HelixGAN: A bidirectional Generative Adversarial Network with search in latent space for generation under constraints

Xuezhi Xie
Computer Science
University of Toronto
xuezhi.xie@mail.utoronto.ca

Philip M. Kim, Ph.D.
Computer Science & Molecular Genetics
University of Toronto
pm.kim@mail.utoronto.ca

Abstract

Protein engineering has become an important field in biomedicine with application in therapeutics, diagnostics and synthetic biology. Due to the complexity of protein structure *de novo* computational design remains a difficult problem. As helices are an abundant structural feature and play a vital role in determination of the protein structure, full atom *de novo* computational design for helices would be an important step. Here, we apply Wasserstein bi-directional Generative Adversarial Networks to generate full atom helical structures. To design for structure or function, we allow the design according to structural constraints and introduce a novel Markov Chain Monte Carlo search mechanism with the encoder such that the generated helices match target "hotspot" residues structures. Our model generates helices matching well to the target hotpots (within 3 Å RMSD) and with near-native geometries for a large fraction of the test cases. We demonstrate that our approach is able to quickly generate structurally plausible solutions, bringing us closer to the final goal of full atom computational protein design.

1 Introduction

Computational protein design holds enormous promise for protein engineering and biomedicine. While deep learning has been successesful at protein structure prediction [6], computational design of novel proteins remains difficult due to the complexity and the intricate functionality of protein structures [8, 12]. Helices are the most commonly occurring secondary structure of proteins, imparting stability, and representing 30% of the structure of the average globular protein [9, 10]. Full atom *de novo* computational design for helices would be an important first step towards computational protein design; with current forcefield based methods, exact design of helices is still lacking. In other words, discovering novel, non-native helix structures could directly facilitate the design of new proteins. While the Protein Data Bank (PDB) offers a relatively large repertoire of native helix structures (roughly 3 million helices), it is still a small fraction of possible stable helices. One potential solution is to train deep generative models on the PDB to create novel helices. In order to be useful designs, these helices need to fit certain structural design parameters such as fitting into certain binding pockets or novel structural elements. We express these parameters as "hotspot" residues which the designs will aim to match structurally as closely as possible; it is known that in real proteins, such hotspot residues mediate target binding, recognition, and receptor activation [3]. The target hotspot residues are mostly identified by experiments [3]. Given identified target hotspots, our research focuses on solving a generative design problem with constraints where the generated helices are as structurally close to the provided hotspot residues as possible. We could apply such hotspot based helix generation to generate novel helices to bind certain pockets as receptor agonists or inhibitors, or to generate "mirror image" peptides with many improved properties as therapeutics.

Machine Learning for Structural Biology Workshop, NeurIPS 2021.

In computational biology, generative methods have been increasingly used to generate realistic data due to their ability to discern patterns from massive complex datasets and create synthetic data with desired properties[5, 7]. Generative Adversarial Networks (GANs) are a recently developed class of generative models that have found wide adoption in biology. In 2017, Killoran et al. [7] applied Wasserstein GANs to produce generic DNA sequences. The Wasserstein GAN is a widely-used variant of the GAN, and it reduces the Earth Mover (Wasserstein) distance between real data and generated data[1]. In 2019, Gupta and Zou [5] proposed a feedback GAN using external function analysers to gradually update the training data to produce DNA with desired properties. In 2021, Repecka et al. [11] developed a GAN model called ProteinGan utilizing convolution and attention layers to produce novel protein sequences with the favored physical property like enzymatic activity. Here, instead of working strictly on sequences, we focus on full atom structures and present a Wasserstein bi-directional Generative Adversarial Network (WBIGAN) full-atom model with a Markov chain Monte Carlo (MCMC) search mechanism to generate our desired helical structures that fit hotspot constraints.

Our work is the first step towards full atom protein design using generative methods. The main contributions of this paper are that (i) we developed a generative full-atom model using WBIGAN to estimate the distribution of helical structures in a way that is invariant to translation and rotation, and (ii) a MCMC search mechanism to optimize the generated data for desired structural properties as way to perform generation under constraint.

2 Methods

2.1 Database Construction and Encoding

The PDB stores over 100,000 protein structures and is a key resource for structural biologists. Our method captures all helices from the PDB to build our helical database. There are around 3 million helical structures in our database. The database is further divided into training and test datasets. We filtered the test dataset to ensure data in the test set is no more than 42.8% sequence identity compared with any data in the training set. The hotspot residues are problem specific and are usually no more than three or four residues[3]. To test our model performance with low cost, the hotspot residues are randomly selected on the test dataset to mimic experiments. We randomly selected three hotspots on each test case and saved their structures as the target hotspot residues. In total we prepared 3118 test samples as the test set.

Regarding encoding, the principle is to make the structure invariant to translation and rotation. Therefore, 3D Cartesian coordinates can not be directly encoded, and we utilize internal coordinates like angles to represent the structure. We developed a novel encoding method where the helices are encoded as the combined vectors of the primary sequence and structural information. The one-hot encoding method is utilized for the amino acid sequences. With respect to geometry, the structure is divided into mainchain and sidechains. As demonstrated in the Figure 1, the mainchain is represented as phi, psi, omega, and bond angles while the sidechain is represented using 5 chi angles. In the end, all information is concatenated together as the encoded vectors.

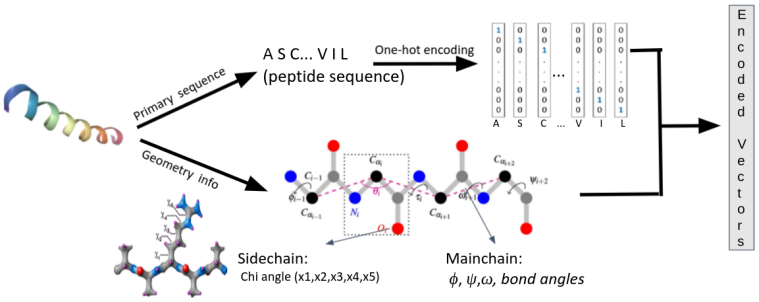


Figure 1: **Encoding**: the helix is encoded as the combined vectors of the primary sequence information and structural information. The structural information is further divided into sidechain and mainchain. The mainchain is represented as phi (ϕ), psi (ψ), omega (ω) and bond angles while the sidechain is represented using 5 chi (χ) angles.

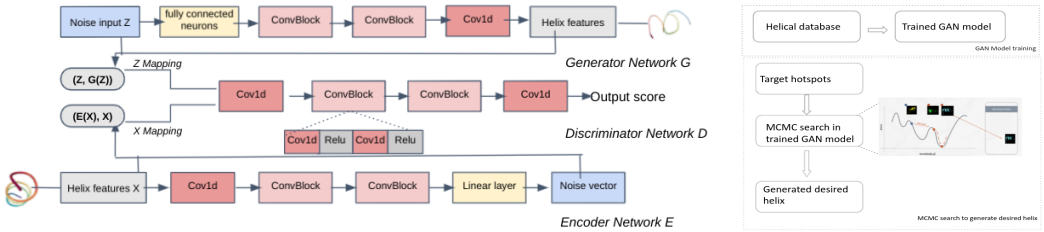
2.2 GAN Model Architecture

GAN is a powerful framework for learning arbitrarily complex data distributions. A GAN is composed of two modules: a generator, to generate data from latent space, and a discriminator, to distinguish whether the input data is real or not. Compared with vanilla GAN, bidirectional GAN (BIGAN) [2] has a third module called encoder which maps data x to latent representations z. We utilize this encoder module to have a starting point in latent space for our MCMC search mechanism. The Wasserstein BIGAN (WBIGAN) is a variant of the BIGAN which minimizes the Earth Mover (Wasserstein) distance between the distribution of real data and the distribution of generated data [1]. A gradient penalty is imposed for gradients above one in order to maintain a Lipshitz constraint [4]. WGANs have been demonstrated empirically helpful to stabilize the GANs' training process. The loss of our model is as shown in Equation 1. Our developed GAN model utilized the WBIGAN architecture with gradient penalty. The architecture of the model is as shown in Figure 2(a). When sampling from the generator, the argmax of the probability distribution is taken to output a single amino acid at each position with other internal coordinate features to construct the structure.

$$\begin{cases} \text{D loss} = & \sum_{z\in p(z)} D(G(z),z) - \sum_x D(x,E(x)) + \lambda * \textit{Gradient Penalty} \\ & \text{where } \textit{Gradient Penalty} = \sum_{z\in p(z)} (\|\nabla_{G(z)} D(G(z),z)\|_2 - 1)^2 \\ \text{EG loss} = & \sum_x D(x,E(x)) - \sum_{z\in p(z)} D(G(z),z) \end{cases} \tag{1}$$

2.3 MCMC Search Mechanism with Encoder

After model training, we developed a MCMC search mechanism to generate the data matching our desired target hotspots. The MCMC search method we used is called simulated annealing and it could obtain approximate global optimization in a large search space for an optimization problem[13]. Our optimization problem is to minimize the RMSD between the target hotpots and matching residues in valid generated data. Regarding MCMC search, a well-chosen starting point is critical since it can avoid getting stuck in some local minima and efficiently save run time. We utilized the encoder module, mapping data into latent space, from trained WBIGAN to produce a starting point. Our trained encoder would take the target hotspot residues and convert it into a starting point in latent space. Regarding our approach, the starting point is firstly converted to the helical structure by our model to calculate the RMSD. After that, a random neighbor move from that point would be given to the model to update the RMSD. If the RMSD is smaller, the move will be accepted. Otherwise, a probability function P = $e^{(RMSD-RMSD')/T}$ is utilized to decide whether to take the move. The process would be repeated until meeting the termination condition. Figure 2(b) is the flowchart to summary the whole process. Overall, the model is firstly trained on helical database and then the MCMC search is performed to search in latent space of our model to generate the desired helices.



(a) Model Architecture (b) Overall Flowchart

Figure 2: **Model Architecture, and overall flowchart** (a): The model contains three modules: generator, discriminator (crtics) and encoder. The modules are largely composed of ConvBlocks which include Relu activation and Conv1d layers. (b):The flowchart for our approach; training the WBIGAN model and then implementing the MCMC search on the latent space of trained WBIGAN

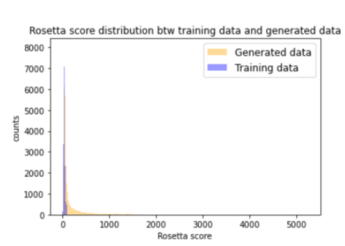
3 Results and discussion

We evaluated the quality of our generated helices according to an independent scoring metric and how well they matched the target hotspot residues. The Rosetta score, a widely used physics-based

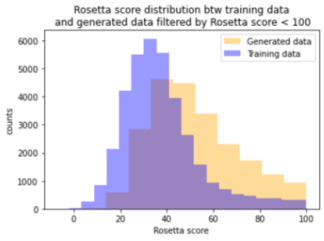
energy function, was utilized as an indication of the data quality. Figure 3 (a) shows the Rosetta score distribution of the training data and the generated data; while generated helices clearly contain a large fraction of physically reasonable helices (with low Rosetta score), the generated data contains a wider range of Rosetta score distribution compared with training data. Figure 3(b) demonstrates the data distribution between generated data and training data where Rosetta score is smaller than 100. Note that additional filtering using the Rosetta score would be possible as it is a fully independent measure. We also examined the PCA of the CA atom coordinates of the backbone, showing that the generated helices are broadly similar, but also filling out a slightly different conformational space in Figure 3 (c). In addition, Figure 3 (d) illustrates the Ramachandran plots (ϕ and ψ angles) for the generated data and the training data. Over all, the model could generate the reasonable structures with the similar distribution as training data.

To test the performance of generating novel helices matching desired target hotspot residues, we evaluated our MCMC Search Mechanism with trained WBIGAN model on the prepared test set. As shown in figure 3(e), the pie chart summarizes the RMSD regarding hotspots between generated data and target hotspots within the limited time. In almost a quarter of the test cases the desired helices were within 2 Å RMSD compared with the target hotspots, so matching very closely, with another 47.6% matching within 3 Å. Figure 3(f) is the Rosetta score distribution regarding the generated helices for test cases. In summary, our method could largely generate high-quality helices with considerably small distance from the given target hotspots.

The performance tests demonstrate that our model generated physically reasonable novel helices and our approach is able to generate helices matching hotspot residues through the MCMC search mechanism. The future work would be to combine our model with reinforcement learning or other folding algorithms to produce the novel protein structure at atomic resolution.

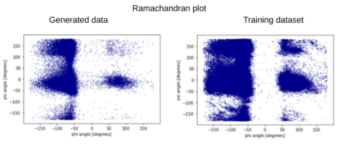


(a) Rosetta score distribution 1

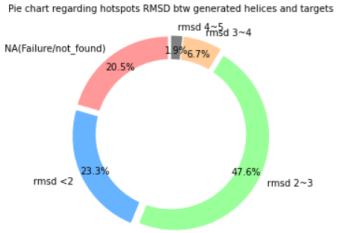


(b) Rosetta score distribution 2

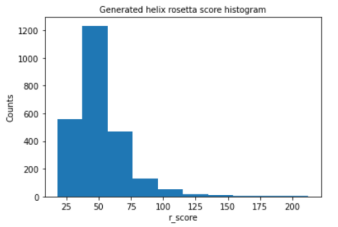
(c) PCA



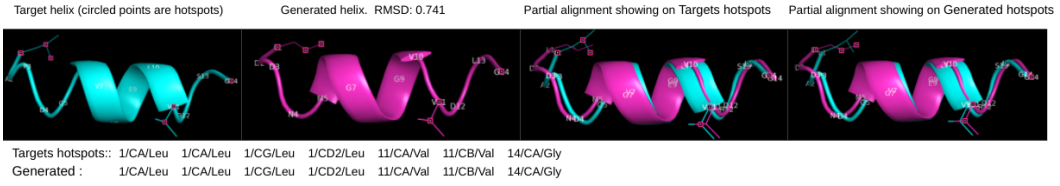
(d) Ramachandran plot



(e) Overall RMSD result



(f) Generated Rosetta distribution



(g) Visualization example

Figure 3: **Results** (a): The Rosetta score distribution between training data and generated data. (b): The Rosetta score distribution where the score is smaller than 100. (c): PCA regarding the CA atom coordinates in backbone. (d): Ramachandran plot on phi, psi angles regarding generated data and training data. (e): RMSD summary for all test cases (f): The generated Rosetta score distribution for all test cases (g): A visualization example about our approach

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[2] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[3] M. Garton, S. Nim, T. A. Stone, K. E. Wang, C. M. Deber, and P. M. Kim. Method to generate highly stable d-amino acid analogs of bioactive helical peptides using a mirror image of the entire pdb. *Proceedings of the National Academy of Sciences*, 115(7):1505–1510, 2018.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[5] A. Gupta and J. Zou. Feedback gan (fbgan) for dna: A novel feedback-loop architecture for optimizing protein functions. *arXiv preprint arXiv:1804.01694*, 2018.

[6] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, pages 1–11, 2021.

[7] N. Killoran, L. J. Lee, A. Delong, D. Duvenaud, and B. J. Frey. Generating and designing dna with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

[8] N. Koga, R. Tatsumi-Koga, G. Liu, R. Xiao, T. B. Acton, G. T. Montelione, and D. Baker. Principles for designing ideal protein structures. *Nature*, 491(7423):222–227, 2012.

[9] C. N. Pace and J. M. Scholtz. A helix propensity scale based on experimental studies of peptides and proteins. *Biophysical journal*, 75(1):422–427, 1998.

[10] L. Pauling, R. B. Corey, and H. R. Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

[11] D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

[12] G. J. Rocklin, T. M. Chidyausiku, I. Goreshnik, A. Ford, S. Houliston, A. Lemak, L. Carter, R. Ravichandran, V. K. Mulligan, A. Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

[13] P. J. Van Laarhoven and E. H. Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.