2016.
- [3] Robert A Langan, Scott E Boyken, Andrew H Ng, Jennifer A Samson, Galen Dods, Alexandra M Westbrook, Taylor H Nguyen, Marc J Lajoie, Zibo Chen, Stephanie Berger, et al., “De novo design of bioactive protein switches,” *Nature*, vol. 572, no. 7768, pp. 205–210, 2019.
- [4] Wenhao Gao, Sai Pooja Mahajan, Jeremias Sulam, and Jeffrey J. Gray, “Deep learning in protein structural modeling and design,” *Patterns*, vol. 1, no. 9, pp. 100142, 2020.
- [5] Andriy Kryshchak, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moulton, “Critical assessment of methods of protein structure prediction (casp)—round xiii,” *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1011–1020, 2019.
- [6] Kevin K Yang, Zachary Wu, and Frances H Arnold, “Machine-learning-guided directed evolution for protein engineering,” *Nature methods*, vol. 16, no. 8, pp. 687–694, 2019.
- [7] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola, “Generative models for graph-based protein design,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, pp. 15820–15831, Curran Associates, Inc.
- [8] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror, “Learning from protein structure with geometric vector perceptrons,” in *International Conference on Learning Representations*, 2021.
- [9] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M. Kim, “Fast and flexible protein design using deep graph neural networks,” *Cell Systems*, vol. 11, no. 4, pp. 402–411.e4, 2020.
- [10] Yue Cao, Payel Das, Vijil Chenthamarakshan, Pin-Yu Chen, Igor Melnyk, and Yang Shen, “Fold2seq: A joint sequence(1d)-fold(3d) embedding-based generative model for protein design,” in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 1261–1271, PMLR.
- [11] Leonardo FR Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych, “Modeling global and local node contexts for text generation from knowledge graphs,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 589–604, 2020.
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [13] David Sehnal, Sebastian Bittrich, Mandar Deshpande, Radka Svobodová, Karel Berka, Václav Bazgier, Sameer Velankar, Stephen K Burley, Jaroslav Koča, and Alexander S Rose, “Mol* viewer: modern web app for 3d visualization and analysis of large biomolecular structures,” *Nucleic Acids Research*, 2021.
- [14] Zhixiu Li, Yuedong Yang, Eshel Faraggi, Jian Zhan, and Yaoqi Zhou, “Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles,” *Proteins: Structure, Function, and Bioinformatics*, vol. 82, no. 10, pp. 2565–2573, 2014.
- [15] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher, “Progen: Language modeling for protein generation,” *arXiv preprint arXiv:2004.03497*, 2020.
- [16] Yuan Zhang, Yang Chen, Chenran Wang, Chun-Chao Lo, Xiuwen Liu, Wei Wu, and Jinfeng Zhang, “Prodcnn: Protein design using a convolutional neural network,” *Proteins: Structure, Function, and Bioinformatics*, vol. 88, no. 7, pp. 819–829, 2020.
- [17] Sheng Chen, Zhe Sun, Lihua Lin, Zifeng Liu, Xun Liu, Yutian Chong, Yutong Lu, Huiying Zhao, and Yuedong Yang, “To improve protein sequence profile prediction through image captioning on pairwise residue distance map,” *Journal of chemical information and modeling*, vol. 60, no. 1, pp. 391–399, 2019.
- [18] James O’Connell, Zhixiu Li, Jack Hanson, Rhys Hefferman, James Lyons, Kuldip Paliwal, Abdollah Dehzangi, Yuedong Yang, and Yaoqi Zhou, “Spin2: Predicting sequence profiles from protein structures using deep neural networks,” *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 6, pp. 629–633, 2018.