Deep generative models create new and diverse protein structures

Zeming Lin
NYU & FAIR
zl2799@nyu.edu,zlin@fb.com

Tom Sercu
FAIR
tsercu@fb.com

Yann LeCun
NYU & FAIR
yann@nyu.edu,yann@fb.com

Alexander Rives
FAIR
arives@fb.com

Abstract

We explore the use of modern variational autoencoders for generating protein structures. Models are trained across a diverse set of natural protein domains. Three-dimensional structures are encoded implicitly in the form of an energy function that expresses constraints on pairwise distances and angles. Atomic coordinates are recovered by optimizing the parameters of a rigid body representation of the protein chain to fit the constraints. The model generates diverse structures across a variety of folds, and exhibits local coherence at the level of secondary structure, generating alpha helices and beta sheets, as well as globally coherent tertiary structure. A number of generated protein sequences have high confidence predictions by AlphaFold that agree with their designs. The majority of these have no significant sequence homology to natural proteins.

Most designed proteins are variations on existing proteins. It is of great interest to create *de novo* proteins that go beyond what has been invented by nature. A line of recent work has explored generative models for protein structures [1, 2, 3, 4, 5, 6]. The main challenge for a generative model is to propose stable structures that can be realized as the minimum energy state for a protein sequence, i.e. the endpoint of folding. The space of possible three-dimensional conformations of a protein sequence is exponentially large [7], but out of this set of possible conformations, most do not correspond to stable realizable structures.

In this work we explore the use of modern variational autoencoders (VAEs) as generative models of protein structures. We find that the models can produce coherent local and global structural organization while proposing varied and diverse folds. We use AlphaFold to assess the viability of sampled sequences, finding that many sequences are predicted to fold with high confidence to their designed structures. To assess the novelty of the generated sequences, we search sequence databases including metagenomic information for homologous sequences, finding no significant matches for a large fraction of the generations.

1 Modeling

1.1 Overview

Figure 1 presents an overview of the approach. The structure is implicitly encoded as the minimum of an energy over possible conformations of the protein chain. We write the structure $x^* = \arg\min_x E(x; z) + R(x)$ as the outcome of this minimization. $E(x; z)$ is the output of a decoder. Optionally $R(x)$ subsumes additional energy terms. During training an encoder and

Machine Learning for Structural Biology Workshop, NeurIPS 2021.

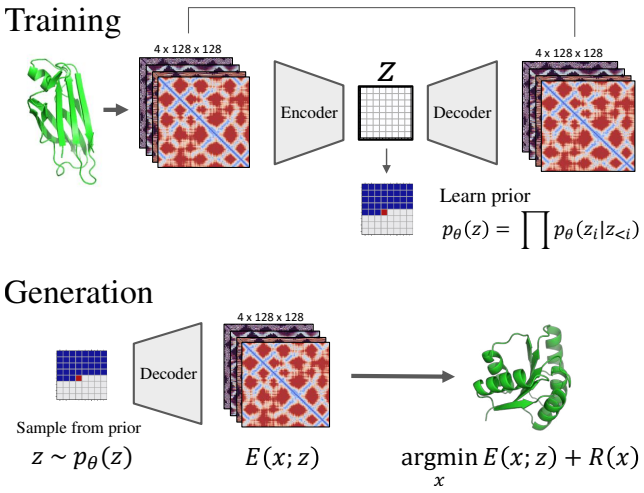

Figure 1: Overview of the method. Each protein structure is represented as a discretized distance map and set of angular coordinates. The model is trained to reconstruct natural protein structures. The decoder output can be interpreted as specifying an energy function over three-dimensional conformations of the protein encoding the structure at its minimum. New structures are generated by passing samples from the prior through the decoder to obtain an energy function specifying the structure. Three-dimensional coordinates are obtained by fitting the angular degrees of freedom of an idealized rigid-body representation of the protein chain to minimize the energy.

decoder are fit to natural structures to learn a discrete latent code z. To produce samples, new codes are drawn from a learned autoregressive prior $p_\theta(z)$, and passed through the decoder to obtain $E(x; z)$.

We compare with several other VAE modeling approaches. Structures are represented through a discretized pairwise distance map and set of angular coordinates. Pairwise representations have been useful in structure prediction [8, 9, 10], and have been recently applied to generative models of protein structures [11, 1]. We use Rosetta to perform the minimization using the all-atom ref2015 score function [12]. Related work is further discussed in Appendix A of the Appendix.

1.2 VAE Models

We consider several VAE model variants to learn the energy $E(x; z)$. All models use a convolutional encoder-decoder architecture unless otherwise specified. We train a downsampling convolutional encoder $q(z|x)$ to compress the structure into a latent variable with a prior $p(z)$; and an upsampling convolutional decoder $E(x; z)$ that maps the latent variable into a structure. We fix models to work on 128 length for simplicity, training on a dataset of full-chain SCOP [13] domains, filtered to the length cutoff.

We present **Conv-VAE** and **MLP-VAE** as classic VAE [14] baselines with a single latent variable, using a convolutional network or MLP for the encoder-decoder. **HVAE** is a hierarchical VAE [15, 16, 17, 18]; **VQ-VAE** is a VQ-VAE architecture [19, 20]; **VQ-VAE-BB** and **HVAE-BB** are versions that learn backbone dihedral angles instead of interresidue angles.

We use Rosetta to solve the minimization problem and obtain a structure. Similar to Yang et al. [21], we input $E(x; z)$ as an additive energy term with `ref2015`, and find a minimum energy poly-alanine chain that fits the energy landscape. Further details on architectural and training choices are described in Appendix B.1.

Across experiments we observe that validation loss does not reflect the quality of generated structures. As a result we use the Rosetta score function for model selection. We evaluate models using a variety of metrics, including the Rosetta score function values, medium and long-range contacts, hydrogen bonds, and secondary structure elements.

Table 1: Comparison of models. Metrics computed across generated structures. $R(x) < 0$ is the percent of all generations with negative Rosetta energy. $R(x)$ is the average absolute value of Rosetta energies. "MR/LR contacts" is the average number of medium or long range contacts, divided by number of residues. "MR/LR polar" is the average number medium or long range hydrogen bonds, divided by number of residues. % helix and % sheet measures the average proportion of α-helices and β-sheets. VQ-VAE generates structures with low Rosetta energies, having long range contacts and hydrogen bonds, and containing β-sheets.

	% $R(x) <$ 0	$R(x)$ Avg	MR/LR contacts	MR/LR polar	% α-helix	% β-sheets
MLP-VAE	0.0043	0.7301	1.5033	0.0329	0.0487	0.0138
Conv-VAE	0.0223	0.3408	1.5426	0.0277	0.1273	0.0154
HVAE	0.0817	0.3526	1.1066	0.0291	0.3229	0.0169
HVAE-BB	**0.3127**	**0.0612**	1.4830	0.0312	**0.4194**	0.0161
VQ-VAE	0.2140	0.2066	1.6849	0.0602	0.3387	0.0720
VQ-VAE-BB	0.1955	0.1671	**1.9890**	**0.0879**	0.2677	**0.1080**

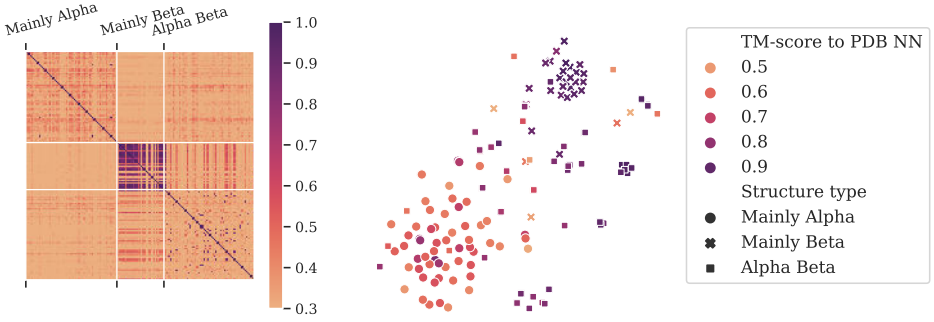

Figure 2: The models generate diverse structures. (Left) Pairwise TM-scores between 180 generated structures. The majority of pairwise TM-scores are lower than the value 0.5 which roughly corresponds to the same fold [22]. (Right) t-SNE [23] plot of the structural space covered by the same proteins. The proteins clearly cluster by category. Each category has structures with higher and lower TM-score to its nearest neighbor in PDB (color-code), indicating the generated structures cover a range of similarity with natural structures.

2 Experiments: Structure Generation

Qualitative inspection As expected, the single latent variable models find it difficult to capture sharpness in the distogram. The associated generations tend to have blurred patches. Visually, HVAE and VQ-VAE both seem to generate sharper and more distinct distograms. To benchmark model fidelity, we generate 3000 samples from $E(x; z)$, with z sampled from the model's respective priors, and find the minimum energy structure using the techniques detailed in Section 1.1. We see that VQ-VAE is the only one that successfully generates high fidelity β-sheets. Examples can be found in Figure S6 and Figure S7. For visualization we select random generations with $R(x) < 0$ and MR/LR contacts > 1.

Quantitative metrics of Rosetta-folded structures Table 1 and Figure S8 compare the models. Large differences in the quality of generations are observed favoring the VQ-VAE models. Although HVAE-BB seem to be best in terms of energy favorable structures, a large proportion of its generated structures are α-helices. For VQ-VAE-BB, we see a large increase in the number of MR/LR contacts and long range hydrogen bonds, Table 1 summarizes data on the quality of generated structures. The specifics of the metrics measured are detailed in Appendix B.2

Novelty and diversity In Figure 2, we use TM-scores between the structures generated with VQ-VAE-BB, and with respect to all of PDB, to evaluate the diversity and novelty of the generated

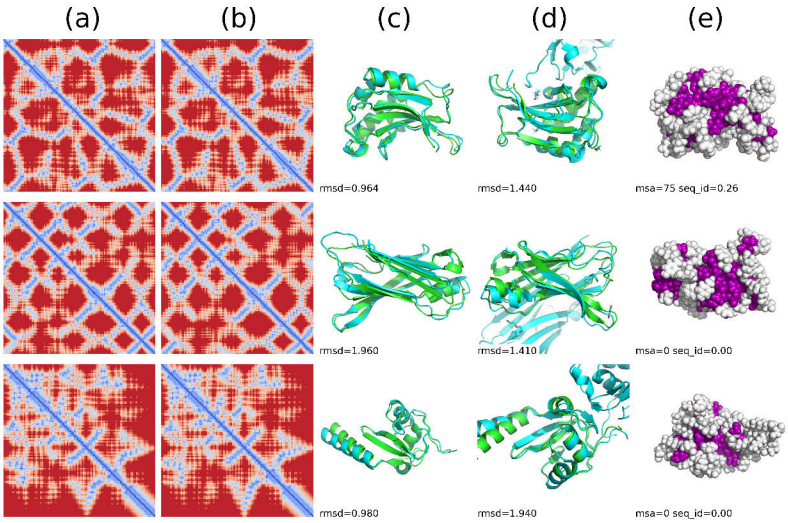

Figure 3: Each row is an individual generation, selected for AlphaFold modeling confidence. (a) $\arg\max$ of the distogram proposed by our model. (b) Distogram of the AlphaFold folded structure. (c) Superposition of designed structure (green) with AlphaFold prediction. (d) Superposition of designed structure (green) with nearest match in PDB by TM-Score. (e) Hydrophobic residues in purple, hydrophilic in white, proteins exhibit hydrophobic core. Most generations have few sequence matches, and low sequence identity with closest match. Full image is found in Figure S10

structures. For generated structures, we apply a filter to discard unlikely proteins, as detailed under "Qualitative inspection". For Figure 2, 60 structures are randomly sampled per structure category. All generated structures are also compared against all known structures in PDB, and the highest TM-score is reported as "TM-score to PDB Nearest Neighbor (NN)". The model generates diverse protein structures with low structural similarity, except in the "Mainly Beta" category.

3 Experiments: Protein Design

We ask whether generated structures can be realized by an amino acid sequence as the endpoint of folding. We design sequences for 40 structures and use AlphaFold to predict their structures. AlphaFold produces high confidence models for 9 of the designed sequences. Of these, 8 proteins had low (< 2.6A) RMSD to AlphaFold predicted structures. We show these proteins in Figure 3. Details on the design methodology can be found in Appendix B.3

Most of the designed sequences do not have homologous sequences in UniRef90 or MGnify [24, 25]. We ran JackHMMER [26] against Uniref90 and MGnify and found no significant sequence matches for 5 designed structures, matches with very low sequence identity (15%) for 2 designed structures, and 75 matches with maximum sequence identity of 26% for the final structure. To confirm this finding, we also ran `FastDesign` against 40 randomly selected natural protein chains between length 64-128. Sequence designs for natural structures tended to result in many more JackHMMER hits against UniRef90, with a higher degree of sequence similarity, shown in Figure S9.

4 Conclusions

We perform a systematic study of deep generative models for designing protein structures. We find that generative models are able to capture local secondary structure as well as globally coherent fold topology, when trained on a diverse set of protein domains. We find that generated structures can be realized by sequence designs that are predicted to correctly fold with high confidence by AlphaFold. Although the generations are structurally similar to existing folds, sequences designed to realize them are often novel. This suggests that neural generative models are capable of generalizing beyond simply recapitulating natural proteins.

References

[1] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in Neural Information Processing Systems*, 31, 2018. URL https://papers.nips.cc/paper/2018/hash/afa299a4d1d8c52e75dd8a24c3ce534f-Abstract.html.

[2] Raphael R Eguchi, Namrata Anand, Christian Andrew Choe, and Po-Ssu Huang. Ig-vae: Generative modeling of immunoglobulin proteins by direct 3d coordinate generation. *bioRxiv*, 2020.

[3] Joe G. Greener, Lewis Moffat, and David T. Jones. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific Reports*, 8(1):16189, November 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-34533-1. URL https://www.nature.com/articles/s41598-018-34533-1.

[4] Ivan Anishchenko, Tamuka M. Chidyausiku, Sergey Ovchinnikov, Samuel J. Pellock, and David Baker. De novo protein design by deep network hallucination. *bioRxiv*, page 2020.07.22.211482, July 2020. doi: 10.1101/2020.07.22.211482. URL https://www.biorxiv.org/content/10.1101/2020.07.22.211482v1.

[5] Namrata Anand, Raphael Eguchi, and Po-Ssu Huang. Fully differentiable full-atom protein backbone generation. March 2019. URL https://openreview.net/forum?id=SJxnVL8YOV.

[6] Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks (gcwgan). *bioRxiv*, page 769919, 2019.

[7] Cyrus Levinthal. How to fold graciously. *Mossbauer spectroscopy in biological systems*, 67: 22–24, 1969.

[8] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted inter-residue orientations. *bioRxiv*, page 846279, 2019.

[9] John Jumper, R Evans, A Pritzel, T Green, M Figurnov, K Tunyasuvunakool, O Ronneberger, R Bates, A Zidek, A Bridgland, et al. High accuracy protein structure prediction using deep learning. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, 22:24, 2020.

[10] Jinbo Xu. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences*, 116(34):16856–16865, August 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1821309116. URL https://www.pnas.org/content/116/34/16856.

[11] Raphael R. Eguchi, Namrata Anand, Christian A. Choe, and Po-Ssu Huang. IG-VAE: Generative Modeling of Immunoglobulin Proteins by Direct 3D Coordinate Generation. *bioRxiv*, page 2020.08.07.242347, August 2020. doi: 10.1101/2020.08.07.242347. URL https://www.biorxiv.org/content/10.1101/2020.08.07.242347v1.

[12] Andrew Leaver-Fay, Michael Tyka, Steven M. Lewis, Oliver F. Lange, James Thompson, Ron Jacak, Kristian Kaufman, P. Douglas Renfrew, Colin A. Smith, Will Sheffler, Ian W. Davis, Seth Cooper, Adrien Treuille, Daniel J. Mandell, Florian Richter, Yih-En Andrew Ban, Sarel J. Fleishman, Jacob E. Corn, David E. Kim, Sergey Lyskov, Monica Berrondo, Stuart Mentzer, Zoran Popović, James J. Havranek, John Karanicolas, Rhiju Das, Jens Meiler, Tanja Kortemme, Jeffrey J. Gray, Brian Kuhlman, David Baker, and Philip Bradley. ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in Enzymology*, 487:545–574, 2011. ISSN 1557-7988. doi: 10.1016/B978-0-12-381270-4.00019-6.

[13] Naomi K. Fox, Steven E. Brenner, and John-Marc Chandonia. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research*, 42(D1):D304–D309, January 2014. ISSN 0305-1048. doi: 10.1093/nar/gkt1240. URL https://doi.org/10.1093/nar/gkt1240.

[14] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. URL http://arxiv.org/abs/1312.6114. arXiv: 1312.6114.

[15] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. *arXiv:1602.02282 [cs, stat]*, May 2016. URL http://arxiv.org/abs/1602.02282. arXiv: 1602.02282.

[16] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. *arXiv:1606.04934 [cs, stat]*, January 2017. URL http://arxiv.org/abs/1606.04934. arXiv: 1606.04934.

[17] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. *arXiv:2007.03898 [cs, stat]*, January 2021. URL http://arxiv.org/abs/2007.03898. arXiv: 2007.03898.

[18] Rewon Child. Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. September 2020. URL https://openreview.net/forum?id=RLRXCV6DbEJ.

[19] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937 [cs]*, May 2018. URL http://arxiv.org/abs/1711.00937. arXiv: 1711.00937.

[20] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. *arXiv:1906.00446 [cs, stat]*, June 2019. URL http://arxiv.org/abs/1906.00446. arXiv: 1906.00446.

[21] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, January 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1914677117. URL https://www.pnas.org/content/117/3/1496.

[22] Yang Zhang and Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, April 2005. ISSN 0305-1048. doi: 10.1093/nar/gki524. URL https://doi.org/10.1093/nar/gki524.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[24] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[25] Alex L Mitchell, Alexandre Almeida, Martin Beracochea, Miguel Boland, Josephine Burgin, Guy Cochrane, Michael R Crusoe, Varsha Kale, Simon C Potter, Lorna J Richardson, Ekaterina Sakharova, Maxim Scheremetjew, Anton Korobeynikov, Alex Shlemov, Olga Kunyavskaya, Alla Lapidus, and Robert D Finn. MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*, 48(D1):D570–D578, January 2020. ISSN 0305-1048. doi: 10.1093/nar/gkz1035. URL https://doi.org/10.1093/nar/gkz1035.

[26] L. Steven Johnson, Sean R. Eddy, and Elon Portugaly. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11(1):431, August 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-431. URL https://doi.org/10.1186/1471-2105-11-431.

[27] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[28] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 2006.

[29] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[31] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1): e1005324, 2017.

[32] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020. ISSN 1476-4687. doi: 10.1038/s41586-019-1923-7. URL https://www.nature.com/articles/s41586-019-1923-7.

[33] Kristian Davidsen, Branden J Olson, William S DeWitt III, Jean Feng, Elias Harkins, Philip Bradley, and Frederick A Matsen IV. Deep generative models for t cell receptor protein sequences. *Elife*, 8:e46935, 2019.

[34] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS computational biology*, 17(2):e1008736, 2021.

[35] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

[36] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, pages 1–8, 2019.

[37] 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:2004.03497*, 2020.

[38] Jingxue Wang, Huali Cao, John Z. H. Zhang, and Yifei Qi. Computational Protein Design with Deep Learning Neural Networks. *Scientific Reports*, 8(1):6349, April 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-24760-x. URL `https://www.nature.com/articles/s41598-018-24760-x`.

[39] John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. 2019.

[40] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible design of novel proteins using graph neural networks. *bioRxiv*, page 868935, 2020.

[41] Namrata Anand, Raphael Ryuichi Eguchi, Alexander Derry, Russ B Altman, and Possu Huang. Protein sequence design with a learned potential. *bioRxiv*, 2020.

[42] Yu-Ru Lin, Nobuyasu Koga, Rie Tatsumi-Koga, Gaohua Liu, Amanda F Clouser, Gaetano T Montelione, and David Baker. Control over overall shape and size in de novo designed proteins. *Proceedings of the National Academy of Sciences*, 112(40):E5478–E5485, 2015.

[43] Jiayi Dou, Anastassia A Vorobieva, William Sheffler, Lindsey A Doyle, Hahnbeom Park, Matthew J Bick, Binchen Mao, Glenna W Foight, Min Yen Lee, Lauren A Gagnon, et al. De novo design of a fluorescence-activating β-barrel. *Nature*, 561(7724):485, 2018.

[44] Po-Ssu Huang, Scott E. Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, September 2016. ISSN 1476-4687. doi: 10.1038/nature19946.

[45] Nobuyasu Koga, Rie Tatsumi-Koga, Gaohua Liu, Rong Xiao, Thomas B Acton, Gaetano T Montelione, and David Baker. Principles for designing ideal protein structures. *Nature*, 491 (7423):222–227, 2012.

[46] Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, 2017.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. *arXiv:1603.05027 [cs]*, July 2016. URL `http://arxiv.org/abs/1603.05027`. arXiv: 1603.05027.

[48] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP Architecture for Vision. *arXiv:2105.01601 [cs]*, May 2021. URL `http://arxiv.org/abs/2105.01601`. arXiv: 2105.01601.

[49] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092 [cs]*, February 2021. URL `http://arxiv.org/abs/2102.12092`. arXiv: 2102.12092.

[50] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv:1701.05517*, 2017.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR arXiv:1512.03385*, 2015.

[52] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. *arXiv:1903.10145 [cs, stat]*, June 2019. URL `http://arxiv.org/abs/1903.10145`. arXiv: 1903.10145.

[53] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[54] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv:1903.12436*, 2019.

[55] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv:2007.03898*, 2020.

[56] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv:2011.10650*, 2020.

[57] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[59] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, page 622803, 2019. URL `https://doi.org/10.1101/622803`.

[60] Patrick Kunzmann and Kay Hamacher. Biotite: a unifying open source computational biology framework in Python. *BMC Bioinformatics*, 19(1):346, October 2018. ISSN 1471-2105. doi: 10.1186/s12859-018-2367-z. URL `https://doi.org/10.1186/s12859-018-2367-z`.

[61] E. N. Baker and R. E. Hubbard. Hydrogen bonding in globular proteins. *Progress in Biophysics and Molecular Biology*, 44(2):97–179, 1984. ISSN 0079-6107. doi: 10.1016/0079-6107(84)90007-5.

[62] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 1983.

[63] Sarel J. Fleishman, Andrew Leaver-Fay, Jacob E. Corn, Eva-Maria Strauch, Sagar D. Khare, Nobuyasu Koga, Justin Ashworth, Paul Murphy, Florian Richter, Gordon Lemmon, Jens Meiler, and David Baker. RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite. *PLOS ONE*, 6(6):e20161, June 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0020161. URL `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0020161`.

A Related Work

Deep generative models A major, long standing goal of deep learning is to represent complex distributions over high dimensional data with a rich hierarchical model [27, 28, 29]. Notable success was achieved with deep generative models with latent variables such as Generative Adversarial Networks (GANs) [30] and Variational Auto-Encoder (VAE) [14]. Vector Quantized VAE (VQ-VAE) [19, 20] finds that discretizing the latent variables to enable higher resolution and fedelity generations. In another notable extension to the original VAE formulation, [17, 15, 18] suggest that hierarchies are the most important part of latent variable models.

Generative models for protein structures While there has been breakthrough progress in deep learning for structure prediction [31, 32, 21, 9], in comparison, little work has been done on generative modeling of protein structures.

There is increasing interest in applying differentiable learning to protein structure generation to create novel proteins. There have been initial successes using GANs and VAEs in generating protein structures [1, 5, 6, 2]. Alternatively, Anishchenko et al. [4] uses a supervised structure prediction model with a "hallucination" loss to jointly design structures and sequences.

Despite the initial success, these methods fall short of the promise of a general-purpose generative model for protein structure. Previous methods would typically focus on generating novel fragments, with not enough global coherence to achieve folding structures [1]. Most methods also consider proteins or fragments of limited length [1, 5], are trained on structures with diversity limited to a few folds like immunoglobulins [2]. The hallucination approach of Anishchenko et al. [4], despite its novelty and promising results, has not shown to produce generations with no known sequence homologs.

Slightly further removed from protein *structure* design (the focus of this paper), are methods to use generative sequence models to generate protein *sequences*. VAEs on amino acid sequences have been applied in several problem settings; VAEs and deep sequence models can capture the sequence diversity of specific protein families without explicitly conditioning on structure information [33, 34, 35, 36, 37, 3]. A more related problem setting is designing protein sequences explicitly conditioned on a backbone structure as in [38, 39, 40, 41]. Finally, conventional approaches to protein design rely on expert design of structures [42, 43], and use the Rosetta toolbox [12] with fragment databases to find sequences that will fold in those structures [44, 45, 46].

B Appendix

B.1 Training and model details

As a baseline, we include a basic encoder-decoder model, where $q(z|x)$ is a ResNet according to He et al. [47]. For **Conv-VAE**, $E(x; z)$ is composed of transposed convolutional blocks. For **MLP-VAE**, $E(x; z)$ is a factorized residual MLP architecture, similar to Tolstikhin et al. [48] - where there are independent MLPs that run across the height, width, and channel dimensions respectively. We use the classical VAE formulation [14] to minimize the Evidence Lower Bound objective.

HVAE is a hierarchical VAE proposed by [15, 16, 17, 18]. Although we tried up to 20 levels of latent variables, we discovered the generations were not as robust and tend to overfit, producing local artifacts without much global coherence, despite being able to have near-perfect reconstruction loss. We settled on a model with just 3 spatial latent variable hierarchies. $q(z|x)$ is a ResNet, and $E(x; z)$ is composed of transposed convolutional blocks.

VQ-VAE is a VQ-VAE architecture, proposed by [19, 20]. One change we introduce is to train a transformer, similar to Ramesh et al. [49], to model the prior of the model. Here, we first train $q(z|x)$ and $E(x; z)$ by setting $z = \text{VQ}(q(z|x))$, with the vector quantizer block introduced in Oord et al. [19]. Then, we train a transformer prior $p_\theta(z) = \Pi_{i=0}^{L} p_\theta(z_i|z_{<i})$, where L is the length of the prior.

VQ-VAE-BB and **HVAE-BB** are versions that learn backbone dihedral angles instead of interresidue angles. We follow the work of Senior et al. [32] in learning a joint phi-psi backbone angle, which then we convert into a von Mises distribution as a part of $E(x; z)$.

In all cases, the encoder resnet is a stack of several bottleneck residual blocks [47] followed by a

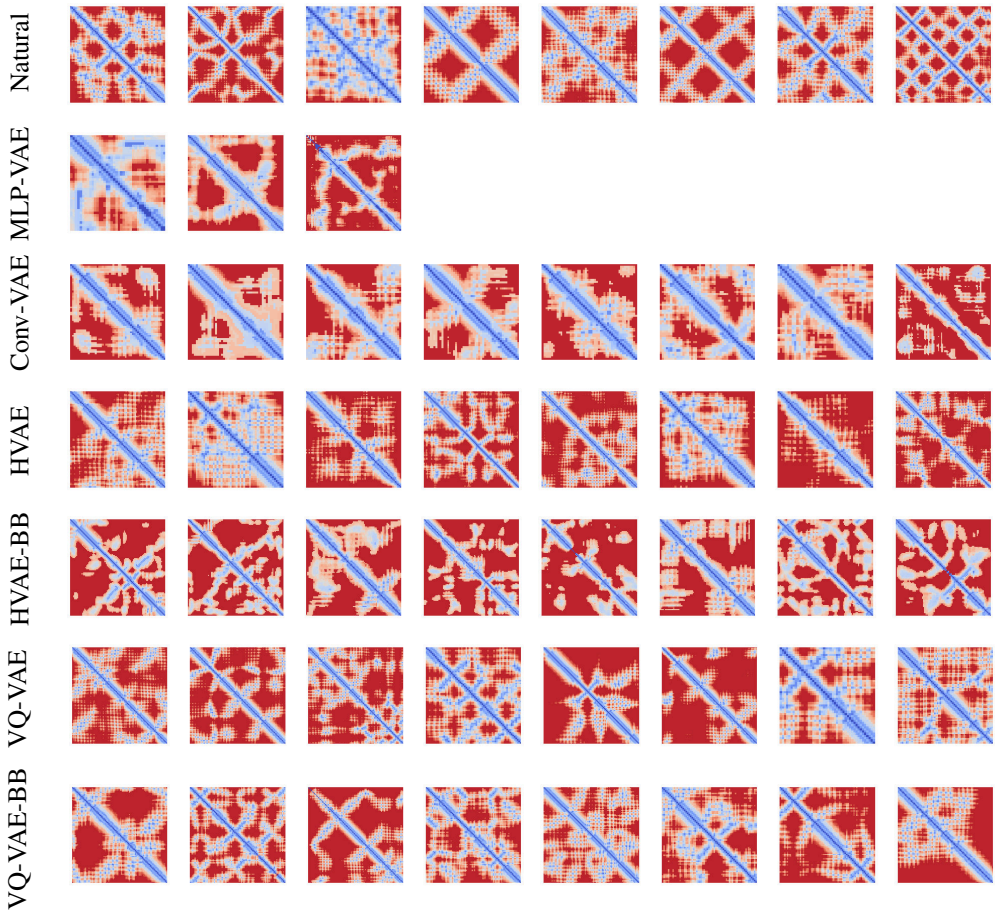

Figure S4: **Top Distograms**: we filter generations to negative rosetta energy and greater than 1 MR/LR contacts per residue. These are a random sample of such distograms

downsample, repeated 7 times until we obtain a 1×1 map, which is fed into the decoder as the initial latent variable. The HVAE and VQ-VAE have shortcut connections at coarser resolutions from the encoder to the decoder, where each shortcut connection is either a Vector Quantization layer or stochastic sampling layer via the reparameterization trick. These intermediate latent variables are output at up to a 32×32 resolution.

Model selection was done in all cases by a randomized hyperparameter sweep, generating 30 structures, and selecting the model with the lowest Rosetta energy of that model class.

The output of the decoder trunk is always routed to an individual 2 layer MLP to output the final predictions for each of the distogram and (interresidue or backbone) angle predictions. Backbone angles are predicted via averaging across the heights and width of the distogram, and then averaging the logits obtained from each of the height and width predictions.

We also found that for spatial latent variables, it was important to symmetrize the latent variables post-hoc, as well as symmetrizing the predictions via a simple average.

We explored the following hyperparameters for all models:

- Architecture (see next paragraphs). This is defined by the number of residual blocks per resolution ("stack" of residual blocks, between each of the 7 downsampling steps).
- Initial channel dimension (64, 128, 196, 256).
- Most resnets increase in the channel dimension as the encoder decreases in resolution, and vice versa for the decoder. We varied with which layer to start increasing the channel dimension. We start doubling the channel dimension at resolution (32×32, 16×16,8×8)

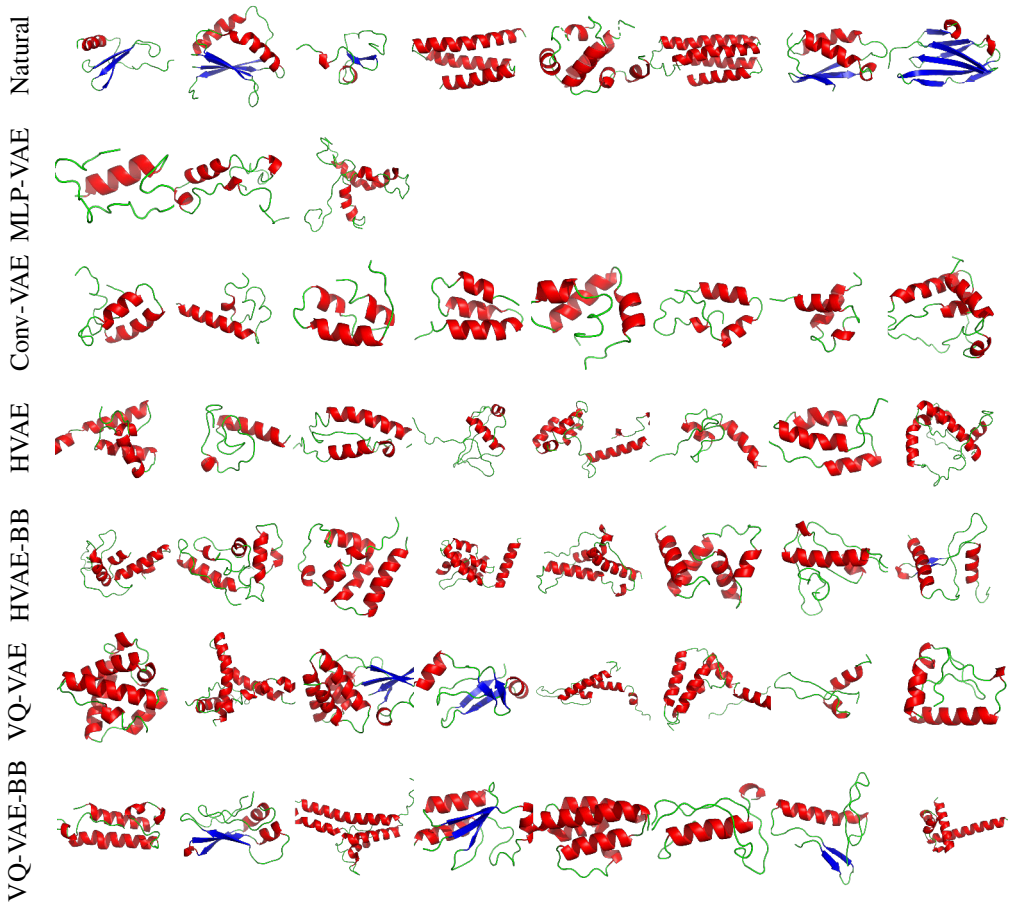

Figure S5: **Top Structures**: Corresponding to Figure S4, we show how each of the distograms fold under constrained folding. These structures are selected to have negative Rosetta energy and > 1 MR/LR contacts per residue We color α-helices red, β-sheets blue, and coils green. Only 3 such structures out of 3000 passed such a filter for MLP-VAE. Only VQ-VAE style models are able to generate many β-sheets.

- Parameters in distogram, angle, and sequence prediction branches. We tried a simple linear model, a MLP, and a few layers of dilated convolutions, as well as varying widths. We chose the MLP head in all cases.
- Representation of the output space. We also implemented the discrete logistic mixture distribution from Salimans et al. [50], but we choose the typical categorical distribution.
- We use Adam in all cases but sweep across different learning rates. We selected $e * 10^{-4}$ in all cases, as it seemed to train stably for all models.

Architecture specification In this section, we will specify the encoder and decoder with a list of 8 numbers, corresponding to the number of bottleneck residual blocks He et al. [51] between the 7 downsampling steps. The first number always refers to the blocks that work at a 128×128 resolution, and the last number always refers to the block that works at a 1×1 resolution. There is a downsampling operation between the stacks of blocks. We mainly varied whether the majority of the compute is at the coarse or fine resolutions. The final choices are specified below.

Conv-VAE The encoder is the same as MLP-VAE. The decoder consists of 61 convolutional blocks of size 3×3. We upsample and replicate the single latent variable to the 128×128 sized output concatenated with a Sinusoidal positional embedding. This model is 89M parameters.

For MLP-VAE and Conv-VAE baselines, we experimented with annealing the coefficient on the KL

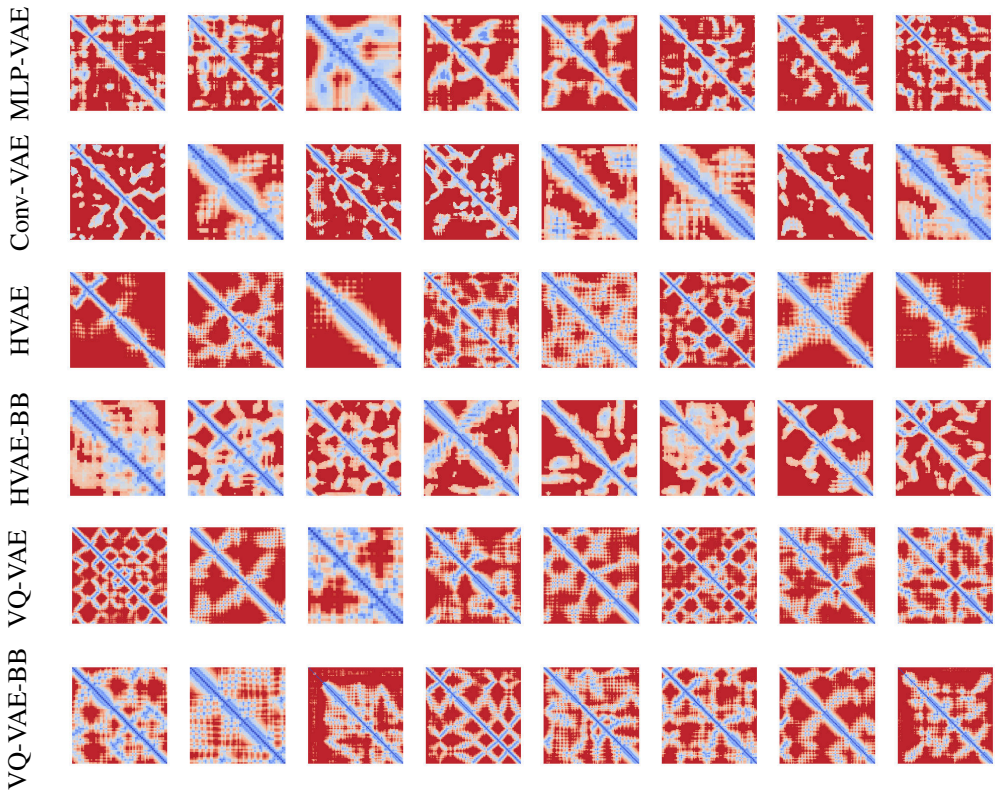

Figure S6: **Random distograms**: We show a random sample of all generated distograms by each model

divergence term from 0 at the start of training. We found that annealing was critical to stable training - the cyclic annealing schedule of Fu et al. [52] was helpful in stable model training across a range of hyperparameters, so this was used in all models reported, with 4 cycles through the first half of the training. We experimented with large models, batch sizes, learning rates, and latent variable sizes. We also experimented with VAE variants like β-VAE [53] and Deterministic Regularized Autoencoders [54], but found little effect on the generation quality.

MLP-VAE The encoder is a stack of convolutional blocks: `[3, 3, 3, 3, 3, 3]`, with an intial hidden size of 64 and doubling every time a downsample.

For MLP-VAE, we found it impossible to train with a flattened structure. Taking inspiration from Tolstikhin et al. [48], we run three residual MLP layers, across the height, width, and channel dimensions. We found that this was able to converge more consistently.

Define an residual MLP block as a sequence of (LayerNorm, Linear, GeLU, Linear) operations. A MLP stack is 3 MLP blocks, the first over the height dimension, the second over the width dimension, and the third over the channel dimension.

We upsample and replicate the single latent variable to the 128×128 sized output concatenated with a Sinusoidal positional embedding. The decoder is then 21 MLP stacks with a hidden dimension of 128. This model is 88M parameters.

HVAE The HVAE model, results in a step function in generation quality over previous models, generating a much larger proportion of α-helices.

In order to implement the hierachical VAE as specified in Vahdat and Kautz [55], Child [56], we used an upsampling convolutional decoder. Between each stack, there is an upsampling operation. In both HVAE and HVAE-BB, we output a 1×1 latent variable, and 2 sets of latent variables at a 4×4 resolution, one at the start of the stack and one at the end.

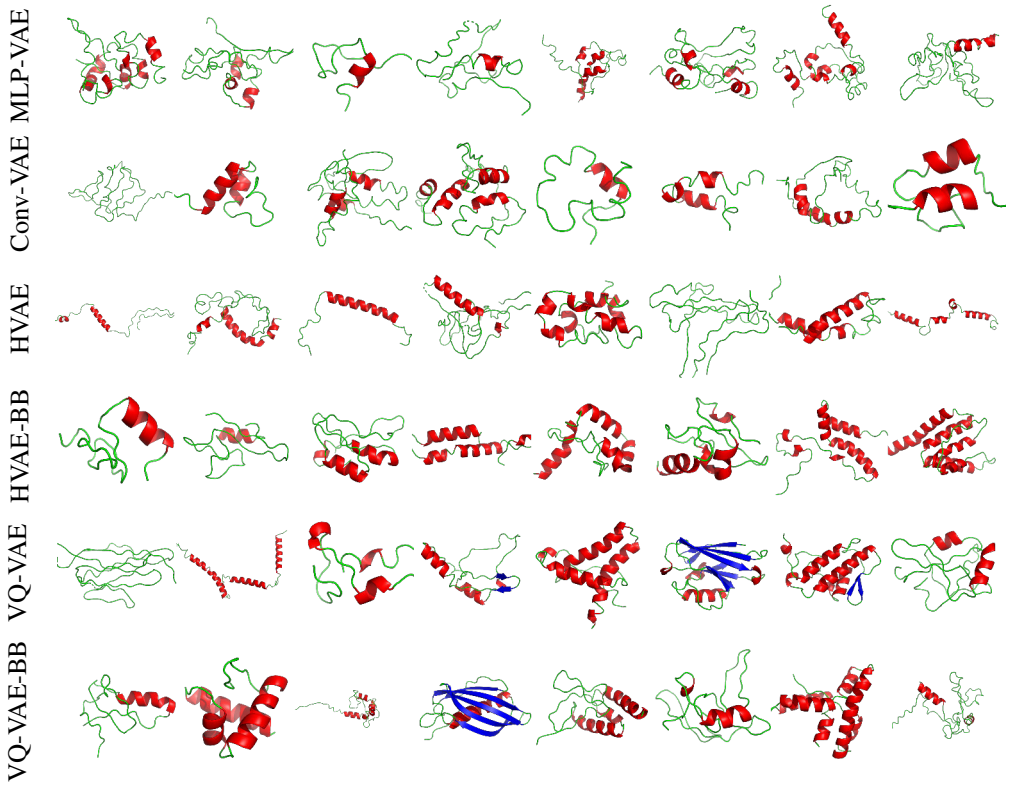

Figure S7: **Random structures**: We show, correspondingly to Figure S6, the results of constrained folding on those distograms. We color α-helices red, β-sheets blue, and coils green. Most baseline generations are unstructed coils.

The encoder for HVAE is `[4, 4, 4, 8, 8, 32, 4, 1]`. The decoder is `[4, 4, 4, 8, 12, 32, 4, 1]`. The model is 101M parameters.

The encoder for HVAE-BB is `[1, 1, 4, 8, 8, 24, 20, 1]`. The decoder is `[1, 4, 8, 12, 12, 28, 20, 1]`. The model is 115M parameters.

The model architectures were determined via random hyperparameter sweeps as specified above. We also tried to vary the number of output latent variables, trying up to 20. The reconstruction loss and ELBO did indeed improve with the number of latent variable scales, but the generation quality did not.

VQ-VAE Similar to the HVAE, we output a latent variable at specific scales. This latent variable is quantized through the Vector Quantization layer of Oord et al. [57].

For the decoder, we stack all encoded latent variables, upsampled to the finest resolution output by the encoder, and run it through the same upsampling convolutional blocks as described above. We use 512 quantized latent variables with dimensionality 64 in all cases.

For VQ-VAE, the encoder is `[2, 2, 2, 2, 2, 2, 2, 1]`, outputting latent variables at the 1×1 and 32×32 scales. The decoder is `[4, 4, 4, 4]`, since the input starts at the 32×32 scale. The model is 114M parameters.

For VQ-VAE-BB, the encoder is `[2, 2, 2, 2, 2, 2, 2, 1]`, outputting latent variables at the 1×1, 4×4 and 16×16 scales. The decoder is `[4, 4, 4, 4]`, since the input starts at the 32×32 scale. The model is 115M parameters.

In both cases, for $p_\theta(x)$, we use an autoregressive, 6 layer vanilla transformer decoder [58] to learn the prior. We use 6 layers with a channel dimension of 512 over 8 heads. This worked robustly, so we only searched through dropout in order to better regularize our models. However, we found that models overfit to the training set provided better generations, so we report only results with the overfit

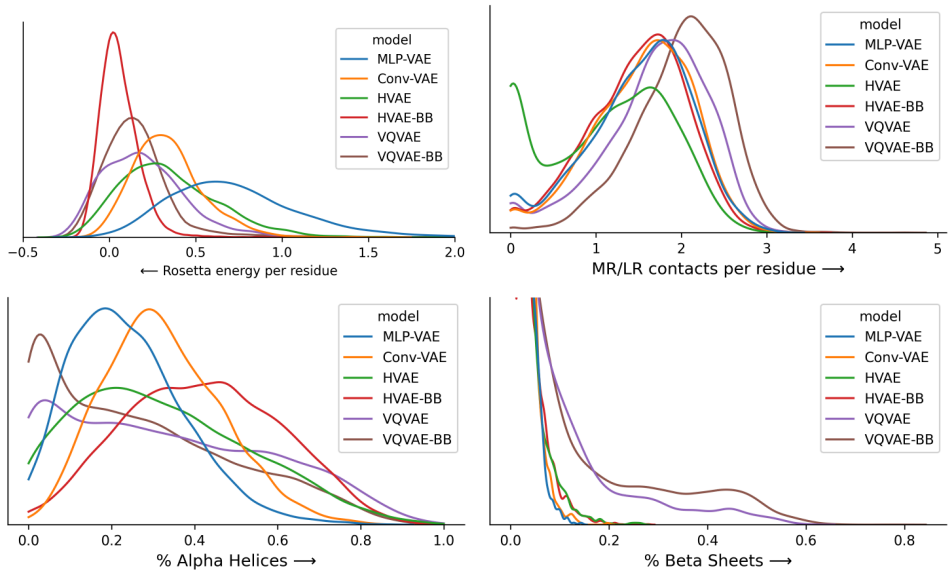

Figure S8: Gaussian kernel density estimate plot of the proposed metrics, for structures generated with different models.

priors in this work. More exploration to explain this phenomenon is needed. The learned priors are 20M parameters in all cases.

Training procedure We use stratified sampling at the family level during training. Stratified sampling has been shown to be critical to training large protein language models, though we did not explore uniform sampling [59]. We also tested results on fold and superfamily-level structural heldout sets, though we found that there was too much noise given the small dataset. Furthermore, since their losses were not correlated to Rosetta folding energies, so we decided not to report on their exact numerical values.

We fix models to work on 128 length proteins. Larger proteins are ignored, as we theorized that sequence-level cropping would compromise structural integrity. Smaller proteins are centered and padded.

Variable length generation We use a heuristic to generate structures of varying lengths. We fit a head on the amino acid identity of each position, and we train with a "out-of-bounds" label for padding tokens. Then, during generation, we noticed that the model consistently generated centered distograms. Therefore, we used the heuristic that the generated protein starts at the first position when $p(\text{out-of-bounds}) < .2$, and ends at the last position when $p(\text{out-of-bounds}) < .2$.

Compute Costs To find optimal hyperparameters, we used tens of GPUs for 2 days at a time. Most models can be trained on a single V100 GPU in a few days. Each `FastDesign` decoy takes from 3-6 hours to run on a single core. Therefore, one round of designs for 40 generations takes approximately 1,500 CPU days.

B.2 Structure generation details

Metrics The metrics presented in Table 1 are as follows.

$R(x)$ is from Rosetta's `ref2015` energy function after the constrained folding procedure. $\%R(x) < 0$ is the percent of all generations with negative Rosetta energy. In general, the more negative the energy, the better. Structures with positive energy tend to not be well formed, exhibiting disordered secondary structure.

We define a Medium or Long Range (MR/LR) contact to be when two residues σ_i, σ_j, have sequence separation in the amino acid chain $|i - j| > 12$ and a Cβ-Cβ distance of less than 8Å. We then

normalize this by the length of the protein and average to obtain the MR/LR contact score. The MR/LR polar contacts similar to Cβ-Cβ contacts, except that we use Kunzmann and Hamacher [60] to find hydrogen bonds via the Baker-Hubbaard algorithm [61] on the generated polyalanine chain. Both of these metrics are a measure of how compact and well-structured the generated proteins are.

Finally, % α-helix and % β-sheet measures the average proportion formed in generated structures. The categories are based on secondary structure assignment with DSSP [62]. These two secondary structure elements are a stabilizing part of protein structures, and generations without decent proportions of helices and sheets are disordered and are unlikely to exist in nature.

Constraints on interresidue distances and orientations We follow the procedure of Yang et al. [21] to incorporate constraints into Rosetta folding, with some variations:

1. Generate constraints given the distogram and interresidue angles. (Renormalize distrogram such that the minimum logit is -10).
2. Coarse grained folding (9 independent trajectories using centroid energy function). Followed by fine grained all-atom fold of all 9 trajectories.
3. Best trajectory selected by Rosetta energy.

Constraints on backbone dihedral angles Some models incorporate constraints on the backbone dihedral angles rather than the interresidue orientations. We discretize the phi/psi dihedral angles of the backbone into a 25 by 25 grid. Then 10000 samples are drawn from the grid, and a Von Mises distribution is fit independently to the marginals for phi and psi. The constraints are upweighted by a factor of 1000 to balance them with the distance contraints.

B.3 Design Verification

Because `FastDesign` is an expensive step requiring many monte-carlo runs, we only experimented with the best performing model and we try to filter our structures for the most likely to fold ones. Therefore, we sample 40 random structures that pass a goodness filter that we heuristically define as $R(x) < -0.1$, MR/LR contacts more than 1.5 per residue, and coils consisting of less than half the protein. This criterion only passes around 5% of the generations.

Given a generated structure, first we run `FastDesign` to generate up to 200 sequences. We run it with RosettaScripts [63], allowing all amino acids and extra rotamer angles with ALLAA EX 1 EX_CUTOFF 3. We also use the linear memory interaction graph to conserve memory, and default databases provided by Rosseta.

We run AlphaFold with default parameters. We randomly choose the predicted LDDT > 0.7 threshold - on average it means the model should be correct until up to 2Å. Only one structure that passes the predicted LDDT > 0.7 threshold did not agree with Alpafold. For many of the structures that Alphafold was not confident on, we discover close matches in PDB, so it may be the case that `FastDesign` was not able to find a suitable sequence for our generated structure. We found that AlphaFold predicted confident structures on only 9 of the sequences designed. Of these, 8 proteins had high (>.75) TM-scores to AlphaFold predicted structures. We show these proteins in Figure S10.

Figure S9 shows many of the designed structures have very low sequence identity with Uniref90. These are unfiltered designs, so it is unclear if they are all realizable, although in Figure 3 we show that most structures agreed on by our design and AlphaFold have no MSAs in common.

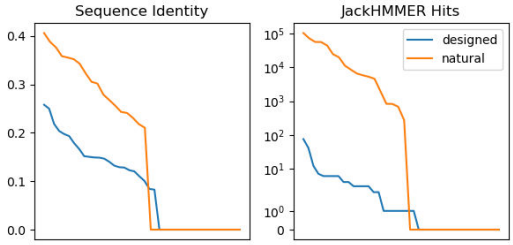

Figure S9: Behavior of `FastDesign` sequences on natural proteins vs our designs, both starting from polyalanine chains. Sequence identity is computed with respect to the closest homolog. Although both show around half have no nearby sequences in Uniref, `FastDesign` sequences for natural proteins tend to have many more homologs. Additionally, the sequence identity of homologs are much higher for natural proteins than generated proteins.

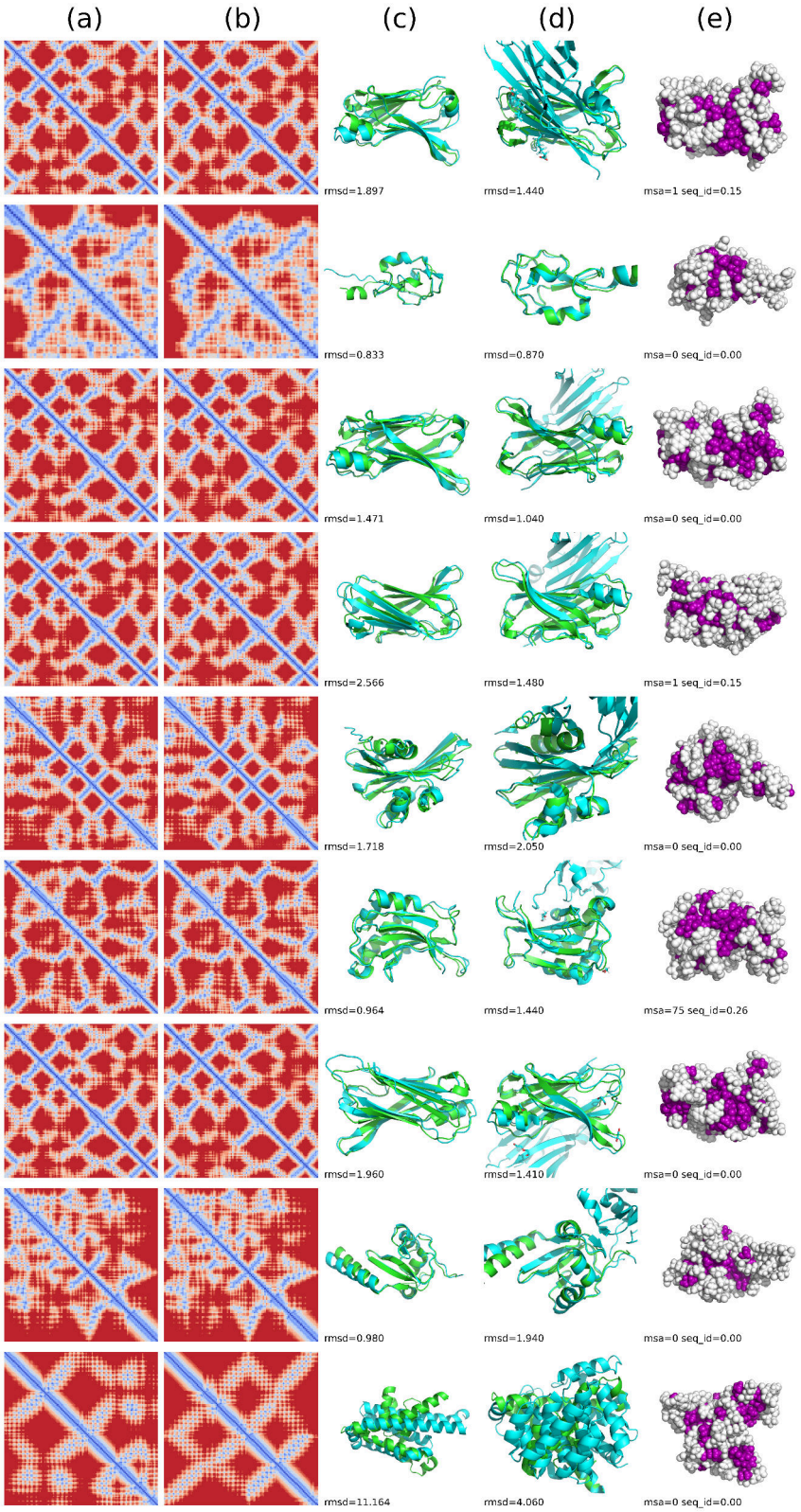

Figure S10: All examples for Figure 3. The last row is the only disagreement our model has with AlphaFold. Several designs are β-barrels of similar distograms - a weakness of our model as demonstrated in Figure 2. We note that although the argmax look exceeding similar, the full distogram distributions are different enough to produce different nearest neighbors in PDB for each of the structures.