Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models

Namrata Anand
namrata.anand2@gmail.com

Tudor Achim
tachim@cs.stanford.edu

Abstract

Proteins are macromolecules that mediate a significant fraction of the cellular processes that underlie life. An important task in bioengineering is designing proteins with specific 3D structures and chemical properties which enable targeted functions. To this end, we introduce a generative model of both protein structure and sequence that can operate at significantly larger scales than previous molecular generative modeling approaches. The model is learned entirely from experimental data and conditions its generation on a compact specification of protein topology to produce a full-atom backbone configuration as well as sequence and side-chain predictions. We demonstrate the quality of the model via qualitative and quantitative analysis of its samples. Videos of sampling trajectories are available at `https://nanand2.github.io/proteins`.

1 Introduction

Proteins are large macromolecules that play fundamental roles in nearly all cellular processes. Two key scientific challenges related to these molecules are characterizing the set of all naturally-occurring proteins based on sequences collected at scale and designing new proteins whose structure and sequence achieve functional goals specified by the researcher. Recently, AlphaFold2, a purely data-driven machine learning approach, has shown great progress in the forward problem of structure prediction [23]. Similarly, machine learning approaches have come to perform well for the sequence generation inverse problem [3, 21, 20]. However, for the task of structure generation, stochastic search algorithms based on handcrafted energy functions and heuristic sampling approaches are still in wide use [25, 34, 2, 26].

Data-driven generative modeling approaches have not yet had the same impact in the protein modeling setting as they have in the image generation setting because of several key differences. First, unlike images, proteins do not have a natural representation on a discretized grid that is amenable to straightforward application of existing generative models. Interpreting the pairwise distance matrix of a protein's atoms as an image to be modeled with existing models has seen limited success because inconsistencies in the predictions lead to nontrivial errors when optimization routines are used to recover the final 3D structures [4]. Second, unlike images, proteins have no natural canonical orientation. As a result, methods that are not rotationally invariant must account for this factor of variation directly in the model weights, which reduces the effective model capacity that can be dedicated to the structural variation of interest. Finally, in protein generation, nontrivial errors in local *or* global structure lead to implausible protein structures.

Previous work has made progress on different aspects of the problem. Rotamer packing has benefited from machine learning approaches [28, 14, 1, 32, 3]. Machine learning has also made an impact on sequence design both in the case of conditioning on structural information [3, 21], and without [17, 9, 37, 38, 16, 18, 31, 20]. However, 3D molecular structure generation is a more challenging

Preprint. Under review.

problem, and existing methods have either been limited to the generation of small molecules [39] or to large proteins in highly restricted settings with only one domain topology [15].

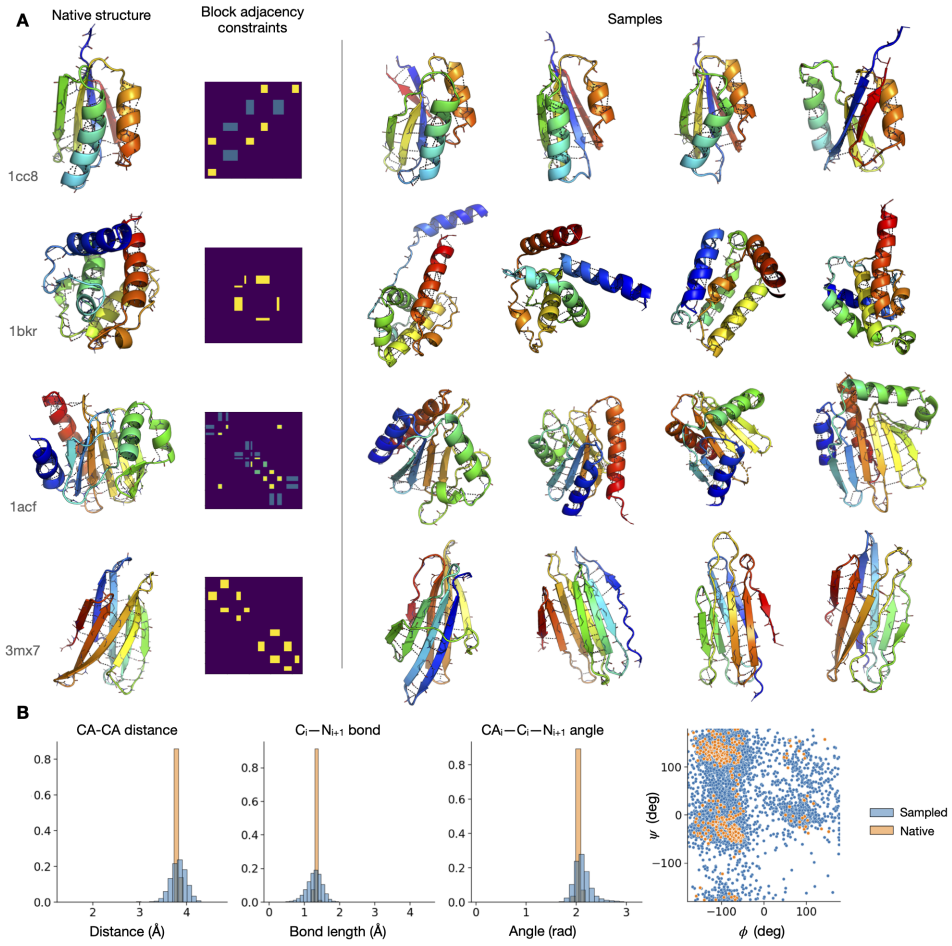

Figure 1: From-scratch protein generation. Section A shows four different sampling scenarios. We show the block adjacencies in the middle and a test set structure matching the adjacencies on the left. On the right are four different samples from the model with no post-processing. The samples show a high degree of variability and excellent hydrogen bond patterns (the dashed lines) within helices and between beta sheets. Section B compares the distributions of bond lengths and angles and backbone torsions for the generated structures relative to native crystal structures.

In this paper we present a new generative model that makes significant progress on closing this gap. We introduce a fully data-driven denoising diffusion probabilistic model (diffusion model) for protein structure, sequence, and rotamers that is able to generate highly realistic proteins across the full range of domains in the Protein DataBank (PDB) [6]. For comparison, protein macromolecules have approximately $100 - 1000\times$ the atom count of the small molecules addressed by previous molecular generative models, and the full set of domain types in the PDB numbers in the hundreds, in contrast to the single domain types addressed in previous work. Our model is equivariant to rotations and translations using invariant point attention (IPA) modules introduced in [23]. To handle the diffusion of rotational frames of reference that are required for protein generation, we use a formulation that leverages an interpolation scheme well-suited to $SO(3)$. For discrete sequence generation, we use an approach akin to masked language modeling that can be interpreted as diffusion in a discrete state space [5]. Finally, to allow for interactive structure generation, we introduce a compact set of constraints that the model conditions on to generate proteins. We show in Figure 1 and Section 3 that the model is able to generate high quality structures and sequences with nontrivial variety. To the best of our knowledge this is the first generative model that is capable of synthesizing physically

plausible large protein structures and sequences across the full range of experimentally characterized protein domain topologies.

2 Approach

We first briefly describe the protein modeling problem and denoising diffusion probabilistic models. Following that, we describe ways in which the diffusion training process is adapted to handling rotations as well as discrete (sequence) sub-problems. Finally, we conclude the Approach section by introducing a compact encoding scheme for constraints that we use in conditional sampling of proteins, and summarize the training and sampling procedures.

2.1 Preliminaries

Proteins Proteins are macromolecules made up of chains of amino acids. The protein backbone consists of repeating atoms $N - C_\alpha - C - O$, with side-chains branching off the backbone from the C_α atom. Each group of backbone atoms with its associated side-chain is referred to as a residue. Interactions between the side-chains, the protein backbone, and the environment give rise to local secondary structure elements – such as helices, strands, or loops – and to the ultimate tertiary structure of the protein. The 3D locations of the backbone and side-chain atoms fully describe the protein structure, and there are several priors that constrain the distribution of the atom locations.

First, the ordering of the backbone atoms from $N-$ to $C-$terminus is fixed, and bond lengths and angles between atoms on the backbone do not deviate much from average values. Second, the secondary structure of the protein is captured by the torsion angles ϕ, ψ of the backbone, which follow an established distribution known as the Ramachandran distribution. Alternatively, the C_α can be interpreted as forming a canonical orientation frame in relation to the N, C, and O atoms, and the backbone atom positions as well as the torsion angles can be derived from these canonical frames. Third, atomic configurations of the 20 different amino acid side-chains can be described by some prefix of torsion angles $\chi_1, \chi_2, \chi_3, \chi_4$, which also follow amino-acid specific distributions. Although there are many experimentally- and theoretically-informed biophysical and statistical models for all of these quantities, we do not use them in training. In Section 3, we measure how well our generative models recover these priors from the data.

Summarizing the above, assuming an N-residue protein, our goal is to learn a generative prior over the following variables:

- $x^i_{C_\alpha} \in \mathbb{R}^3$ for $i \in \{1, \dots, N\}$, the 3D coordinates of the C_α backbone atoms.
- $q^i \in SU(2)$, the unit quaternion defining the global rotation of the canonical frame centered at $x^i_{C_\alpha}$. Using q^i and $x^i_{C_\alpha}$ we can recover the positions of the associated N, C, O atoms in closed form and by extension the backbone torsion angles.
- $r^i \in \{1, \dots, 20\}$, the amino acid type of the i^{th} residue.
- $\chi^i_1, \chi^i_2, \chi^i_3, \chi^i_4 \in [-\pi, \pi)$, the four χ angles for the side-chain attached to the i^{th} C_α atom. Note that some side-chains are made up of fewer atoms and thus have only a proper prefix of these angles.

Diffusion Models Diffusion models are a class of latent variable models that model the data generation process as iterative denoising of a random prior, with a specific parameterization of the approximate posterior distribution that can be interpreted as "diffusing" toward the fixed prior distribution [36]. We briefly review the formulation below. The data generation (reverse) process for a datapoint x^0 sampled from the data distribution $q(x^0)$ is defined recursively with a transition kernel p_θ and prior distribution π:

$$p_\theta\left(x^T\right) = \pi\left(x^T\right) \qquad p_\theta\left(x^0\right) = \int_{x^{1:T}} p_\theta\left(x^T\right) \prod_{t=1}^{T} p_\theta\left(x^{t-1} | x^t\right) \tag{1}$$

The approximate posterior, referred to as the forward process, in the continuous case diffuses the datapoint x^0 toward the random prior:

$$q\left(x^{1:T}|x_0\right) = \prod_{t=1}^{T} \mathcal{N}\left(x^t; \sqrt{1-\beta_t}x^{t-1}, \beta_t I\right) \tag{2}$$

where the β_t are chosen according to a fixed variance schedule. We use a neural network μ_θ to parameterize the reverse transition kernel: $p_\theta(x^{t-1}|x^t) = \mathcal{N}(x^{t-1}; \mu_\theta(x^t, t), \sigma_t^2 I)$. We obtain μ_θ by minimizing the following variational bound during training, following [19]:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,x^0}\left[\mathcal{L}_{\text{FAPE}}(x^0, \mu_\theta(x^t, t))\right] \tag{3}$$

where x^t is obtained by noising x^0 by q, and the rotationally invariant loss function $\mathcal{L}_{\text{FAPE}}$ is described in Section 2.2. Finally, sampling relies on the learned μ_θ to execute a reverse process with maps a sample from the prior distribution to a sample from the data distribution.

There are important differences between the image and protein generation settings which impact the architecture of μ_θ as well as the training and sampling algorithms. The first we cover in the next section, and the second in Section 2.5 and Table 1.

2.2 Equivariant Diffusion Training

Diffusing Rotations Unlike coordinates, our rotation variables q^i and $\chi^i_{1:4}$ do not live on Euclidean manifolds with flat geometry; therefore during training and sampling they cannot be diffused towards their prior distribution simply by randomly scaling and perturbing their encoding as is the case with coordinates. To address such limitations, recent work has extended the diffusion framework to compact Riemannian manifolds [11], which has been in turn adapted to modeling rotational diffusion as the repeated application of a heat kernel on a torus [22]. However, our experiments indicated that a simpler method suffices in practice. For our prior distribution π_q we sample uniform random rotations from $SU(2)$. Next, instead of diffusing from x^0 towards π_q with Brownian motion and thus modifying the reverse process to use an Euler-Maruyama sampler [11], we *interpolate* from x^0 to a sample $\epsilon \sim \pi_q$ based on the schedule of variances (see Table 1). We choose spherical linear interpolation ($\mathsf{SLERP}(x, y, \alpha)$, where we interpolate from x to y by a factor of $\alpha \in [0, 1]$) [35]. For rotamer torsion angle diffusion (1D rotations) we use a uniform prior over $\mathcal{S}^1$ and interpolate on the unit circle for noising and sampling ("Interp"). These design choices have the desired effect of exposing the network to a similar distribution of random rotations both at training and test time, and our experiments demonstrate that they work well in practice.

Rotational Invariance As mentioned in the introduction, one key difference between images and proteins is that proteins have no canonical orientation. As a result, we use an *equivariant* transformer for our denoising model μ_θ. The model takes as input an intermediate protein structure, x^t and produces an estimate of the denoised ground truth structure $\hat{x}^0$. We replace the standard attention mechanism in the transformer [7] with invariant point attention (IPA) as described in [23]. Put simply, IPA partitions node query and value features into 3-dimensional vectors and transforms them from the target node's reference frame into the a global reference frame before computing both attention weights and the output of the attention mechanism. The output of the attention layer is invariant to the global orientation of the input protein, and thus the resulting corrections predicted by μ_θ in the local coordinate frames of the C_αs are equivariant.

Training μ_θ requires a loss function that can stably account for errors in all of the predictions of the generative model. Therefore we follow [23] and use the frame-aligned point error (FAPE) loss. FAPE penalizes errors in rotation by aligning the predicted local transformation $(q^i, {x^i}_{C_\alpha})$ with the ground-truth local frame *at each residue in turn* and computing the clamped squared distance between the ground-truth and predicted atoms. Because coordinate frames are aligned when computing loss, the training procedure is invariant to the orientation of protein structures in the dataset.

2.3 Discrete Sequence Diffusion

We use an approach akin to a masked language model [12] to generate the sequences on top of the backbones, which can be interpreted as diffusion with an absorbing state [5]. Concretely, we train the model by randomly masking a fraction of the residues, where the fraction is linearly interpolated in $[0, 1]$ during training as a function of t. At test time, we run the reverse process by masking

all residues at $t = T$, and iteratively sampling from a model whose input residues are masked independently with probability t/T, with t stepping from $t = T$ to $t = 0$ during sampling.

2.4 Constraints

Besides encoding a manifold on which relevant inverse problems can be solved, the main value of a generative model for protein structures and sequences is in allowing a researcher to specify simple, compact conditioning information encoding their desired structural properties, sample many valid protein configurations based on that, and iterate on the conditioning information until the desired results are obtained. We introduce below one such constraint specification.

Given the secondary structure topology of a protein, the residues can be divided into contiguous adjacent blocks based on the secondary structure of the block (i.e. each block corresponds to either a helix, a beta sheet, or a loop of some length). Furthermore, each pair of blocks can be considered to be adjacent or non-adjacent based on whether or not their closest atoms are within some distance threshold. For paired beta sheets, besides adjacency we can specify whether the sheets are parallel or anti-parallel to each other. Therefore, one way to compactly describe a protein is by specifying a number of residues N, then a tuple of numbers of length B adding up to N which indicate the block sizes, then a block secondary structure assignment $\{\text{helix}, \text{sheet}, \text{loop}\}^B$, and finally a symmetric block adjacency matrix in $\{0, 1\}^{B \times B}$ together with a parallel/anti-parallel prior on each beta sheet pairing. In practice, we allow for the underspecification of the block adjacency matrix by dropping out block adjacencies during training and by not including any adjacency information for loop blocks by default. This coarse encoding scheme for the topology prior does not overly constrain the model to produce just one structure; it allows for significant variation as seen in Section 3.1.

2.5 Training and Sampling Summary

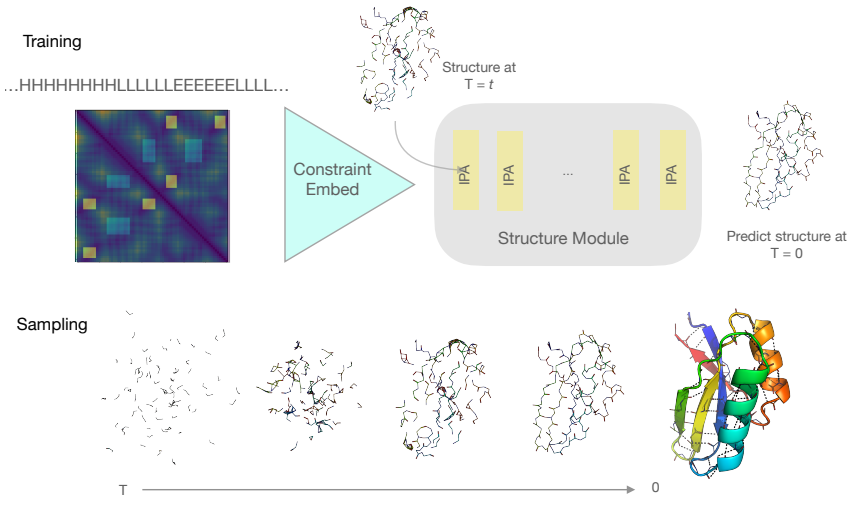

Figure 2: The model receives as input secondary structure and coarse constraints (shown here overlaid on the C_α distance matrix of the ground truth structure). The constraints are used by the IPA modules during training to denoise the diffused input. At test time, the model produces samples that adhere to the coarse constraints.

We summarize the overall design choices here and in Figure 2. The generative model conditions on a compact specification of constraints for a given protein, described in Section 2.4. These constraints are embedded using a transformer with triangular self-attention to produce feature embeddings which are processed using Invariant Point Attention to produce updates to the translations, rotations, and residues in the local coordinate frames of the C_α atoms. These updates are used in training to compute rotationally-invariant losses, and in sampling to take steps toward the final structure.

In Table 1 we summarize for each variable the prior distribution, the approach used to interpolate between the data distribution and the noise distribution, and finally the method for taking a step

at sampling time. To generate a structure, we sample a starting point from the prior distribution corresponding to $t = T$, and iteratively apply the update described in the "Sample Step" column in Table 1 for all variables for $t = T$ down to $t = 1$. The sample of the generative model is taken to be the value at $t = 0$.

Table 1: Diffusion Process Hyperparameters

Variable	Prior Distribution (π)	Training Noising (step t)	Sample Step (step t)
$x^i_{C_\alpha}$	$\mathcal{N}(0, I)$	Diffusion with x_0 prediction [19] and cosine schedule [13]	
q^i	$\mathsf{Uniform}(SU(2))$	$q^i_t = \mathsf{SLERP}(q^i_0, q^i_T, \alpha_t)$	$q^i_{t-1} = \mathsf{SLERP}(\hat{q}^i_0, q^i_t, (1 - \alpha_t))$
$\chi^i_{1:4}$	$\mathsf{Uniform}(-\pi, \pi)$	$\chi^i_t = \mathsf{Interp}(\chi_0, \chi_T, \gamma_t)$	$\chi^i_{t-1} = \mathsf{Interp}(\chi^i_t, \hat{\chi}^i_0, (1 - \gamma_t))$
r^i	Fully Masked	Mask each residue with probability t/T; predict and incur loss on masked.	Mask each residue with probability t/T; predict masked.

3 Experiments

We train our model on X-ray crystal structure data of CATH 4.2 S95 domains [27, 10] from the Protein Data Bank (PDB) [6]. We separate domains into train and test sets based on CATH topology classes, splitting classes into $\sim 95\%$ and 5%, respectively (1374 and 78 classes, 53414 and 4372 domains each). This largely eliminates sequence and structural redundancy between the datasets, which enables evaluation of the approach's ability to generalize.

Constraint Embedding Model The secondary structure information is encoded via a 1D GPT-3-like architecture [8]. Pairwise secondary structure embeddings and block adjacencies are then downsampled and passed through triangle multiplication and attention layers as in [23]. The full details of the model architecture can be found in the Supplementary Material.

Diffusion Model μ_θ The diffusion model conditions on the output of the constraint network and the current structure and produces a guess for the final structure configuration. When predicting $\hat{x}_{C_\alpha}$ and $\hat{q}$, each IPA module produces an intermediate backbone update which we apply to the structure before computing the next round of IPA. The full details of the model architecture may be found in the Supplementary Material.

Training and Sampling During training we use the prior distributions and noising procedures described in Table 1, sampling t uniformly at random in $[1, T]$. We use the AdamW optimizer and a cosine learning rate decay schedule [24, 30, 29]. The models are trained on single K80 and V100 GPUs on Google Cloud.

In the following three sections we use three separate models for structure (x_{C_α} and q), sequence (r), and rotamer (χ) diffusion. The structure model is trained using ground truth centered x_{C_α} coordinates that are scaled down $15\times$. The sequence model is trained on ground truth structures, and the rotamer model is trained on ground truth structures and sequences.

We share results on context-free generation, protein completion, and sequence design and rotamer repacking. There is *no post-processing* on the samples produced; all results are based on the raw output of the diffusion process at $T = 0$.

3.1 Context-free Generation

We begin our analysis of our approach by assessing its performance on the task of synthesizing accurate 3D designs of proteins, relying just on the compact specification of the protein. This task is difficult because the model must produce a physically plausible structure that also respects the coarse adjacency priors. To assess the degree of generalization of the algorithm, we compare against native backbones from the test set which have CATH-defined topologies not seen by the model during

training. We select four test case backbones that span the major CATH classes—all alpha, alpha–beta, and all-beta.

We can see in Figure 1 that the model is able to produce structures that are highly variable and physically plausible. In section A we show the test case native backbones and, for each one, we show the block adjacency and parallel/anti-parallel constraints as well as four high-fidelity samples from the model. The samples are of high quality, with intra-backbone hydrogen bonds forming within the helices as well as between the beta strands. The beta sheets are especially challenging to synthesize because the local structure needs to be precisely correct for the bonds to form, which in turn imposes constraints on the global structure to support the positioning of the sheets. In Figure 7, we show random samples for completeness.

Quantitatively, the charts in section B indicate that the model has learned biophysical priors of proteins directly from the data distribution. The various bond lengths and angles show good histogram overlap between the native and sampled structures. The distribution of generated backbone torsion angles are consistent with the Ramachandran distribution [33].

3.2 Inpainting and Controllable Generation

We find that the model is also suitable for the task of completing existing proteins in novel ways. For this task we train an additional model μ_θ to condition on existing structures by holding parts of the structure fixed during training and executing the forward diffusion process on the complement of the fixed parts. For all residues that do not diffuse toward the prior, their position is held fixed at their ground truth positions during training and during sampling. Figure 3 shows that the distribution of bond geometries for the inpainted regions is consistent with the corresponding distribution in the native structure. We also see from the samples that the model can find discrete modes of the loop distribution at the atomic level. Random samples per test case are given in Figure 8.

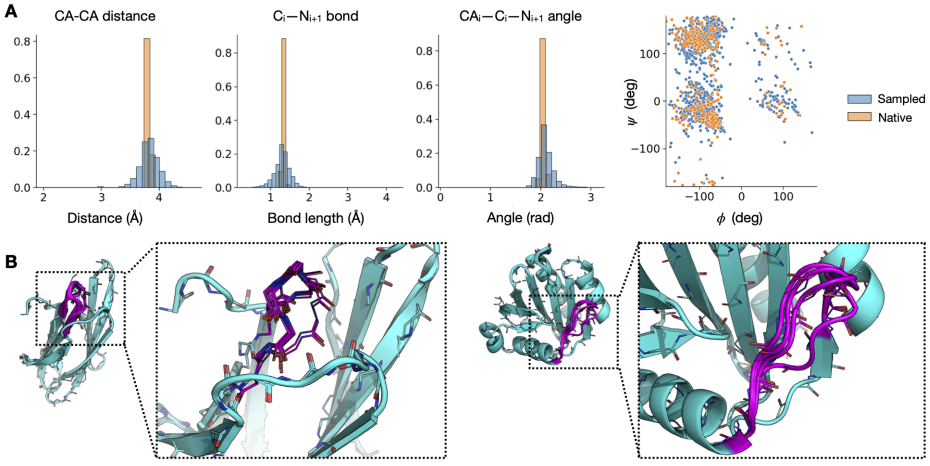

Figure 3: Inpainting and loop design: Section A compares the distributions of bond lengths and angles and backbone torsions for the completed regions relative to the native structures. Section B shows examples of loop completions highlighted in purple. The image on the left highlights the model's ability to find discrete modes of the possible loop configurations.

We explore whether the model can go beyond sampling variants of topologies of existing proteins, to modifying the topologies themselves. In this case, we use the same underlying μ_θ model as for the inpainting case but at sampling time we modify the block adjacency conditioning information by simply modifying the lengths of underlying secondary structures for blocks. In Figure 4, we show how the model can be used to modify the structures in physically plausible ways – generating idealized topologies, altering loop lengths, and modifying secondary structure lengths for a fixed topology. We emphasize that these synthetic structures are distinct from the natural structures found in the PDB, which suggests that the model has encoded useful physical priors for use in sampling. In Figure 9, we show additional random TIM-barrel samples, and in Figure 10 we provide additional examples of model contextual design of variable-length loops.

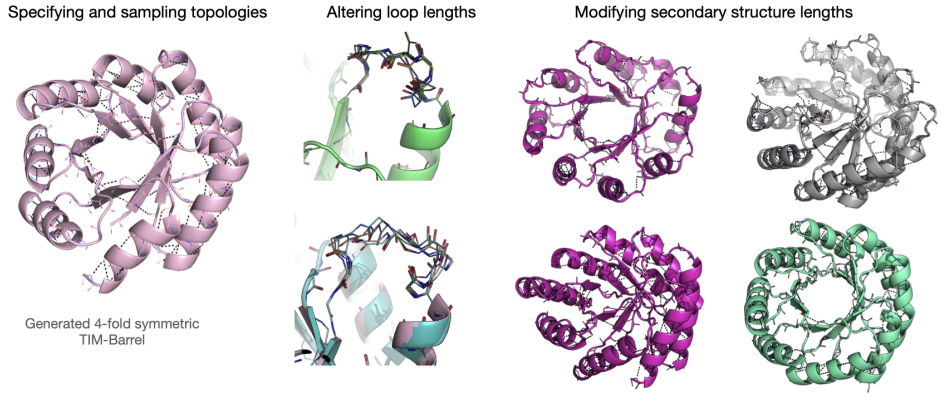

Figure 4: Controllable generation: The model enables many modes of controllable generation of protein structure. Here, we highlight (A) new and/or idealized topology generation, (B) loop engineering, and (C) secondary structure modification.

3.3 Sequence Design and Rotamer Packing

We see similarly strong performance for sequence design and rotamer packing as for structure generation. We measure the model's ability to recover ground truth sequences and rotamer configurations on native structures, because the physical variation in sampled structures implies a different set of optimal residues and rotamer configurations which cannot be compared to the ground truth directly. The sequence recovery rates are compared across 50 sampled sequences, each starting from the native full-atom backbone with no side-chain information. In Figure 5 we see that the model has comparable sequence recovery performance to baselines [3]. The 3DConv baseline refers to a machine learning approach for sequence design and rotamer packing using 3D convolutions [3]. RosettaFixBB and RosettaRelBB are baselines using heuristic energy functions; RosettaFixBB holds the backbone fixed during sequence sampling, which is the same setting as our model, and RosettaRelBB allows it to vary slightly in a "relaxation" procedure [26]. The rotamer packing performance is comparable at the most stringent metric cutoffs (5 and 10 degrees).

3.4 Joint Modeling

In the previous sections we considered one model each for structure, sequence, and rotamer diffusion. We now compare to a model trained to jointly diffuse structure and sequence concurrently. Structure variables x_{C_α} and q are diffused for the full $T_{structure} = 1000$ steps with the diffusion training and sampling approaches described in Section 2.5. The sequence variables r are diffused from $T_{sequence} = 100$ to $T = 0$, with an additional network that conditions on the output of the structure component of μ_θ at each step. That is, for a given $0 \leq t \leq 100$, we perform masked prediction of the sequence using the schedule in Table 1, conditioning on the prediction of $\hat{x}^0_{C_\alpha}$ and $\hat{q}^0$ from the structure network. Rotamer diffusion is then run on the sampled backbone and sequence.

In Figure 6a, we do contextual inpainting of both the backbone and sequence and find that the model is able to at times nearly recover the native solution both in terms of native sequence and backbone atom positions for inpainted regions. This type of a model enables us to do, for example, full-atom loop generation (Figure 6b) where we generate both the loop backbone and candidate sequence for the loop region jointly. This capability opens up the avenue to interesting engineering problems, such as immunoglobulin (Ig) loop design. Antibody variable Ig domains host highly variable CDR (complementarity-determining region) loops that allow them to selectively bind practically any target. In Figure 6c, we demonstrate how this type of generative model can be used to vary the CDR backbone loops and sequence jointly on a fixed Ig backbone.

Ultimately, we want to be able to sample jointly over backbone and sequence in a way that is self-consistent – namely, that the generated sequence folds to the generated backbone structure. Although we can jointly sample structures and corresponding sequences from scratch with this approach, sample

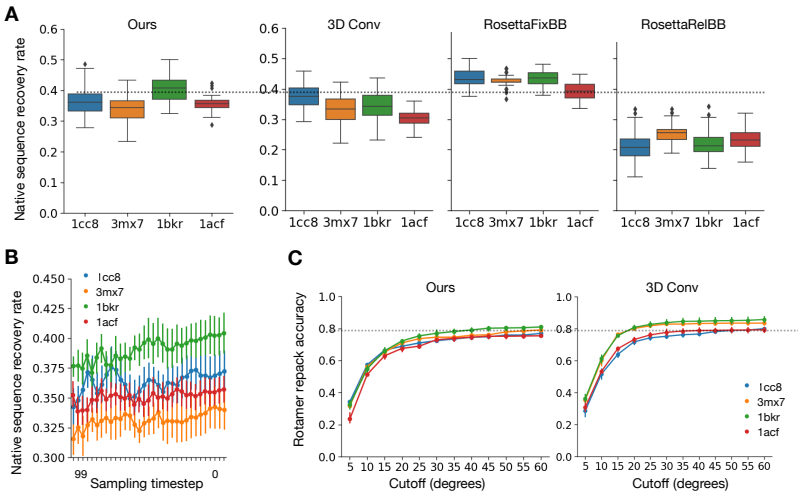

Figure 5: Sequence design and rotamer repacking. Section A reports native sequence recovery rates across 50 designs for test case structures. We reproduce design baselines reported in [3]. Section B shows the sequence recovery rate during the sampling trajectory, starting from predicting from all masked tokens. Section C shows the rotamer packing accuracy after χ diffusion as a function of degree cutoff, baselined against data reported in [3]. This approach to sequence design and rotamer packing is comparable to baselines and faster by an order of magnitude.

quality will likely improve once we allow the network to cross-condition on structure, sequence, and rotamers, rather than generate them sequentially, as in the current set-up. We leave exploration of models of this type to future work.

4 Conclusions

We have introduced a generative model of protein structures, sequences, and rotamers that can produce large protein domains that are both physically plausible and highly varied across domain types in the PDB. To this end, we designed a compact constraint specification which our model conditions on to produce highly-varied proteins. We demonstrated the model's performance both qualitatively and quantitatively using biophysical metrics and showed its potential for modifying existing proteins, from designing loops to varying the underlying topology. We concluded with an analysis of the model's ability to design sequences and pack rotamers, indicating its potential as a fully end-to-end tool for protein design.

There are many interesting areas for further exploration. First, one may replace the supervised learning "recycling" procedure for predictions in AlphaFold2 with the diffusion formulation in this paper (or, equivalently, replace the Constraints conditioning information with the output of the Evoformer blocks from [23]). In predicting the structure of a protein there is often nontrivial aleatoric uncertainty, which arises from the fact that that there are often many conformations that the protein could adopt, of which we only observe one via crystallography. Our model enables a simple way of quantifying uncertainty, via measurement of the spread of samples, which may be of interest to practitioners as an additional signal beyond the per-residue uncertainty quantification made available by AlphaFold2 [23].

Second, auxiliary energy functions could be used to interactively guide sampling for more fine-grained control over the sampling process. The current constraint specification is intentionally compact to allow for easy specification as well as wide variance in the generated structures. However, the gradients of simple energy functions (such as ones that penalize deviation from distance constraints) could guide more precise modifications during sampling.

Third, we anticipate natural applications of this type of model to problems in rational design and structure determination. Adaptations of these models could be effective for direct synthesis of proteins in protein-protein complexes. Moreover, we anticipate natural applications of this type of

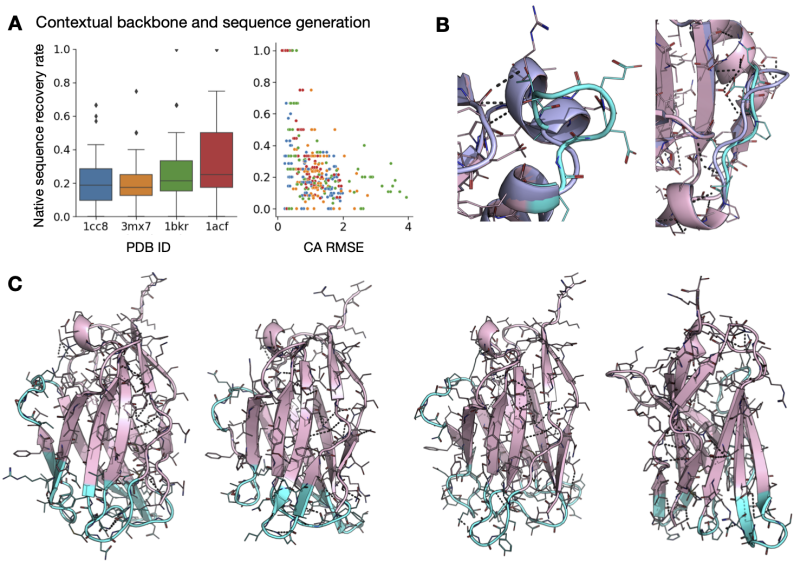

Figure 6: Contextual joint sampling of sequence and structure. For all cases shown, rotamers are packed with the rotamer diffusion model. Section A reports native sequence recovery rates and C_α RMSE after inpainting masked regions of test case proteins and sampling both backbone and sequence. While we do not expect perfect sequence recovery, we see that for some cases the model can nearly recover the native loop and sequence. Section B shows examples of model generated loops and sequences (cyan) with the native backbone (purple) for context. In Section C, given a fixed immunoglobulin backbone and sequence (pink), we sample variable-length loops and residues (cyan) jointly.

model in fitting proteins to Cryo-EM volumes. Current approaches typically use auto-regressive methods to iteratively fit structures and can face difficulties when the volumes are ambiguous and significant backtracking and search is necessary to correct for mistakes in the fitting process. Our protein diffusion model, which forms the structure globally during sampling, may help mitigate these failure modes.

While it is highly beneficial for researchers to have access to better models that can inform the design of therapeutics, vaccines, and more, there are indeed risks associated with having powerful tools for protein design. It may be the case that the accelerated rate of progress in computational methods brought on by the application of AI techniques makes it more difficult for the research to community to self-regulate in response to rapid changes in method capabilities.

Acknowledgements

We thank Jonathan Ho for helpful discussion on diffusion models.

References

[1] Deniz Akpinaroglu, Jeffrey A Ruffolo, Sai Pooja Mahajan, and Jeffrey J Gray. Improved antibody structure prediction by deep learning of side chain conformations. *BioRxiv*, 2021.

[2] Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

[3] Namrata Anand, Raphael Eguchi, Irimpan I. Mathews, Carla P. Perez, Alexander Derry, Russ B. Altman, and Po-Ssu Huang. Protein sequence design with a learned potential. *Nature Communications*, 13(1):746, February 2022. Number: 1 Publisher: Nature Publishing Group.

[4] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in neural information processing systems*, 31, 2018.

[5] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. Technical Report arXiv:2107.03006, arXiv, July 2021. arXiv:2107.03006 [cs] type: article.

[6] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Egbert Castro, Abhinav Godavarthi, Julian Rubinfien, Kevin B Givechian, Dhananjay Bhaskar, and Smita Krishnaswamy. Guided generative protein design using regularized transformers. *arXiv preprint arXiv:2201.09948*, 2022.

[10] Natalie L Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, David Lee, Paul Ashford, Christine A Orengo, and Ian Sillitoe. Cath: an expanded resource to predict protein function through structure and sequence. *Nucleic acids research*, 45(D1):D289–D295, 2016.

[11] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian Score-Based Generative Modeling. *arXiv:2202.02763 [cs, math, stat]*, February 2022. arXiv: 2202.02763.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[14] Yilun Du, Joshua Meier, Jerry Ma, Rob Fergus, and Alexander Rives. Energy-based models for atomic-resolution protein conformations. *arXiv preprint arXiv:2004.13167*, 2020.

[15] 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*, 2020.

[16] Noelia Ferruz and Birte Höcker. Towards controllable protein design with conditional transformers. *arXiv preprint arXiv:2201.07338*, 2022.

[17] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. A deep unsupervised language model for protein design. *bioRxiv*, 2022.

[18] Zhangyang Gao, Cheng Tan, Stan Li, et al. Alphadesign: A graph protein design method and benchmark on alphafolddb. *arXiv preprint arXiv:2202.01079*, 2022.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. ISBN: 2006.11239 Publication Title: arXiv [cs.LG], June 2020.

[20] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. preprint, Systems Biology, April 2022.

[21] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative Models for Graph-Based Protein Design. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[22] Bowen Jing, Gabriele Corso, Regina Barzilay, and Tommi S. Jaakkola. Torsional diffusion for molecular conformer generation. In *ICLR2022 Machine Learning for Drug Discovery*, 2022.

[23] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Brian Kuhlman and David Baker. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, 97(19):10383–10388, 2000.

[26] Andrew Leaver-Fay, Matthew J. O'Meara, Mike Tyka, Ron Jacak, Yifan Song, Elizabeth H. Kellogg, James Thompson, Ian W. Davis, Roland A. Pache, Sergey Lyskov, Jeffrey J. Gray, Tanja Kortemme, Jane S. Richardson, James J. Havranek, Jack Snoeyink, David Baker, and Brian Kuhlman. Scientific benchmarks for guiding macromolecular energy function improvement. *Methods in Enzymology*, 523:109–143, 2013.

[27] Tony E Lewis, Ian Sillitoe, Natalie Dawson, Su Datt Lam, Tristan Clarke, David Lee, Christine Orengo, and Jonathan Lees. Gene3d: extensive prediction of globular domains in proteins. *Nucleic acids research*, 46(D1):D435–D439, 2017.

[28] Ke Liu, Xiangyan Sun, Jun Ma, Zhenyu Zhou, Qilin Dong, Shengwen Peng, Junqiu Wu, Suocheng Tan, Günter Blobel, and Jie Fan. Prediction of amino acid side chain conformation using a deep neural network. *arXiv preprint arXiv:1707.08381*, 2017.

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[31] 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.

[32] Matthew McPartlon and Jinbo Xu. Attnpacker: An end-to-end deep learning method for rotamer-free protein side-chain packing. *bioRxiv*, 2022.

[33] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99, 1963.

[34] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using rosetta. In *Methods in enzymology*, volume 383, pages 66–93. Elsevier, 2004.

[35] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[36] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ISBN: 1503.03585 Publication Title: arXiv [cs.LG], March 2015.

[37] 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*, 11(4):402–411, 2020.

[38] Alexey Strokach and Philip M Kim. Deep generative modeling for protein design. *Current opinion in structural biology*, 72:226–236, 2022.

[39] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

A Model details

A.1 Training

Models are optimized with the AdamW optimizer with and a cosine learning rate decay schedule [24, 30, 29]. All models are trained on single K80 and V100 GPUs on Google Cloud. We use gradient accumulation to increase the effective batch sizes.

A.2 Constraint embedding

Beta stands are considered adjacent if their minimum C_α pairwise distance is less than 5 Å during training. When considered adjacent, the orientation (parallel or anti-parallel) of beta-beta pairing is also given as an input. The other pairs of secondary structure elements (beta-helix, helix-helix) are considered adjacent if their minimum C_α pairwise distance is less than 7 Å. No adjacency information is encoded for loop blocks.

The constraint embedding network consists of a 5-layer GPT-3-like transformer followed by 25 layers of triangle attention blocks [23].

A.3 Diffusion decoders

Structure diffusion The diffusion decoder model conditions on input constraints and intermediate noised protein structures and predict corrected protein structures. The model consists of layers of Invariant Point Attention (IPA) blocks [23], which encode the current structure and features, predict backbone update parameters, and then update the structure rotations and translations.

The structure diffusion network has 12 layers, 4 IPA heads, 4 query points per residue, 4 value points per residue, and is trained with subsampled backbones up to to 256 residues. The network is trained with $T = 1000$, batch size of 160, and learning rate 10^{-3}.

Sequence diffusion For sequence diffusion, no constraint embeddings are used, and no backbone updates are done in the forward pass.

The network has 15 layers, 4 IPA heads, 4 query points per residue, 4 value points per residue, and is trained with subsampled backbones up to to 128 residues. The network is trained with $T = 100$, batch size 160, and learning rate 10^{-3}.

Rotamer diffusion For rotamer diffusion, the model sees as an input the entire full-atom protein structure with diffused side-chains. No constraint embeddings are used, and no backbone updates are done in the forward pass. The network has 6 layers, 4 IPA heads, 4 query points per residue, 4 value points per residue, and is trained with subsampled backbones up to to 75 residues. The network is trained with $T = 500$, batch size 100, and learning rate 5×10^{-4}.

Structure inpainting For structure inpainting, we keep the entire structure fixed except for a region that is masked. For each datapoint during training, we execute "block diffusion" with probability 0.6 and "contiguous diffusion" with probability 0.4. In block diffusion each loop is masked with probability 0.25, and the other blocks are masked with probability 0.05. In contiguous diffusion, we choose contiguous blocks at random to diffuse towards the prior with probability 0.03 for each starting residue and with length distributed uniformly between 1 and 15.

The structure inpainting model is fine-tuned from the structure diffusion model with $T = 1000$, batch size 160, and learning rate 1×10^{-3}.

Joint structure and sequence inpainting For joint structure and sequence diffusion, the pretrained structure inpainting network is kept frozen, and the pretrained sequence diffusion network is finetuned on the predicted structures outputted by the structure diffusion network. The sequence model is trained with $T_{\text{sequence}} = 100$, batch size 160, and learning rate 5×10^{-4}.

B Experiment details

B.1 TIM-barrel topology sampling

We specify TIM-barrel block adjacencies corresponding to a parallel beta-barrel surrounded by a ring of helices that are each adjacent to their immediate neighbors. We assume each quarter of the TIM-barrel to have the following secondary structure blocks in order: L-E-L-H-L-E-L-H-L (where L – loop, E – beta strand, H – helix). We sample random lengths for these secondary structure blocks with reasonable selected lower and upper bounds, and repeat the sampled topologies $4\times$ for the four-fold symmetric topology. We eliminate topologies longer than 256 residues. We then run the diffusion model on the sampled secondary structure string and block adjacencies to decode a range of TIM-barrel structures with varied underlying topologies.

B.2 Ig domain joint loop backbone and sequence sampling

We start with a random Ig domain from the train set with CATH ID 5tdoA01. We mask out the CDR loops, as well as an additional loop that might interact with the CDRs but is not natively hypervariable. We then sample random loop lengths for these regions. Note that the adjacency information does not change for the structure, as we do not encode any adjacency information for loop regions. Moreover, note that although CDR loops can have some structured regions, we ask the model to produce loops only. We run the joint structure and sequence model to generate new variable-length loop backbones as well as a sequence for the new loop regions. We then use the learned rotamer packer to pack the sequence of the entire structure, including the new loop side-chains.

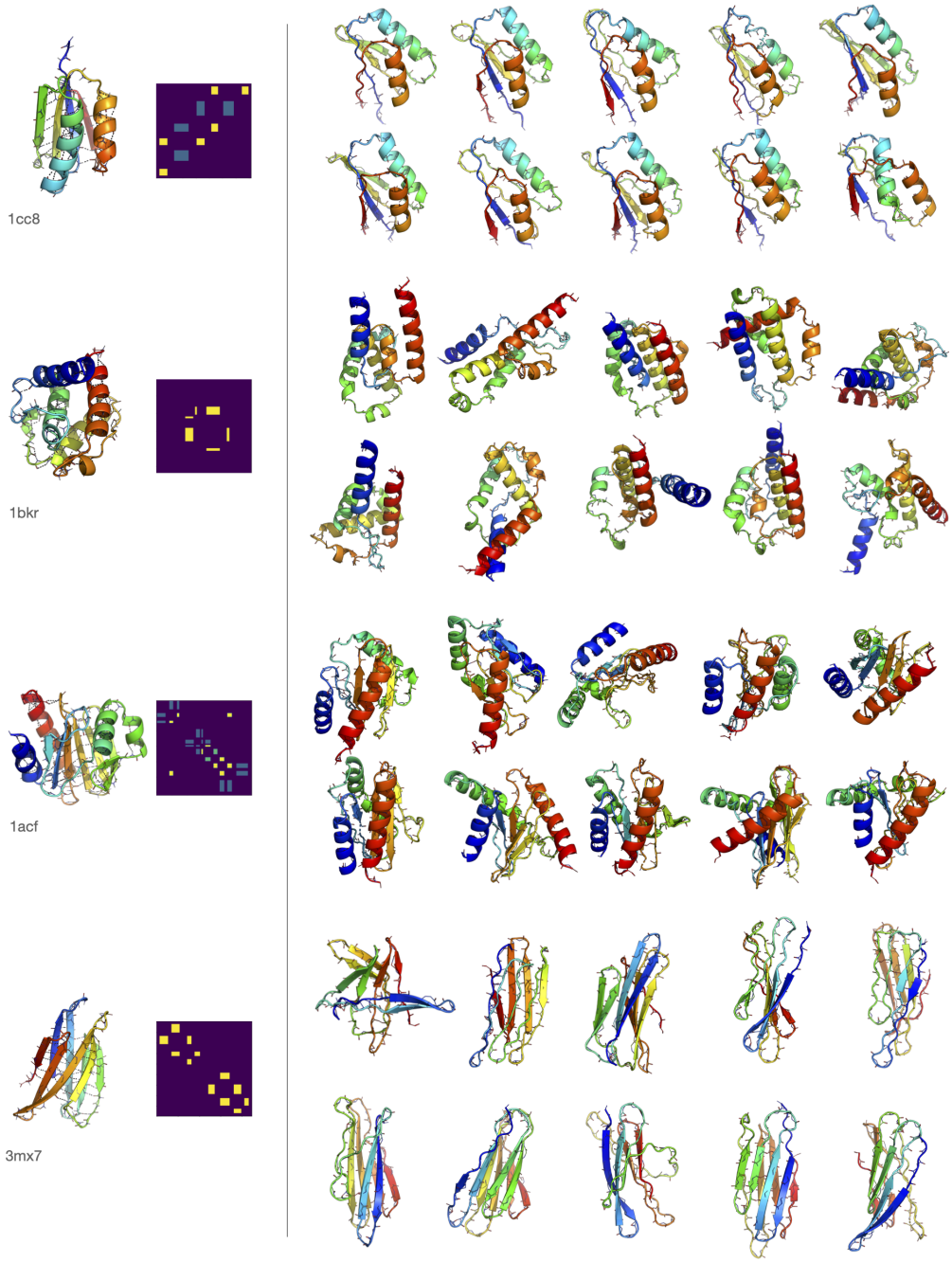

Figure 7: Random generated backbone samples.

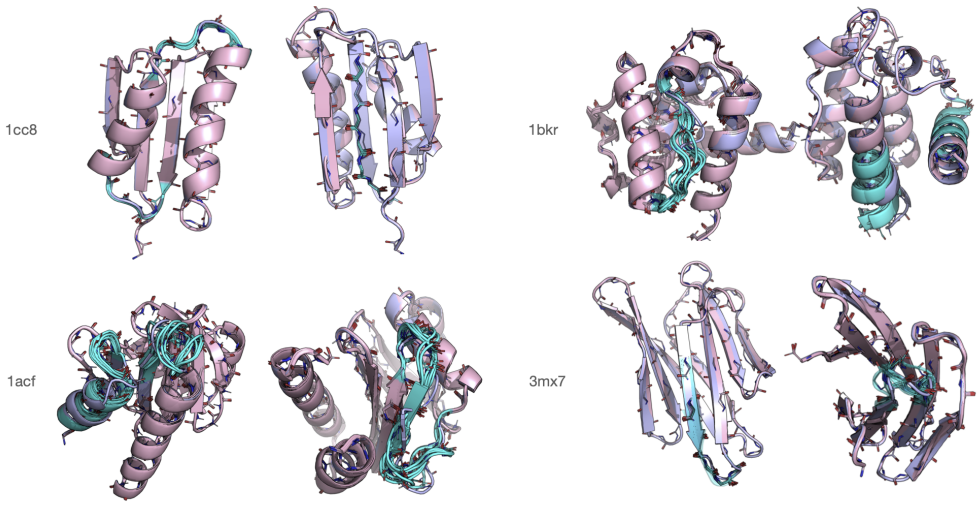

Figure 8: Structure modification: Random generated inpainting samples.

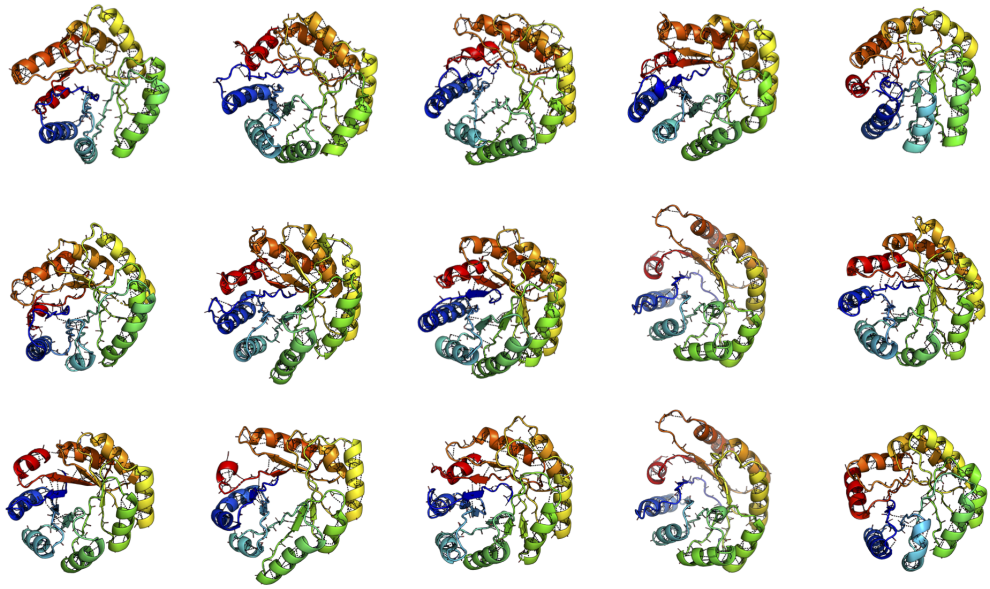

Figure 9: Random generated TIM-barrels with varied underlying four-fold symmetric topologies. Lengths of secondary structure elements in each quarter are sampled randomly.

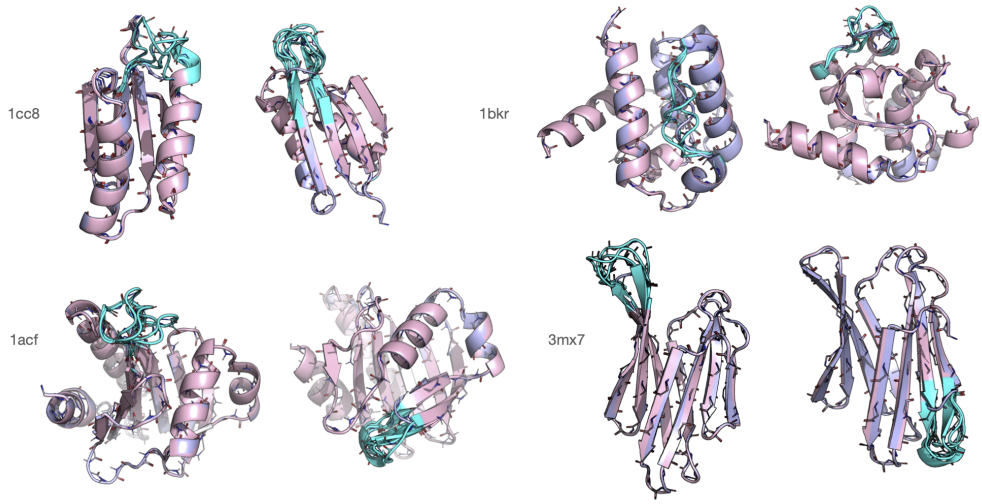

Figure 10: Structure modification: Sampling loops of varying lengths.