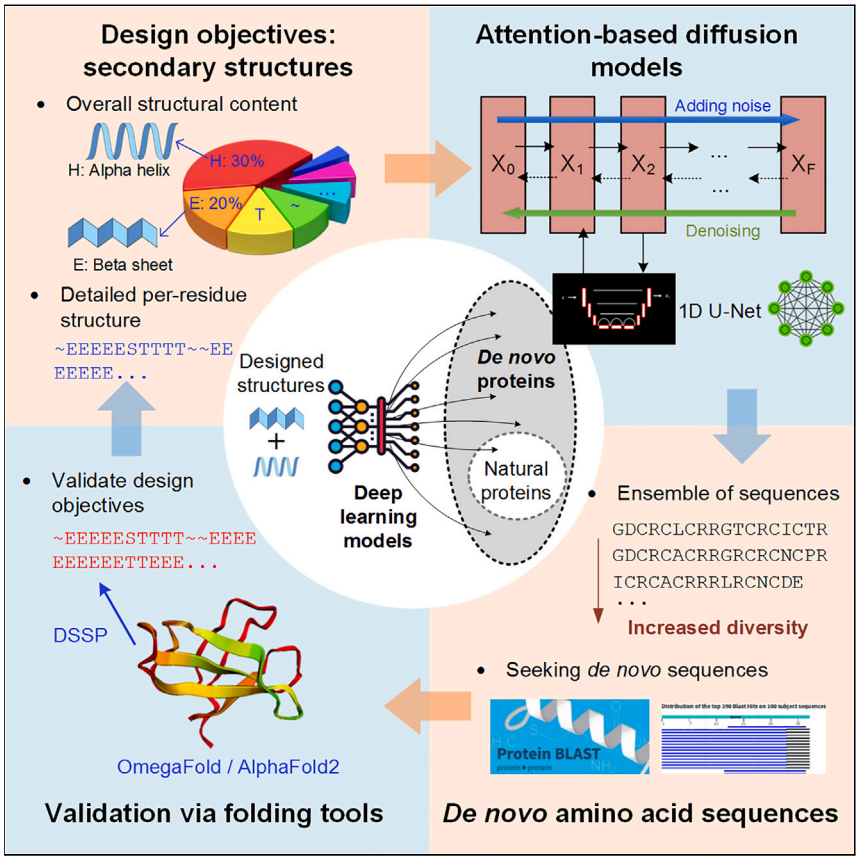
Chem

CellPress

Article

Generative design of *de novo* proteins based on secondary-structure constraints using an attention-based diffusion model



Designing proteins beyond the naturally existing ones is of great potential for numerous scientific and engineering applications; however, to date, it remains prohibitively expensive. Here, we leverage attention-based diffusion models to efficiently generate novel protein sequences with prescribed secondary structures. Our models handle variegated design objectives robustly and predict *de novo* protein sequences that have not yet been discovered in nature, opening up many avenues for discovering superior protein materials and systems, including those beyond structural objectives.

Bo Ni, David L. Kaplan, Markus J. Buehler

mbuehler@mit.edu

Highlights

Diffusion models can efficiently generate proteins with desired secondary structures

De novo protein sequences, not yet discovered in nature, can be generated

The models remain robust regarding imperfect or even unrealistic design goals

The models can be extended to generate *de novo* proteins for other properties

Ni et al., Chem 9, 1828–1849
July 13, 2023
https://doi.org/10.1016/j.chempr.2023.03.020

CellPress

Chem

Article

Generative design of *de novo* proteins based on secondary-structure constraints using an attention-based diffusion model

Bo Ni,[1] David L. Kaplan,[2] and Markus J. Buehler[1,3,4,*]

SUMMARY

We report two generative deep-learning models that predict amino acid sequences and 3D protein structures on the basis of secondary-structure design objectives via either the overall content or the per-residue structure. Both models are robust regarding imperfect inputs and offer *de novo* design capacity because they can discover new protein sequences not yet discovered from natural mechanisms or systems. The residue-level secondary-structure design model generally yields higher accuracy and more diverse sequences. These findings suggest unexplored opportunities for protein designs and functional outcomes within the vast amino acid sequences beyond known proteins. Our models, based on an attention-based diffusion model and trained on a dataset extracted from experimentally known 3D protein structures, offer numerous downstream applications in the conditional generative design of various biological or engineering systems. Future work could include additional conditioning and an exploration of other functional properties of the generated proteins for various properties beyond structural objectives.

INTRODUCTION

Proteins are critical biological building blocks that constitute fundamental functions of all life, as well as significant biomaterials emerging from natural evolution, including silks, collagens, complex assemblies (such as cells), and tissue assemblies (such as skin).[1–4] These various structures and associated outstanding properties depend on the underlying sequences of amino acids (AAs) and the subsequent folded three-dimensional (3D) structures. For millennia, it has been a popular and fruitful approach to take inspiration from nature about how to design materials,[2,5] including recent lessons learned from proteins (e.g., designing synthetic materials inspired by nacre[6,7] and silks[8,9]). Furthermore, considering that there exist 20^{100} possible AA sequences for a 100-residue protein and that natural evolution has only sampled a small fraction of these, there remains a broad unexplored design space and hence significant potential for discovering *de novo* proteins with potentially unprecedented properties and functions.[10] However, it is also due to this enormous design space and costs associated with experimental testing that great challenges remain in finding appropriate tools to design *de novo* protein sequences that yield a set of targeted structural features or properties.[8,9,11–13]

In the present work, we are particularly focused on the mechanical properties of proteins, for which the secondary-structure content is key (previous studies have demonstrated that elementary units of secondary structures and their interactions

THE BIGGER PICTURE

The design of *de novo* protein sequences has great potential in achieving superior combinations of novel functions and mechanical properties beyond known, natural proteins. However, the tremendous number of possible sequences and the cost of experimental testing make the effective search and validation of superior *de novo* protein candidates extremely challenging. Here, we leverage a diffusion-model-based deep-learning framework to efficiently generate novel protein sequences that meet a desired overall secondary-structure fractional content or per-residue type of secondary structure. The generated sequences show novelty beyond existing, natural ones. By robustly generating various novel sequences with the desired structural features, our model provides rapid strategies for a target-guided *de novo* protein design that leads to novel discoveries of superior protein materials for various biological and engineering applications and can be extended for other design objectives in future work.

within one chain can still play key roles in determining the mechanical properties of protein materials). For instance, α-helical proteins[14] tend to yield stretchy materials, β-sheet-rich ones yield rather rigid materials,[15–17] and combinations (as in silks) provide simultaneous rigidity, strength, and toughness via a nanocomposite design strategy.[15,17,18] Furthermore, using protein molecular building blocks as a means of constructing higher-level hierarchical materials offers high degrees of design and flexibility to generate targeted mechanical properties, such as flaw tolerance,[19–21] or tunable properties, such as stiffness or toughness,[22–24] among many others. For example, for tissue repair and regenerative medicine, one can tune the durability of protein-based implants by designing the content and types of the secondary structures.[25–31] For silk proteins, combining sequences with different ratios of β sheet, β turn, and random coil content can vary the mechanical properties of the resulting materials in an expanded range.[26,32,33] Therefore, the significant role played by the secondary-structure content in influencing the mechanical properties of protein materials and complex tissue systems suggests that this feature can be a valuable target or condition for the *de novo* protein design of mechanical properties.

The application of machine learning (ML) approaches to protein studies in recent years has provided effective avenues for predicting structure, properties, and functions on the basis of protein sequences. Let's take the folding problem of predicting 3D structures for the given sequences as an example: the deep learning (DL)-based tool AlphaFold 2 represented a breakthrough in achieving competitive accuracy while bypassing expensive and time-consuming measurements in conventional experimental methods.[34] The latest AlphaFold model has enabled the prediction of the structures of ~200 million proteins in the human proteome, as well as that of several other organisms, going far beyond the pace of experimental methods.[35] Furthermore, end-to-end models based on DL that predicts the secondary structures and properties for given sequences have also been developed.[36,37] For instance, from their primary sequences, the secondary-structure types and contents can be predicted with good accuracy by various ML models, including feed-forward neural networks,[38] recurrent neural networks,[39] deep convolutional networks,[40] and transformer-based language models.[41,42] The development of these DL models greatly reduces the cost of screening large numbers of protein sequences.

In contrast, the inverse design of *de novo* proteins that satisfy targeted features presents unique challenges and remains largely open for exploration, even with DL models. On the one hand, stochastic search algorithms constructed with handcrafted optimization functions and sampling approaches are often adopted for such inverse designs.[43,44] For example, searching for protein sequences that yield the desired ratio of the secondary-structure content requires a combination of genetic algorithms and DL-based predictors.[45] However, even with the efficient DL-based predictor, the iterative process of searching can potentially still be time consuming, whereas the convergence of the iterations and the quality and varieties of the discovered sequences are not necessarily correlated.

On the other hand, compared with applications of image generation, generative models have not yet been broadly generalized for protein structures, and their potential in solving such problems remains largely unexplored. Various DL methods have been used to generate images and image-like field data, including autoencoders,[46,47] generative adversarial nets (GANs),[48–50] and transformer-based diffusion models.[51,52] For example, diffusion-model architectures,[51] such as DALL-E2,[53] Imagen,[54] and latent or stable diffusion,[55] have recently produced state-of-the-art

[1]Laboratory for Atomistic and Molecular Mechanics (LAMM), Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA

[2]Department of Biomedical Engineering, Tufts University, Medford, MA 02155, USA

[3]Center for Computational Science and Engineering, Schwarzman College of Computing, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA

[4]Lead contact

*Correspondence: mbuehler@mit.edu

https://doi.org/10.1016/j.chempr.2023.03.020

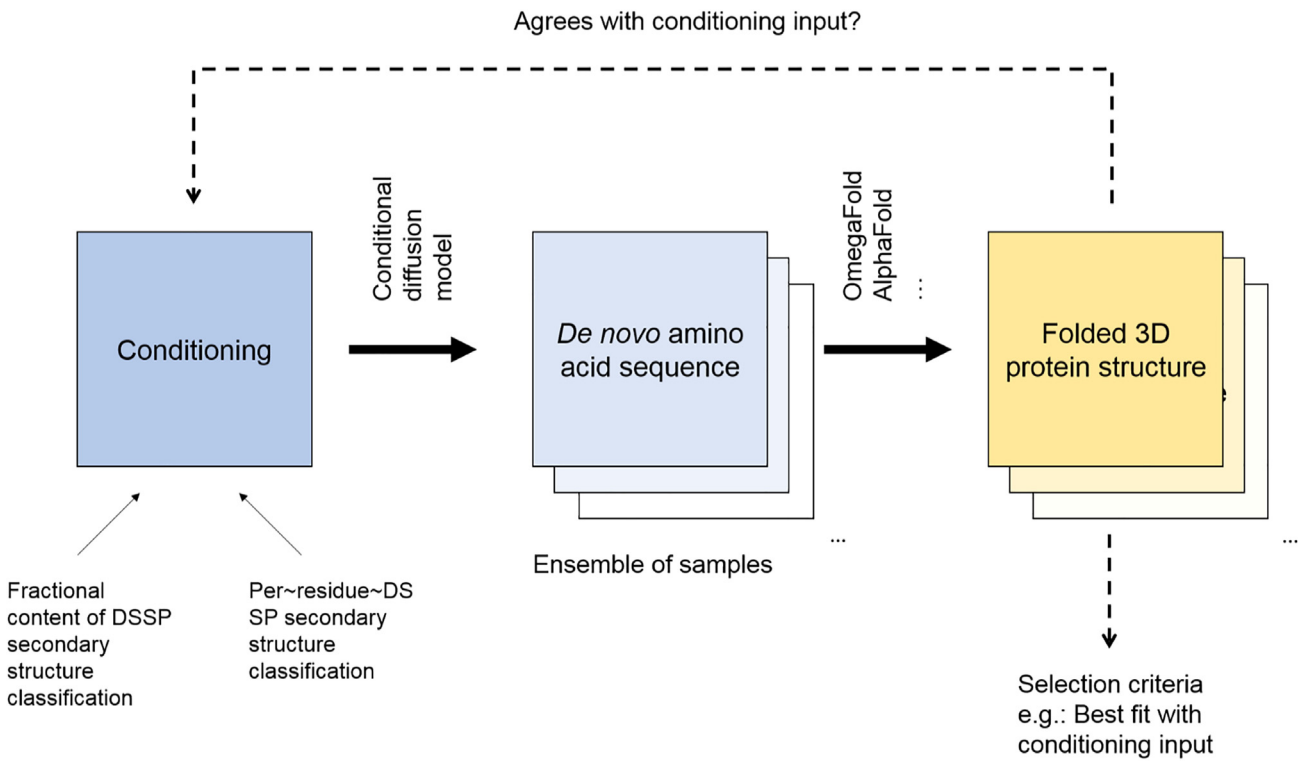


Figure 1. Overview of the model developed here

The model takes a conditioning description as input and produces, via a conditional attention-based diffusion model (Figure 2), an AA sequence. We use OmegaFold[64] and AlphaFold[34,65] to then predict the 3D structures of the resulting proteins. Two models are trained; model A takes fractional input of the secondary structure and then predicts sequences, and model B takes per-residue secondary-structure information as input and predicts sequences. The generative model can be used to produce a number of samples that can be analyzed for further down selection (e.g., we can select samples that meet the target conditioning input the best or are the least similar to the known natural proteins).

performance in text-to-image generation tasks for an unprecedented degree of photorealism and language understanding. For engineering applications, conditional GAN models have been demonstrated to be capable of generating stress/strain fields from modulus fields[56,57] and vice versa,[58,59] and progressive transformer diffusion models can learn and predict various behaviors during dynamic fracture processes.[60] In comparison, protein sequences and structures show different formats and features from those of images, graphs, or image-like fields. Attempts to bridge such generative models for protein studies remain few but grow rapidly. For example, variational autoencoders have been adjusted for the generation of diverse new protein structures without designed conditions,[61] equivariant denoising diffusion probabilistic models have been developed for generating proteins according to a given topology and constraints,[62] and another diffusion model has been applied to the construction of scaffold structures that support a desired motif in proteins.[63] With these recent successes, it is promising to explore how to leverage these generative models in image domains and adjust them to handle *de novo* protein designs effectively and efficiently.

In this paper, we propose an attention-based diffusion model for proteins and report a generative DL model that predicts AA sequences and 3D protein structures on the basis of secondary-structure design objectives (Figure 1). By proposing a U-Net architecture that handles one-dimensional (1D) protein data, we construct a model that takes a conditioning description of the desired secondary structures as input

and produces various AA sequences from random noise vector sources (akin to the schematic shown in Figure 2A). We achieve this through an attention-based diffusion model that we trained by minimizing the L2 difference between the actual and predicted noise levels (Equation 4), realizing a stepwise denoising strategy, by using a U-Net architecture with intersecting convolutional and attention layers, as depicted in Figures 2B and 2C.

We then integrate the model with folding prediction methods to determine the 3D structures of the resulting sequences, classify their secondary structures, and compare these outputs with the input conditions. Finally, we check the designed sequences with known proteins to analyze their novelty. After training the model on a set of Protein Data Bank (PDB) proteins, we demonstrate that the models are able to generate various *de novo* protein sequences of stable structures that closely follow the given secondary-structure conditions, thus bypassing the iterative search process in previous optimization methods.[43–45] Given our model's capability and the known significance of secondary structures for the mechanical properties of proteins, we expect that our model can be useful in numerous applications in conditional generative designs of various scientific and engineering protein systems.

RESULTS AND DISCUSSION

Figure 1 depicts an overview of the model developed here: generating novel protein structures on the basis of conditioning parameters. In this process, the model takes a conditioning description as input and produces, via a conditional attention-based diffusion model (Figure 2), an AA sequence that is then used for constructing a 3D protein model. We use OmegaFold[64] and AlphaFold[34,65] to predict the 3D structures of the resulting proteins. Two models are developed, trained, and applied. First, model A takes fractional input of the secondary structure and then predicts sequences. Second, model B takes per-residue secondary-structure information as input and predicts the sequences. Both models construct the predicted sequences from random signals, under conditioning, by reversing the diffusion process in a step-by-step fashion, as summarized in Figures 2A and 2B. The deep neural network is tasked with identifying the added noise at each step so that it can be removed successfully. Details of the models, training procedure, and related topics are included in the experimental procedures, and Table 1 shows a summary of the eight secondary-structure parameters used here according to the Define Secondary Structure of Proteins (DSSP) convention.[66] Additional aspects, including an analysis of the distribution of the data in the training set, are included in Figures S1–S3.

Figure 3 shows the results for protein generation based on the fractional secondary-structure content (model A). Figures 3A–3F show a variety of representative cases, including high β sheet content (Figure 3A), a mix of α-helical and β sheet content (Figures 3B and 3F), pure α-helical content (Figure 3C), α-helical content with significant disorder (Figure 3D), and a completely disordered protein (Figure 3E). The left column shows the conditioning vector (eight components reflecting the eight different types of secondary-structure content shown in Table 1) and the resulting AA sequences. The columns in the middle depict the resulting protein structures predicted by OmegaFold in two renderings. The right column shows a comparison of secondary-structure fractions between the input conditions and those reconstructed from the generated sequences.

We now focus on one of the predictions, the β sheet structure in Figure 3A, and explore the assembly of this protein into higher-order arrangements. These

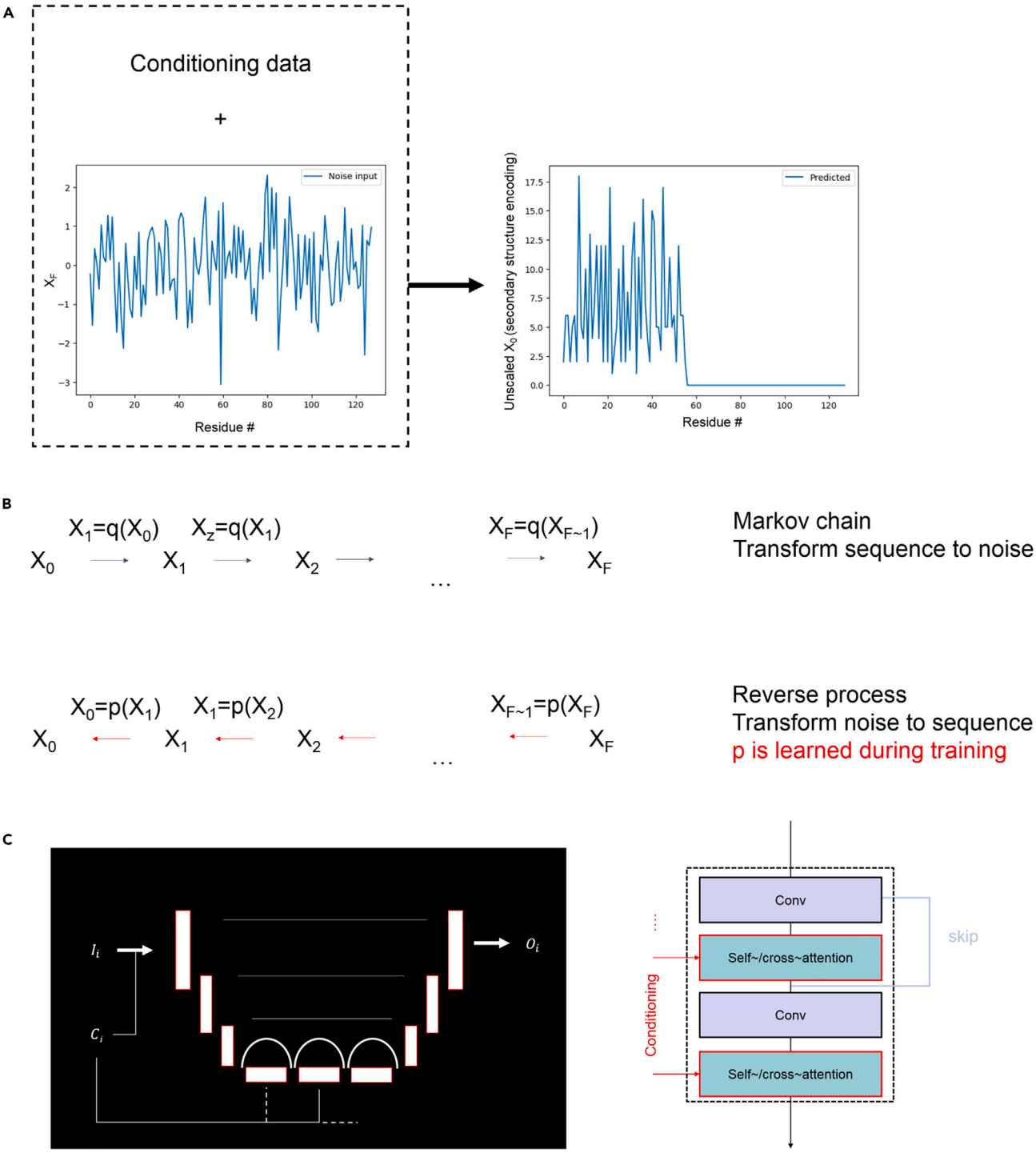


Figure 2. Overview of the attention-based diffusion model

(A) Visualization of how noise and conditioning data are transformed into a solution. During training, pairs of conditioning data and a resulting output are used.

(B) Illustration of the Markov chain of noising (top) and denoising (bottom).

(C) Depiction of the 1D U-Net architecture that translates input I_i into output O_i under condition set C_i. The model features 1D convolutional layers and self-/cross-attention layers, as shown on the right.

Table 1. Secondary-structure code according to the DSSP convention

DSSP code[66]	Description
H	α helix
E	extended parallel and/or anti-parallel β sheet conformation
T	hydrogen-bonded turns (3, 4, or 5 turns)
~	unstructured
B	β bridge (single-pair β sheet hydrogen-bond formation)
G	$3/3_{10}$ helix
I	π helix
S	bend

assemblies are not predicted by our model; rather, we explore whether the predicted β strands would assemble into higher-order filamentous structures. Figure 4 shows the analysis of the results from Figure 3A in greater detail. Figures 4A and 4B show a comparison of the predictions between OmegaFold and AlphaFold. The results are comparable and indicate that both folding methods yield similar results. Figure 4C shows the structure prediction, by AlphaFold-Multimer,[67] of an assembly of three of these β sheet building blocks. Figure 4D shows, similarly, the assembly geometry of eight β sheets. It is notable that the model arranges three subparts consisting of two mini β barrels and one bivalent assembly in the middle, whereas for the individual chains, the designed secondary structures are preserved.

Next, we analyze the predicted AA sequences to assess whether, and to what extent, they represent novel sequences or closely related forms of the existing and/or known proteins. This is done via Basic Local Alignment Search Tool (BLAST) analysis.[68] Table 2 shows the results of the BLAST analysis for the various cases for the results from model A. We find that, generally, the model predicts structures that are similar to the existing protein sequences, as can be seen from the BLAST results. However, some generated sequences (as for the second and third cases) do not exist in the PDB-based training set. Although there is some novelty in the sequences and there are measures one could take to drive further variation, we focus on ways to ensure a greater diversity of sequence predictions (strategies to enhance this could be to increase the conditioning probability dropout or to add noise to the conditioning vector during training).

This is accomplished via model B, where we use residue-level conditioning. In this scenario, each residue is conditioned on the basis of one of the eight DSSP secondary-structure codes. Figure 5 depicts results for protein generation based on residue-level secondary-structure content implemented in model B. We consider five cases with different distributions of the secondary structure, including a predominant β sheet (Figure 5A), a long α helix with a breaker in the center (Figure 5B), a small α helix (Figure 5C), a sandwich α helix/β sheet structure (a β sheet centered between two α-helical domains) (Figure 5D), and a partially disordered-helical protein (Figure 5E). The folded results (left column in Figure 5) reveal generally good agreement with the design objectives specified in the right (blue font) and confirm that the model enables us to design specific geometric details and localizations of secondary structures. Even though these proteins are *de novo* sequences (see BLAST analysis in Table 3), OmegaFold (and AlphaFold) reaches relatively high pLDDT scores (denoting a per-residue estimate of prediction confidence in a range from 0 to 100), and it is typically the largest for the α-helical structures.

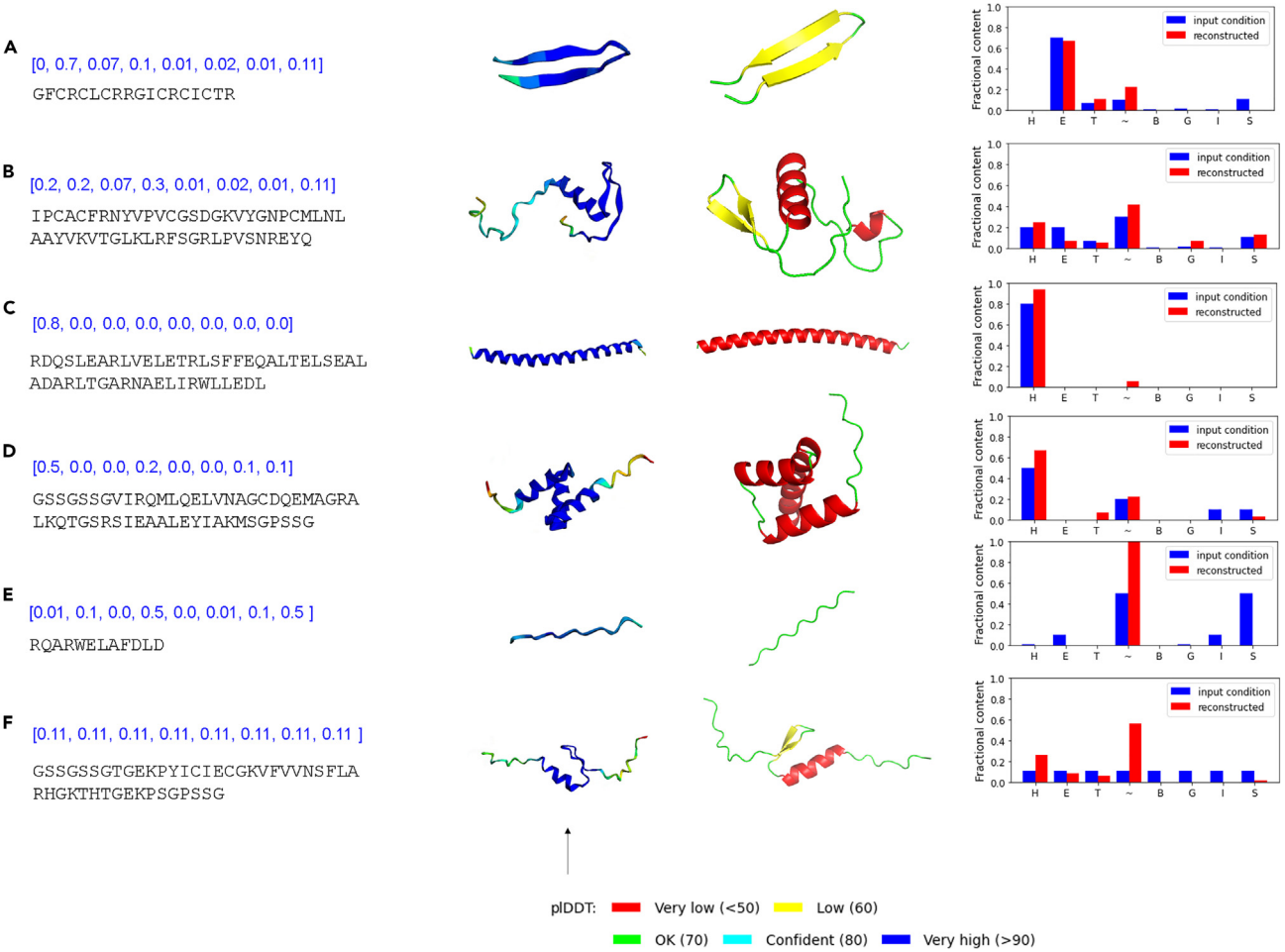


Figure 3. Results for protein generation based on the fractional secondary-structure content (model A)

A variety of representative cases include high β sheet content (A), a mix of α-helical and β sheet content (B and F), pure α-helical content (C), α-helical content with a significant disorder (D), and a completely disordered protein (E). The fractional patterns used here are similar to those that have been observed in known proteins and relate to mechanical properties. The left column shows the conditioning vector (eight components reflecting the eight different types of secondary-structure content shown in Table 1) and the resulting AA sequences. The two columns in the middle depict the resulting protein structures predicted by OmegaFold in two renderings (middle left, pLDDT score; middle right, secondary-structure analysis via PyMol; yellow, β sheet; red, helix; and green, disordered). The right column shows the comparison of secondary-structure fractions between the input conditions and those reconstructed from the generated sequences. Additional designs are depicted in Figure S7.

On the basis of these predictions, we now conduct a more detailed analysis of one of the designs. Figure 6 reveals a detailed analysis of the two designs (Figures 5A and 5E), including a comparison between OmegaFold and AlphaFold predictions. The indication of the residue number (C and N termini) and the localization of the specific secondary structure shows excellent agreement with the design objective. The top row in each case depicts the structure color coded by the residue number (rainbow plot), and the bottom row shows the secondary-structure color coding.

Table 3 summarizes the results of the BLAST analysis for the various cases of model B. The analysis shows that the model generates protein structures that reflect the design objectives well (see analysis in Figure 5) and that are also *de novo* sequences that have little similarity with the existing AA sequences. We find that the BLAST results indicate similarities of around 50%–60% for most cases, but one case reaches 85.71% for 66% of the query cover. We further note that most of the other cases

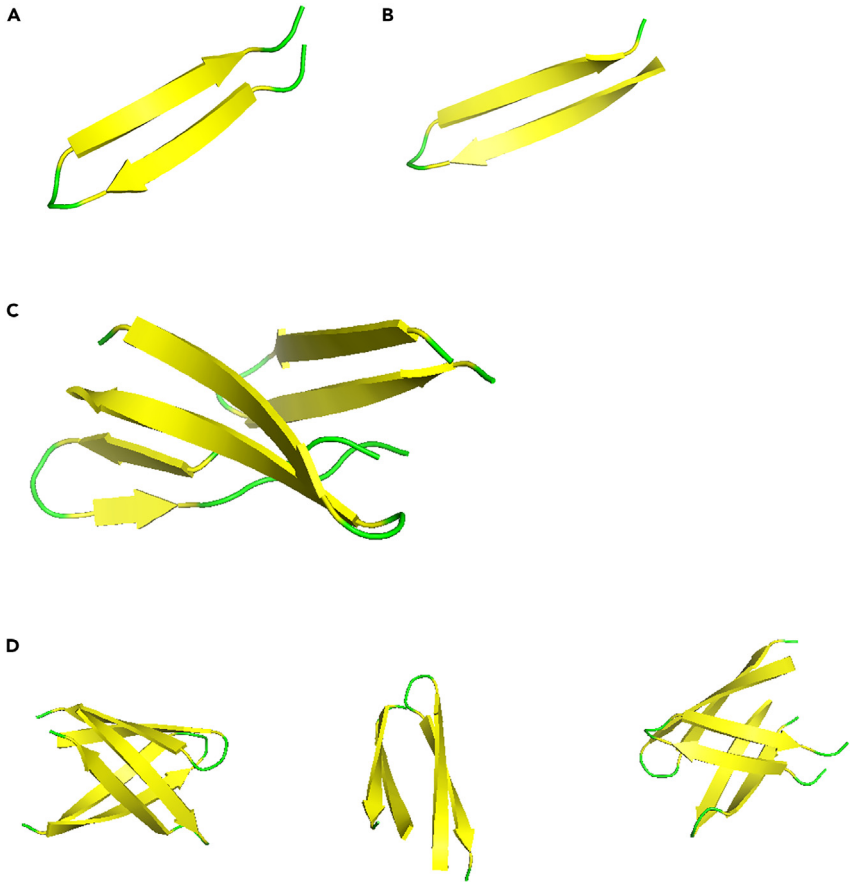


Figure 4. Assembly of the generated proteins into higher-order arrangements

Analysis of the results from Figure 3A in greater detail and exploration of how β strands—which are known to assemble into higher-scale structural assemblies such as amyloid filaments or other fibrous structures—yield such emerging structures.

(A and B) Comparison of the predictions between OmegaFold (A) and AlphaFold (B). The results are comparable.

(C) AlphaFold structure prediction of an assembly of three of these β sheet building blocks.

(D) The assembly geometry of eight β sheets. It is notable that the model arranges in three subparts consisting of two mini β barrels and one bivalent assembly in the middle. The designed protein fragments are preserved well when multiple chains aggregate, which indicates, for some cases, that the current design models could be applied even for multimer cases, especially for β-sheet-rich protein assemblies.

We use the AlphaFold-Multimer model for the folding prediction of protein complexes in (C) and (D).

have much smaller similarities and/or small query covers. This result is noteworthy because it provides strong evidence that model B is capable of discovering new protein designs. The difference in this explorative potential is most likely due to the way in which novel sequences are generated (with condition dropouts). Because the conditioning sequences are short, there is limited variability as the model explores new designs that conform less strongly to the conditioning task.

Several DL-based protein design systems have emerged recently.[69–71] Complementing these alternative approaches, our model serves a unique design perspective focused on the secondary-structure content (model A) or the residue-specific secondary-structure content (model B). We briefly discuss the differences with

Table 2. Results of the BLAST analysis for the various cases for the results from model A

		BLAST result: The sequence producing the most significant alignment	
Conditioning	Sequence	Among PDB proteins	Beyond PDB proteins
[0, 0.7, 0.07, 0.1, 0.01, 0.02, 0.01, 0.11]	GFCRCLCRRGICRCICTR	100% query cover; 94.44% identical with PDB: 1HVZ	N/A
[0.2, 0.2, 0.07, 0.3, 0.01, 0.02, 0.01, 0.11]	IPCACFRNYVPVCGSDG KVYGNPCMLNLAAYVK VTGLKLRFSGRLPVSNREYQ	–	94% query cover; 76.00% identical with GenBank: AHW57452.1
[0.8, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	RDQSLEARLVELETRLSF FEQALTELSEALADARL TGARNAELIRWLLEDL	–	100% query cover; 94.12% with NCBI: WP_011036690.1
[0.5, 0.0, 0.0, 0.2, 0.0, 0.0, 0.1, 0.1]	GSSGSSGVIRQMLQELV NAGCDQEMAGRALKQ TGSRSIEAALEYIAKMSGPSSG	100% query cover; 96.30% identical with PDB: 2COS	N/A
[0.01, 0.1, 0.0, 0.5, 0.0, 0.01, 0.1, 0.5]	RQARWELAFDLD	91% query cover, 90.91% identical with PDB: 1ID6	N/A
[0.11, 0.11, 0.11, 0.11, 0.11, 0.11, 0.11, 0.11]	GSSGSSGTGEKPYICIEC GKVFVVNSFLARHGKTH TGEKPSGPSSG	80% query cover; 80.43% identical with PDB: 2ENE	N/A

Generally, the model predicts structures that are similar to existing protein sequences, as can be seen from the BLAST results.[68] However, some generated sequences (as for the second and third cases) do not exist in the PDB-based training set.

regard to other generative models. For example, to generate the desired sequences, proteinMPNN[69] requires atomic details of the protein backbone design as input, including distances between Cα-Cα atoms, relative Cα-Cα-Cα frame orientations and rotations, and backbone dihedral angles. In contrast, our model only conditions the secondary-structure types of the generated sequences in the global (model A) or the residual (model B) level, leaving the detailed atomic coordinate information unspecified. Indeed, for one pick of secondary-structure contents, in terms of either the global fractions or the secondary-structure type sequence, there could exist many different backbone atomic structures. So, these two are protein design tools working on different levels with different respective advantages.

Another model is Chroma,[70] which adopts a graph neural-network representation of proteins and is based on a diffusion process between the protein backbone and the noisy structure of a collapsed polymer system instead of Gaussian noise. In the RFdiffusion model,[71] the folding tool (RoseTTAFold[72]) is fined-tuned as the denoising network. Both models pay particular attention to the backbone structures of the designed protein. Thus, various conditions regarding the atomic structure of the backbone can be implemented for both monomer and protein complex designs, making them general-purpose structure-focused protein design tools. In contrast, our model specifically focuses on the mapping between secondary structures and primary sequences, bypassing the construction of the atomic details of the backbone. Our model implicitly learns the rules of which sequences can be generated. Correspondingly, users can directly work on the residual level (model B) and the monomer level (model A) and skip the detailed choice of conditions on the atomic level. They can also combine model A as a way to get a first estimated protein sequence and then refine the design by using model B by specifying the residue-level secondary-structure detail.

We note that, in testing and applying our models, there is a lack of explicit rules regarding which secondary-structure distributions are physically possible and which are not given the finite AAs as the building blocks. Therefore, it remains unclear and challenging to directly construct an exhausted set of physically possible input conditions to test our models. Instead, in the current work, we focus on testing our

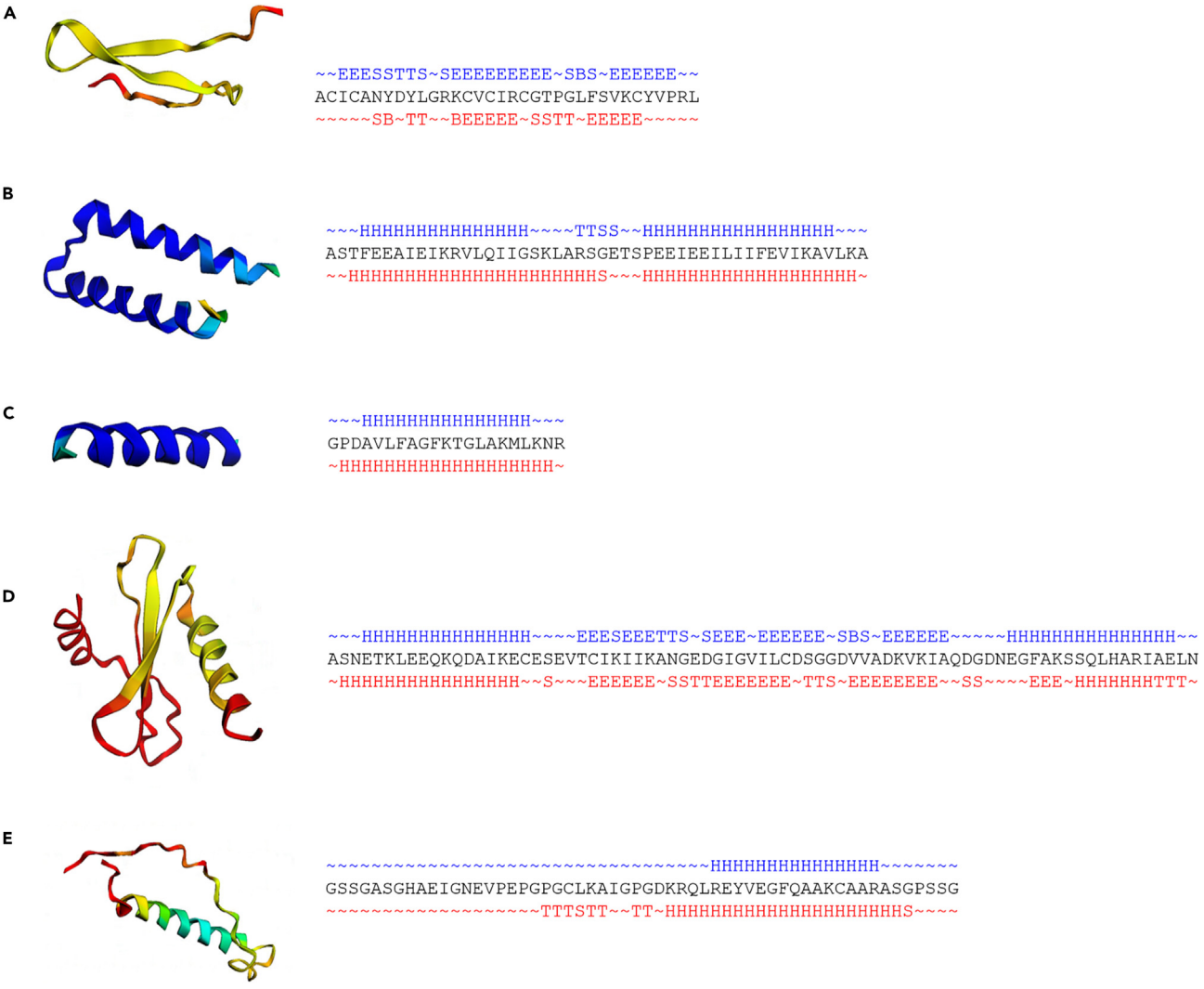


Figure 5. Results for protein generation based on residue-level secondary-structure content (model B)

We consider five cases with different distributions of secondary structures, including a predominant β sheet (A), a long α helix with a breaker in the center (B), a small α helix (C), a sandwich α helix/β sheet structure (a β sheet centered between two α-helical domains) (D), and a partially disordered helical protein (E). The right column shows the input secondary-structure sequences (in blue), generated AA sequences (in black), and secondary-structure contents reconstructed from the generated proteins (in red). The left column depicts the resulting protein structures in the rendering of the pLDDT score. The secondary structures of the folded results show good agreement with the design objectives, and the per-residue accuracy (defined in Equation 6) of those cases is 44% (A), 79% (B), 81% (C), 60% (D), and 73% (E), confirming that the model enables us to design specific geometric details and localizations of secondary structures. Even though these proteins are *de novo* sequences (see analysis in Table 3), OmegaFold (and AlphaFold) reaches relatively high pLDDT scores (typically the largest for α-helical structures). Several additional designs are depicted in Figures S4–S6.

models with secondary-structure patterns that have been observed in known proteins and relate to mechanical properties. In Figures 3 and 5, we manually constructed the input conditions so that they resemble some typical secondary-structure patterns in known PDB proteins, especially those that are known to affect the deformation process and mechanical properties, such as β sheets and α helices. Here, we intentionally included noises or even errors in the input conditions to test and demonstrate the behaviors and robustness of our model. For example, for model A, the sum of the secondary-structure fractions in some of the tested conditions in Figure 3 slightly deviates from 1. However, even with those imperfect inputs, the model is still able to generate sequences that respect the relative

Table 3. Results of the BLAST analysis for the various cases for model B

Case	Sequence	BLAST result: The sequence producing the most significant alignment	
		Among PDB proteins	Beyond PDB proteins
A	ACICANYDYLGRKCVCIRCGTPGLFSVKCYVPRL	–	82% query cover; 43.24% identical with GenBank: MBR4744867.1
B	ASTFEEAIEIKRVLQIIGSKLARSGETSPEEIEEILIIFEVIKAVLKA	–	60% query cover; 57.58% identical with GenBank: MBR3311276.1
C	GPDAVLFAGFKTGLAKMLKNR	–	66% query cover; 85.71% identical with GenBank: MCI9553838.1
D	ASNETKLEEQKQDAIKECESEVTCIKIIKANGEDGIGVILCDSG GDVVADKVKIAQDGDNEGFAKSSQLHARIAELN	–	45% query cover; 44.68% identical with NCBI: WP_257531070.1
E	ASNETKLEEQKQDAIKECESEVTCIKIIKANGEDGIGVILCDSG GDVVADKVKIAQDGDNEGFAKSSQLHARIAELN	–	41% query cover; 61.54% identity with GenBank: TBR12876.1

In the interest of space, the secondary-structure design objective is not shown; rather, the case IDs (A–E) correspond to the cases shown in Figure 5. The analysis shows that the model generates protein structures that reflect the design objectives well (see analysis in Figure 5) and that are also *de novo* sequences that have little similarity with existing AA sequences. The BLAST results indicate similarities around 50%–60% for most cases, but one case reaches 85.71% for 66% of the query cover. Most other cases have much smaller similarities and small query covers.

concentrations of different secondary-structure types, as shown in the right column of Figure 3. Moreover, the generated proteins have a correct fractional content that does add up to 1. To expand on this, Figure S8 shows an analysis of the rigor of input conditioning in model A. We examine here the difference in predictions that do not necessarily add up to 1 ($\neg\square\sum_{i=1\ldots8}n_i = 1$; Figure S8A) and conditioning that adds up to 1 ($\sum_{i=1\ldots8}n_i = 1$; Figure S8B). Similar predictions are obtained in most cases, except for the second to last from the bottom. This result shows that the model has a certain degree of robustness and can deal with unphysical input (e.g., fractional secondary-structure content that does not add up to 1) and still produce reliable results.

Although the rules of protein construction are complex, especially for secondary structures that yield long-range interactions (such as β sheets, where close or distant parts of the protein combine to form the 3D structure), primarily local-focused structures, such as α helices, do not have such constraints. Hence, we conducted a case study to explore whether the model can solve systematic variations of the input. Applying model B, we generate a series of α-helical sequences of increasing length. As shown in Figure S4, as we systematically control the length of the helical residue in the inputs, the model predicts a length-dependent conformation transition from an unconstructed chain to a straight α-helical segment to multiple α-helical segments with kinks to a slightly curved continuous α-helical segment. This transition indicates that the trained model not only seeks to respect the input conditions but also yields to the underlying constraints of physically possible secondary structures learned implicitly during training.

In another test, Figure S5 depicts designed sequences based on secondary-structure sequences identified from recently deposited PDB proteins taken from the CASP15 (a; PDB: 7ROA) and CASP14 (b; PDB: 7JTL) target sets. The generated proteins, shown on the right, are *de novo* sequences with no significant overlap with any existing known sequences. This shows that the model can generate new protein sequences—different from those generated via evolutionary mechanisms—and offer alternative or candidate sequence options for structural templates. In addition to this result, Figure S6 shows results for designed sequences based on several design targets with different complexities (hard to easier from top to bottom). The left panels show the error value over the iterations (we repeat generation until a maximum number of iterations is reached or until the design error is below a critical value). The critical error is set to 0.1 in all cases, and only the α-helical structures

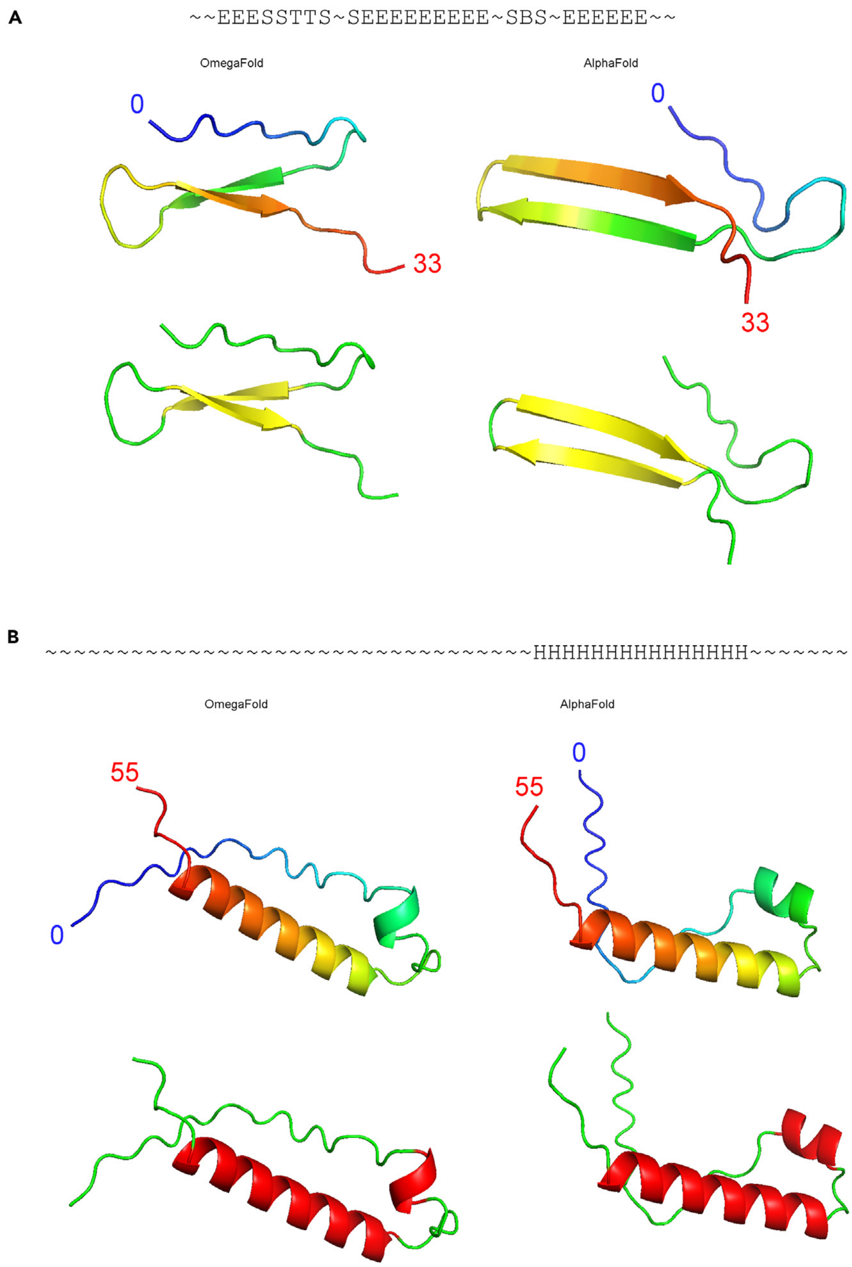


Figure 6. Structural and organizational analysis of the proteins generated by model B

Detailed analysis of two designs (Figures 5A and 5E), including a comparison between OmegaFold and AlphaFold predictions (the design objective is noted at the top of each subplot). The indication of the residue number (C and N termini, respectively) and the localization of specific secondary structure shows excellent agreement with the design objective. The top row in each case depicts the structure color coded by the residue number (rainbow plot), and the bottom row shows color coding for secondary structures.

reach that goal. Generally, error values fluctuate around a mean value, reflecting the general level of challenge to solve the design problem. Furthermore, because we trained our model by using a classifier-free strategy, we can explore how removing conditioning information affects the results. In Figure S7, we show the results of these experiments, which demonstrate that increasing the parameter ξ from 1 to

30 offers additional tuning of the predictions (for $\xi = 1$, the model is fully conditioned and becomes less conditioned as the parameter increases).

Conclusions

We report a method to generate novel protein designs on the basis of an attention-based diffusion model. Two variants of the model are presented: one that conditions predictions on the basis of the overall fractional content of secondary structures (model A) and a second that conditions predictions on the basis of per-residue-level secondary-structure conditioning (model B). The results show that model B is more effective at generating *de novo* sequences that are not found in nature or that have not been discovered yet (see analysis in Table 3). The model is capable of predicting a variety of new sequences; although they reveal some similarities with the existing sequences (on the order of 50%–60% similarity), they introduce significant amounts of new design cues. This level of variation can be regarded as an interesting measure to add to the diversity of natural protein designs.

Future work could include further refining sequences to meet additional criteria, for instance, biological activity. Other critical steps include experimental validation of the designs, especially for cases that have little similarities with known proteins. In addition, it could be interesting to add more explorative capacity to model A to achieve greater sequence diversity much more distinct from existing proteins at levels similar to what we achieved in model B. One way to achieve this could be to use model B to generate a greater variety of protein sequences, fold the proteins, and then use these new data points to expand the existing dataset. Millions of new protein structures have already been predicted by available computational methods, and these could also be added to the training dataset. So far, we excluded these predicted proteins because we wanted to focus the training procedure on the experimental data. One could also use integrated optimization algorithms where sequence predictions are accepted, rejected, or altered according to the performance criteria, such as invoking the structure prediction tools or methods to directly assess the secondary-structure content in an end-to-end fashion. These strategies could offer interesting research avenues and the possibility of achieving multiple objective functions.

This work can be taken in other future research avenues, including inpainting strategies where part of the sequence is provided as a boundary condition and the model is asked to fill in a missing part. If conditioning strategies are used as proposed in this paper, such models would probably be able to solve protein design tasks of the kind where a domain is sought to be altered, for instance, for mechanical or other (e.g., biological and other) property optimization. A useful application of this strategy can be, for instance, the design of silk materials where key domains could be strengthened by the addition of greater β-sheet-forming domains, whereas others can be rendered more stretchy through the use of helical domains. The sandwich design shown in Figure 5D is an example that indicates the feasibility of such design strategies. An important next step is the experimental synthesis of such protein structures and the use of mechanical analysis tools, such as optical tweezers, to assess the outcomes. The tool reported here could also be combined with other generative protein models,[69–71] including those focused on predicting sequences to meet a certain geometric or shape demand.

Our model can generate *de novo* proteins according to desired secondary structures. As another future research topic, the novelty of these new sequences could lead to superior mechanical properties and related functions that go beyond what

has been observed in natural proteins. For example, studies of natural proteins have demonstrated that the mechanical stiffness, strength, and toughness of silk show size effect and length dependence on the β strands in silk proteins.[73] Combining this mechanistic knowledge with our generative model, one can systematically generate new sequences that yield the secondary structures of rationally optimized designs and verify the performances via simulations and experiments.

Although we have explored the potential to study how β strand monomers assemble into larger assemblies (Figure 4), the models and scope of this work are strictly limited to single-chain proteins. It has been demonstrated that the individual pieces of secondary structures and their interactions can play important roles in determining the mechanical proteins of protein and protein materials. For example, the unfolding of individual α helices in vimentin in intermediate filament contributes mainly to its great extensibility,[19] and the interaction between parallel pieces of β strands in β-sheet-rich proteins governs their rupture strength.[74] Our models have generated sequences with similar patterns (e.g., Figures 3A and 3C or Figures 5A and 5C) effectively under different conditioning formats.

We also identified ways to increase the *de novo* nature of predicted sequences, especially by using model A, which tends to synthesize less diverse sequences than model B. With strategies such as classifier-free guidance and early stopping during training, we can generate significantly more diverse sequences (another option, not yet explored here, can be to vary the number of sampling steps, which can be powerful in generating a greater variety of generative results). A visual summary of the sequence alignments, as examples for some of the designs, is shown in Figures S9 and S10. Figure S11 shows sample alignments to explain the nature of the variations of the generated sequences to provide visual depictions of the results and the novel nature of the sequences. As evidenced from these analyses, the generative algorithm realized in model A has the capacity to discover sequence designs from a deep reservoir of patterns, some of which have also been discovered via natural processes.

As another future direction, it could be interesting to generalize our models toward the design of multimers under similar secondary-structure conditions. One straightforward strategy to do so is to introduce trivial linkers that represent the break between individual chains and further train the model with available data of protein complexes given that a similar scheme has been used to generalize AlphaFold2 for multimer tasks.[75,76] A more systematic way to generalize our current models for multimer designs would be to include more conditions, including those specifying the cross-chain geometry in protein complexes, such as binding residues and residue ties for symmetric or repeat protein designs.[69] Next, we could envision combining the current models and their possible generalizations for multimer tasks to generate large numbers of sequences and screen for those that undergo conformational changes in terms of secondary structure when forming multimer complexes. Because those would require significant additional research (including the development of a proper dataset), we leave them to future studies.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Professor Markus Buehler (mbuehler@mit.edu).

Materials availability

This study did not generate new unique reagents.

Data availability

Data and codes, as well as trained weights, either are available on GitHub at https://github.com/lamm-mit/ProteinDiffusionGenerator or will be provided by the lead contact upon reasonable request.

Dataset

We use the same dataset as in the earlier work by Yu et al.[45] Figure S1 shows a statistical analysis of the dataset used to train model B. Figure S1A summarizes the sequence-length distributions for lengths up to 128 AA. Figure S1B depicts the secondary-structure coding statistics. Figure S1C shows the AA residue statistics.

The right panel in Figure S1 shows the tokenizer dictionary for both secondary-structure and AA codes. The associations listed in these are used to translate string characters (for both AA codes and secondary-structure codes) into a numerical integer value. We then normalize the data to lay between 0 and 1, resulting in float numbers for input and output.

For model A, the input vector is the fractional content of the eight different types of DSSP secondary structures:

$$C = \frac{1}{N}[n_H, n_E, n_T, n_\sim, n_B, n_G, n_I, n_S] \quad \text{(Equation 1)}$$

where n_i is the number of residues with type i secondary structure and N is the total number of AAs in a protein.

Unused tokens in the predicted sequence (both models) and in the conditioning sequence (model B) are identified with token 0 for padding.

Figure S2 shows the distribution of the fractional contents of the eight types of secondary structures in the training set used for model A. Note that for many types of secondary structures, the distribution is not uniform and quite skewed with respect to the fraction between 0 and 1. These distributions are most likely a reflection of chemical, biological, and/or evolutionary principles that resulted in the protein designs. There exist relatively few sequences with a higher fraction of β bridge (>0.4), β sheet (>0.9), $3/3_{10}$ helix (>0.8), hydrogen-bonded turn (>0.8), π helix (>0.5), or bend (>0.7). Figure S3 depicts plots of the fractional content distributions of different pairs of secondary structures in the training set. These pair correlation plots are useful for understanding naturally occurring combinations of secondary-structure contents.

Design of the neural-network architectures

All codes were developed in PyTorch,[77] except that tokenizer was developed on the basis of, and trained with, TensorFlow Keras.[78]

The model is built on a 1D U-Net architecture composed of convolutional and transformer layers with skip connections (Figure 2C). U-Nets are a type of neural network that features the same input dimension as the output dimension, commonly used in problems such as image segmentation. The U-Net implemented here features a more complex architecture,[54,79] including the use of ResNet blocks, attention modules, and skip connections (see Table 4 for details). As depicted in Figure 2C, the U-Net features

Table 4. Parameters used in the progressive transformer diffusion model

Parameter	Value
U-Net	
U-Net dimension	128/512/768 (ratio-based model, model A); 256 (residue-level model, model B)
Dimension multipliers	1, 2, 4, 8
ResNet blocks	1
Attention heads	8
Feed-forward multiplier	2.0
Overall architecture	
Sequence length	64/128
Conditional dropout probability	0.2/0.1
Sample steps	96/64
σ_{min}	0.002
σ_{max}	80, 160
σ_{data}	0.5
ρ	7
P_{mean}	−1.2
P_{std}	1.2
P_{churn}	80
$S_{t,min}$	0.05
$S_{t,max}$	50
S_{noise}	1.003
Optimizer and parameters	Adam; learning rate = 1E−4; ε = 1e−8; β = (0.9, 0.99)
Additional parameter	
Batch size	256
Additional parameters: Model A	
Condition embedding dimension	512
Positional encoding depth	128
Signal embedding depth	368

self- and cross-attention blocks. These are used to contextualize the denoising process with both the conditioning data and the diffusion time step. The use of transformer-type architectures provides a meaningful way to also learn long-range relationships in the construction of AA sequences and how they interact to satisfy various physical and chemical constraints. During training, these are all learned.

The U-Net is used to translate the secondary-structure fraction vector (model A) or input sequence (model B) into the final field output via a learned denoising process. Figure 2B visualizes the denoising process, where the top defines Markov chain operator q, which adds Gaussian noise step by step (according to a defined noise schedule that defines how much noise ε_i is added at each step i), translating the original sequence, X_0 (left), into pure noise, X_F (right), as follows:

$$X_{i+1} = q(X_i) \quad \text{(Equation 2)}$$

The deep neural network is then trained to *reverse* this process, identifying operator p, which maximizes the likelihood of the training data, thereby offering a means to translate noise to solutions and thereby realizing the transition illustrated in Figure 2A (noise to solution), in a step-by-step fashion as indicated in the lower row of Figure 2B:

$$X_{i-1} = p(X_i) \quad \text{(Equation 3)}$$

Specifically, the U-Net is tasked with predicting the added noise, and correspondingly, the L2 distance of the actual added noise, ε_i, and the predicted added noise, ε_i', is adopted as the loss function for the training process.

$$\mathcal{L} = \|\varepsilon_i - \varepsilon_i'\|_{L_2} \quad \text{(Equation 4)}$$

That way, the trained model can predict the added noise. Knowing this quantity then allows us to realize a numerical solution to the problem stated in Equation3, used to generate the next iteration of the *denoised* sequence:

$$X_{i-1} = X_i - \varepsilon_i' \quad \text{(Equation 5)}$$

In Equation 2, the sequence X_i at step i is transformed by the removal of noise ε_i'. This process is performed iteratively, whereas the neural network predicts, given the current state X_i, the noise to be removed at a given time step in the denoising process (see Figure 2B).

We use an improved noise schedule, sampling, and training processes proposed by Karras et al.[80] because it provides us with enhanced and computationally efficient predictions, and we specifically obtain results within just 64 (model A) or 96 (model B) denoising steps. Table 4 provides details about the model architecture parameters. The implementation is based on the code published at Github[79] but extended to feature a new U-Net architecture to feature 1D sequence data with higher-order embeddings.

The conditional encoder scales the sequence data values to be between 0 and 1 and feeds each in the embedding dimension (and unscaled for reverse tokenization and analysis).

In model A, the conditioning is fed via embeddings that are used for cross-attention with the input after being expanded into higher-order embedding dimensions through linear layers. Trainable positional encoding is used (ordinal ordering of each conditioning parameter, from 1 to 8 as defined in Equation 1, is designated and then encoded via a fully connected embedding layer). In model B, the conditioning is provided as sequence conditioning, where the conditioning sequence of secondary-structure encodings is concatenated to the input, i.e., the noise vector. We find the latter strategy to work better when the conditioning signal has similar dimensionality as the output (and this is the case in model B since the conditioning and prediction are of the same dimension). In both cases, this yields a conditional algorithm where

$$X_{i-1} = \mathrm{p}(X_i, C_i) \quad \text{(Equation 6)}$$

so that the model can produce samples that meet certain target features defined by C_i.

Model A tends to generate sequences with comparatively more limited novelty and diversity. We explored ways to address this limitation. One possible way is to use early stopping to avoid overfitting, as had also been proposed by Nichol and Dhariwal.[81] The results shown in Figure S8 are obtained, for instance, by a model with a relatively large 768-dimensional U-Net and early stopping. This strategy, combined with classifier-free guidance[82] parameter variations as presented in Figure S7, can be a means of increasing the creative capacity of this model. Classifier-free guidance is achieved by training the model with conditional dropout; in our case, 20% of the time the conditioning information is removed so that we can make predictions both with conditioning ($\mathrm{p}(X_i, C_i)$) and without conditioning ($\mathrm{p}(X_i)$). During sampling, the

predictions are then combined according to $p_{CFG}(X_i, C_i; \xi) = p(X_i) + (p(X_i, C_i) - p(X_i))\xi$, where ξ is a conditioning parameter that determines how the conditional and unconditional solutions are mixed ($\xi = 1$ yields the fully conditioned model, and for larger values, less conditioning is obtained). Other than in Figure S7, all results in this paper are obtained with $\xi = 1$.

The entire prediction pipeline involves all of the steps shown in Figure 1: (1) taking a conditioning parameter, (2) using the diffusion model to make conditional predictions, and (3) predicting the folded 3D protein structure by using OmegaFold.[64] For validation, we also implement folding predictions by using alternative methods, including AlphaFold 2 via ColabFold and some testing with trRosetta. For details regarding the primary folding strategies used, see section "protein folding."

As indicated in Figure 1, both models sample solutions and are hence capable of generating a set of possible solutions to the same design problem. As a systematic way to obtain the best-possible solution, we implement an iterative algorithm as outlined in Figure 1. We repeat generation until we reach a maximum number of iterations or until we are below a critical error value (see Figure S6 for an analysis of errors over iterations). Alternative approaches could target those designs that are the least similar to the known natural proteins or that show a compromise between novelty and meeting the design demand.

Training and validation

We use an Adam optimizer,[83] and Table 4 includes a variety of key models and hyperparameters. Figure 7 shows several validation examples, comparing ground truth and predictions, for two cases (A and B) for model B (results are similar for model A). These are protein sequences taken from the validation set (10% of the total data available), so they are not *de novo* sequences and do not merit further analysis. Both cases show that the model can accurately predict sequences according to specific secondary-structure content. For cases where multiple sequences correspond to the same or similar secondary-structure input, multiple, varied predictions are made.

To measure the conditioning capability of model B, we define per-residue accuracy, A_{pr}, as the fraction of residues with the designed secondary-structure types for the generated protein sequence, i.e.,

$$A_{pr} = \frac{n}{N,} \qquad \text{(Equation 7)}$$

where n is the number of residues with the same secondary-structure types as the input condition and N is the total number of AAs in the protein. A_{pr} is between 0 and 1, and the error is defined as $E_{pr} = 1 - A_{pr}$.

Protein folding

We implement OmegaFold[64] directly in our model architecture for rapid prediction of protein structures from the sequence. OmegaFold offers a rapid alternative because it does not require multiple sequence alignment (MSA) yet produces results with accuracy similar to that of AlphaFold 2 and trRosetta (and similar, related state-of-the-art methods).

To check the results, we further use AlphaFold 2[34] for monomers and AlphaFold-Multimer[67] for multimers via ColabFold[65] to conduct these experiments by using the ColabFold implementation. We use the pdb70 template set, and MMseqs2 (UniRef

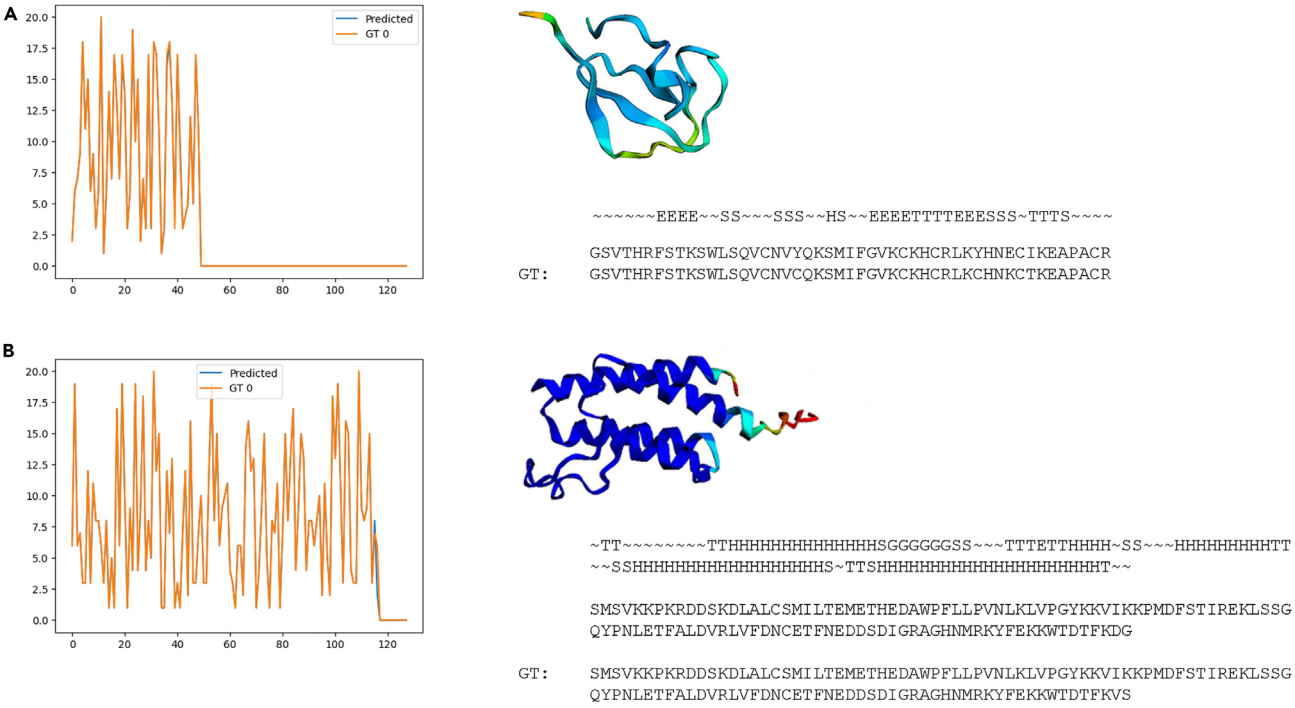


Figure 7. Tests of the predictions from model B using cases from the validation set

Validation examples, comparing ground truth and predictions, for two cases (A and B). These are protein sequences taken from the validation set (10% of the total data available), so they are not *de novo* sequences. Both cases show that the model can accurately predict sequences according to specific secondary-structure content. For cases where multiple sequences correspond to the same or similar secondary-structure input, multiple, varied predictions are made.

and Environmental), by using six cycles. The highest prediction is used for the analysis in this paper. Additional comparisons are conducted with trRosetta[84] for validation.

The inclusion of folding tools and the comparison of several such prediction strategies here are aimed at validating whether the generated sequences are likely to fulfill the designed secondary-structure conditioning parameters. Although the ultimate validation of our model might require producing and examining those protein sequences experimentally, the applications of the state-of-the-art *in silico* folding tools, such as AlphaFold2, provide a useful benchmarking pathway of higher efficiency and lower costs, which has been shown to be a viable strategy in recent works. As shown in Figures 3 and 5, the majority parts of the folded structures achieve a relatively high pLDDT score (~70), which indicates that the predicted structures are expected to be modeled well and remain stable. Furthermore, to fold *de novo* sequences with potentially limited MSA information, we use OmegaFold, which is designed to make predictions directly from the primary sequence accurately and efficiently without MSAs. We find good agreement between high-confidence folded structures predicted by these two methods for the sequences we generated, including the *de novo* designs. This gives us confidence that the novel sequences generated by our model are likely to deliver the desired secondary structures.

BLAST analysis

The BLAST[68] analysis for the various cases is conducted with the BLASTp (protein-protein BLAST) algorithm and the NCBI non-redundant protein sequence database (nr). Summary results are shown in Tables 2 and 3, and detailed results are provided in Figures S9–S11.

Visualization

We use PyMol[85] and Py3DMol[86] for visualization of the protein structures.

Software versions and hardware

We use Python 3.8.12, PyTorch 1.10[77] with CUDA (CUDA version 11.6), and a NVIDIA RTX A6000 with 48 GB VRAM for training and inference.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.chempr.2023.03.020.

ACKNOWLEDGMENTS

We acknowledge support from the MIT-IBM Watson AI lab, USDA (2021-69012-35978), DOE-SERDP (WP22-S1-3475), ARO (79058LSCSB, W911NF-22-2-0213, and W911NF2120130), NIH (U01EB014976 and R01AR077793), and ONR (N00014-19-1-2375 and N00014-20-1-2189).

AUTHOR CONTRIBUTIONS

M.J.B. conceived the study, developed and trained the neural network, and performed the associated data analysis, including of the protein models. B.N. analyzed proteins, sequences, and errors in collaboration with M.J.B. B.N. and D.L.K. supported the analysis and wrote the paper with M.J.B.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: December 1, 2022
Revised: February 27, 2023
Accepted: March 20, 2023
Published: April 20, 2023

REFERENCES

1. López Barreiro, D., Yeo, J., Tarakanova, A., Martin-Martinez, F.J., Buehler, M.J., López, D., Yeo, J., Tarakanova, A., Martin-Martinez, F.J., and Buehler, M.J. (2019). Multiscale modeling of silk and silk-based biomaterials—a review. Macromol. Biosci. *19*, e1800253. https://doi.org/10.1002/MABI.201800253.

2. Gronau, G., Krishnaji, S.T., Kinahan, M.E., Giesa, T., Wong, J.Y., Kaplan, D.L., and Buehler, M.J. (2012). A review of combined experimental and computational procedures for assessing biopolymer structure–process–property relationships. Biomaterials *33*, 8240–8255. https://doi.org/10.1016/J.BIOMATERIALS.2012.06.054.

3. Vepari, C., and Kaplan, D.L. (2007). Silk as a biomaterial. Prog. Polym. Sci. *32*, 991–1007. https://doi.org/10.1016/J.PROGPOLYMSCI.2007.05.013.

4. Ling, S., Kaplan, D.L., and Buehler, M.J. (2018). Nanofibrils in nature and materials engineering. Nat. Rev. Mater. *3*, 1–15. https://doi.org/10.1038/natrevmats.2018.16.

5. Wegst, U.G.K., Bai, H., Saiz, E., Tomsia, A.P., and Ritchie, R.O. (2014). Bioinspired structural materials. Nat. Mater. *14*, 23–36. https://doi.org/10.1038/nmat4089.

6. Gu, G.X., Takaffoli, M., and Buehler, M.J. (2017). Hierarchically enhanced impact resistance of bioinspired composites. Adv. Mater. *29*, 1700060. https://doi.org/10.1002/ADMA.201700060.

7. Barthelat, F., Yin, Z., and Buehler, M.J. (2016). Structure and mechanics of interfaces in biological materials. Nat. Rev. Mater. *1*, 1–16. https://doi.org/10.1038/natrevmats.2016.7.

8. Huang, W., Tarakanova, A., Dinjaski, N., Wang, Q., Xia, X., Chen, Y., Wong, J.Y., Buehler, M.J., and Kaplan, D.L. (2016). Design of multistimuli responsive hydrogels using integrated modeling and genetically engineered silk–elastin-like proteins. Adv. Funct. Mater. *26*, 4113–4123. https://doi.org/10.1002/ADFM.201600236.

9. Krishnaji, S.T., Bratzel, G., Kinahan, M.E., Kluge, J.A., Staii, C., Wong, J.Y., Buehler, M.J., and Kaplan, D.L. (2013). Sequence–structure–property relationships of recombinant spider silk proteins: integration of biopolymer design, processing, and modeling. Adv. Funct. Mater. *23*, 241–253. https://doi.org/10.1002/ADFM.201200510.

10. Huang, P.S., Boyken, S.E., and Baker, D. (2016). The coming of age of de novo protein design. Nature *537*, 320–327. https://doi.org/10.1038/nature19946.

11. Paladino, A., Marchetti, F., Rinaldi, S., and Colombo, G. (2017). Protein design: from computer models to artificial intelligence. WIREs Comput. Mol. Sci. *7*, e1318. https://doi.org/10.1002/WCMS.1318.

12. Wang, J., Cao, H., Zhang, J.Z.H., and Qi, Y. (2018). Computational protein design with deep learning neural networks. Sci. Rep. *8*, 6349. https://doi.org/10.1038/s41598-018-24760-x.

13. Qin, Z., Wu, L., Sun, H., Huo, S., Ma, T., Lim, E., Chen, P.Y., Marelli, B., and Buehler, M.J. (2020). Artificial intelligence method to design and fold alpha-helical structural proteins from the primary amino acid sequence. Extreme Mech. Lett. *36*, 100652. https://doi.org/10.1016/J.EML.2020.100652.

14. Ackbarow, T., Chen, X., Keten, S., and Buehler, M.J. (2007). Hierarchies, multiple energy

barriers, and robustness govern the fracture mechanics of α-helical and β-sheet protein domains. Proc. Natl. Acad. Sci. USA *104*, 16410–16415. https://doi.org/10.1073/pnas.0705759104.

15. Qin, Z., and Buehler, M.J. (2010). Cooperative deformation of hydrogen bonds in beta-strands and beta-sheet nanocrystals. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *82*, 061906. https://doi.org/10.1103/PhysRevE.82.061906.

16. Xu, Z., and Buehler, M.J. (2010). Mechanical energy transfer and dissipation in fibrous beta-sheet-rich proteins. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *81*, 061910. https://doi.org/10.1103/PhysRevE.81.061910.

17. Knowles, T.P.J., and Buehler, M.J. (2011). Nanomechanics of functional and pathological amyloid materials. Nat. Nanotechnol. *6*, 469–479. https://doi.org/10.1038/nnano.2011.102.

18. Hu, X., Kaplan, D., and Cebe, P. (2006). Determining beta-sheet crystallinity in fibrous proteins by thermal analysis and infrared spectroscopy. Macromolecules *39*, 6161–6170. https://doi.org/10.1021/ma0610109.

19. Qin, Z., Kreplak, L., and Buehler, M.J. (2009). Hierarchical structure controls nanomechanical properties of vimentin intermediate filaments. PLoS One *4*, e7294. https://doi.org/10.1371/JOURNAL.PONE.0007294.

20. Ackbarow, T., Sen, D., Thaulow, C., and Buehler, M.J. (2009). Alpha-helical protein networks are self-protective and flaw-tolerant. PLoS One *4*, e6015. https://doi.org/10.1371/JOURNAL.PONE.0006015.

21. Spivak, D.I., Giesa, T., Wood, E., and Buehler, M.J. (2011). Category theoretic analysis of hierarchical protein materials and social networks. PLoS One *6*, e23911. https://doi.org/10.1371/JOURNAL.PONE.0023911.

22. Studart, A.R. (2013). Biological and bioinspired composites with spatially tunable heterogeneous architectures. Adv. Funct. Mater. *23*, 4423–4436. https://doi.org/10.1002/ADFM.201300340.

23. Keten, S., Chou, C.C., van Duin, A.C.T., and Buehler, M.J. (2012). Tunable nanomechanics of protein disulfide bonds in redox microenvironments. J. Mech. Behav. Biomed. Mater. *5*, 32–40. https://doi.org/10.1016/J.JMBBM.2011.08.017.

24. Wray, L.S., Rnjak-Kovacina, J., Mandal, B.B., Schmidt, D.F., Gil, E.S., and Kaplan, D.L. (2012). A silk-based scaffold platform with tunable architecture for engineering critically-sized tissue constructs. Biomaterials *33*, 9214–9224. https://doi.org/10.1016/J.BIOMATERIALS.2012.09.017.

25. Dinjaski, N., Ebrahimi, D., Qin, Z., Giordano, J.E.M., Ling, S., Buehler, M.J., and Kaplan, D.L. (2018). Predicting rates of in vivo degradation of recombinant spider silk proteins. J. Tissue Eng. Regen. Med. *12*, e97–e105. https://doi.org/10.1002/TERM.2380.

26. Keten, S., and Buehler, M.J. (2010). Nanostructure and molecular mechanics of spider dragline silk protein assemblies. J. R. Soc. Interface *7*, 1709–1721. https://doi.org/10.1098/RSIF.2010.0149.

27. Xiao, S., Xiao, S., and Gräter, F. (2013). Dissecting the structural determinants for the difference in mechanical stability of silk and amyloid beta-sheet stacks. Phys. Chem. Chem. Phys. *15*, 8765–8771. https://doi.org/10.1039/C3CP00067B.

28. Keten, S., and Buehler, M.J. (2008). Geometric confinement governs the rupture strength of H-bond assemblies at a critical length scale. Nano Lett. *8*, 743–748. https://doi.org/10.1021/nl0731670.

29. Ackbarow, T., Keten, S., and Buehler, M.J. (2009). A multi-timescale strength model of alpha-helical protein domains. J. Phys. Condens. Matter *21*, 035111. https://doi.org/10.1088/0953-8984/21/3/035111.

30. Keten, S., Rodriguez Alvarado, J.F., Müftü, S., and Buehler, M.J. (2009). Nanomechanical characterization of the triple β-helix domain in the cell puncture needle of bacteriophage T4 virus. Cell. Mol. Bioeng. *2*, 66–74. https://doi.org/10.1007/s12195-009-0047-9.

31. Buehler, M.J., and Yung, Y.C. (2009). Deformation and failure of protein materials in physiologically extreme conditions and disease. Nat. Mater. *8*, 175–188. https://doi.org/10.1038/nmat2387.

32. Jaleel, Z., Zhou, S., Martín-Moldes, Z., Baugh, L.M., Yeh, J., Dinjaski, N., Brown, L.T., Garb, J.E., and Kaplan, D.L. (2020). Expanding canonical spider silk properties through a DNA combinatorial approach. Materials (Basel) *13*, 3596. https://doi.org/10.3390/MA13163596.

33. Hayashi, C.Y., Shipley, N.H., and Lewis, R.V. (1999). Hypotheses that correlate the sequence, structure, and mechanical properties of spider silk proteins. Int. J. Biol. Macromol. *24*, 271–275. https://doi.org/10.1016/S0141-8130(98)00089-0.

34. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature *596*, 583–589. https://doi.org/10.1038/s41586-021-03819-2.

35. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. (2022). AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res. *50*, D439–D444. https://doi.org/10.1093/NAR/GKAB1061.

36. Liu, F.Y.C., Ni, B., and Buehler, M.J. (2022). Presto: rapid protein mechanical strength prediction with an end-to-end deep learning model. Extreme Mech. Lett. *55*, 101803. https://doi.org/10.1016/J.EML.2022.101803.

37. Khare, E., Gonzalez-Obeso, C., Kaplan, D.L., and Buehler, M.J. (2022). CollagenTransformer: end-to-end transformer model to predict thermal stability of collagen triple helices using an NLP approach. ACS Biomater. Sci. Eng. *8*, 4301–4310. https://doi.org/10.1021/acsbiomaterials.2c00737.

38. Zhang, B., Li, J., and Lü, Q. (2018). Prediction of 8-state protein secondary structures by a novel deep learning architecture. BMC Bioinformatics *19*, 1–13. https://doi.org/10.1186/S12859-018-2280-5/TABLES/13.

39. Pollastri, G., and McLysaght, A. (2005). Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics *21*, 1719–1720. https://doi.org/10.1093/BIOINFORMATICS/BTI203.

40. Mirabello, C., and Pollastri, G. (2013). Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. Bioinformatics *29*, 2056–2058. https://doi.org/10.1093/BIOINFORMATICS/BTT344.

41. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. (2022). ProtTrans: toward understanding the language of life through self-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. *44*, 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381.

42. Høie, M.H., Kiehl, E.N., Petersen, B., Nielsen, M., Winther, O., Nielsen, H., Hallgren, J., and Marcatili, P. (2022). NetSurfP-3.0: accurate and fast prediction of protein structural features by protein language models and deep learning. Nucleic Acids Res. *50*, W510–W515. https://doi.org/10.1093/NAR/GKAC439.

43. Lew, A.J., and Buehler, M.J. (2021). A deep learning augmented genetic algorithm approach to polycrystalline 2D material fracture discovery and design. Appl. Phys. Rev. *8*, 041414. https://doi.org/10.1063/5.0057162.

44. Khare, E., Yu, C.H., Gonzalez Obeso, C., Milazzo, M., Kaplan, D.L., and Buehler, M.J. (2022). Discovering design principles of collagen molecular stability using a genetic algorithm, deep learning, and experimental validation. Proc. Natl. Acad. Sci. USA *119*. e2209524119. https://doi.org/10.1073/pnas.2209524119.

45. Yu, C.H., Chen, W., Chiang, Y.H., Guo, K., Martin Moldes, Z., Kaplan, D.L., and Buehler, M.J. (2022). End-to-end deep learning model to predict and design secondary structure content of structural proteins. ACS Biomater. Sci. Eng. *8*, 1156–1165. https://doi.org/10.1021/acsbiomaterials.1c01343.

46. Hinton, G.E., and Zemel, R.S. (1993). Autoencoders, minimum description length and Helmholtz free energy. Advances in Neural Information Processing Systems 6 (NIPS 1993), *6*.

47. Dong, G., Liao, G., Liu, H., and Kuang, G. (2018). A review of the autoencoder and its variants: a comparative perspective from target recognition in synthetic-aperture radar images. IEEE Geosci. Remote Sens. Mag. *6*, 44–68. https://doi.org/10.1109/MGRS.2018.2853555.

48. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. Commun. ACM *63*, 139–144. https://doi.org/10.1145/3422622.

49. Makoś, M.Z., Verma, N., Larson, E.C., Freindorf, M., and Kraka, E. (2021). Generative adversarial networks for transition state geometry prediction. J. Chem. Phys. *155*, 024116. https://doi.org/10.1063/5.0055094.

50. Lebese, T., Mellado, B., and Ruan, X. (2021). The use of generative adversarial networks to characterise new physics in multi-lepton final states at the LHC. Int. J. Mod. Phys. A. https://doi.org/10.48550/arxiv.2105.14933.

51. Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. 34th Conference on Neural Information Processing Systems (NeurIPS 2020), *33*, pp. 6840–6851.

52. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Cui, B., and Yang, M.-H. (2022). Diffusion models: a comprehensive survey of methods and applications. Preprint at arXiv. https://doi.org/10.48550/arxiv.2209.00796.

53. Marcus, G., Davis, E., and Aaronson, S. (2022). A very preliminary analysis of DALL-E 2. Preprint at arXiv. https://doi.org/10.48550/arxiv.2204.13807.

54. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. Preprint at arXiv. https://doi.org/10.48550/arxiv.2205.11487.

55. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 10674–10685. https://doi.org/10.1109/CVPR52688.2022.01042.

56. Yang, Z., Yu, C.H., Guo, K., and Buehler, M.J. (2021). End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures. J. Mech. Phys. Solids *154*, 104506. https://doi.org/10.1016/J.JMPS.2021.104506.

57. Yang, Z., Yu, C.H., and Buehler, M.J. (2021). Deep learning model to predict complex stress and strain fields in hierarchical composites. Sci. Adv. *7*. https://doi.org/10.1126/sciadv.abd7416.

58. Buehler, M.J. (2022). FieldPerceiver: domain agnostic transformer model to predict multiscale physical fields and nonlinear material properties through neural ologs. Mater. Today *57*, 9–25. https://doi.org/10.1016/J.MATTOD.2022.05.020.

59. Ni, B., and Gao, H. (2021). A deep learning approach to the inverse problem of modulus identification in elasticity. MRS Bull. *46*, 19–25. https://doi.org/10.1557/s43577-020-00006-y.

60. Buehler, M.J. (2022). Modeling atomistic dynamic fracture mechanisms using a progressive transformer diffusion model. J. Appl. Mech. *89*, 121009. https://doi.org/10.1115/1.4055730.

61. Lin, Z., Fair, T.S., Lecun, Y., and Rives, A. (2021). Deep generative models create new and diverse protein structures. Machine Learning for Structural Biology Workshop, NeurIPS 2021.

62. Anand, N., and Achim, T. (2022). Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. Preprint at arXiv. https://doi.org/10.48550/arxiv.2205.15019.

63. Trippe, B.L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. (2022). Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. Preprint at arXiv. https://doi.org/10.48550/arxiv.2206.04119.

64. Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. (2022). High-resolution de novo structure prediction from primary sequence. Preprint at bioRxiv. https://doi.org/10.1101/2022.07.21.500999.

65. Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., and Steinegger, M. (2022). ColabFold: making protein folding accessible to all. Nat. Methods *19*, 679–682. https://doi.org/10.1038/s41592-022-01488-1.

66. Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers *22*, 2577–2637. https://doi.org/10.1002/BIP.360221211.

67. Evans, R., O'Neill, M., Pritzel, A., Antropova, N., Senior, A., Green, T., Žídek, A., Bates, R., Blackwell, S., Yim, J., et al. (2022). Protein complex prediction with AlphaFold-Multimer. Preprint at bioRxiv. https://doi.org/10.1101/2021.10.04.463034.

68. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. (1990). Basic local alignment search tool. J. Mol. Biol. *215*, 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2.

69. Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R.J., Milles, L.F., Wicky, B.I.M., Courbet, A., de Haas, R.J., Bethel, N., et al. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. Science *378*, 49–56. https://doi.org/10.1126/SCIENCE.ADD2187/SUPPL_FILE/SCIENCE.ADD2187_SM.PDF.

70. Ingraham, J., Baranov, M., Costello, Z., Frappier, V., Ismail, A., Tie, S., Wang, W., Xue, V., Obermeyer, F., Beam, A., et al. (2022). Illuminating protein space with a programmable generative model. Preprint at bioRxiv. https://doi.org/10.1101/2022.12.01.518682.

71. Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E./, Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., et al. (2022). Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. Preprint at bioRxiv. https://doi.org/10.1101/2022.12.09.519842.

72. Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G.R., Wang, J., Cong, Q., Kinch, L.N., Dustin Schaeffer, R.D., et al. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. Science *373*, 871–876. https://doi.org/10.1126/SCIENCE.ABJ8754/SUPPL_FILE/ABJ8754_MDAR_REPRODUCIBILITY_CHECKLIST.PDF.

73. Keten, S., Xu, Z., Ihle, B., and Buehler, M.J. (2010). Nanoconfinement controls stiffness, strength and mechanical toughness of β-sheet crystals in silk. Nat. Mater. *9*, 359–367. https://doi.org/10.1038/nmat2704.

74. Keten, S., and Buehler, M.J. (2008). Asymptotic strength limit of hydrogen-bond assemblies in proteins at vanishing pulling rates. Phys. Rev. Lett. *100*, 198301. https://doi.org/10.1103/PhysRevLett.100.198301.

75. Moriwaki, Y. (2021). AlphaFold2 can also predict heterocomplexes. All you have to do is input the two sequences you want to predict and connect them with a long linker. Twitter. https://t.co/BhmWcnlQed.

76. Baek, M. (2021). Adding a big enough number for "residue_index" feature is enough to model hetero-complex using AlphaFold (green&cyan: crystal structure/magenta: predicted model w/ residue_index modification). #AlphaFold #alphafold2. Twitter. https://t.co/TX1PnRk5Wd.

77. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: an imperative style, high-performance deep learning library. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), *vol 32*.

78. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16). usenix.org.

79. GitHub. lucidrains/imagen-pytorch: implementation of Imagen, Google's text-to-image neural network, in Pytorch. https://github.com/lucidrains/imagen-pytorch.

80. Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. Preprint at arXiv. https://doi.org/10.48550/arxiv.2206.00364.

81. Nichol, A.Q., and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models Proceedings of the 38th International Conference on Machine Learning, PMLR*139*, pp. 8162–8171.

82. Ho, J., and Salimans, T. (2022). Classifier-free diffusion guidance. Preprint at arXiv. https://doi.org/10.48550/arxiv.2207.12598.

83. Kingma, D.P., and Ba, J.L. (2014). Adam: a method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. https://doi.org/10.48550/arxiv.1412.6980.

84. Du, Z., Su, H., Wang, W., Ye, L., Wei, H., Peng, Z., Anishchenko, I., Baker, D., and Yang, J. (2021). The trRosetta server for fast and accurate protein structure prediction. Nat. Protoc. *16*, 5634–5651. https://doi.org/10.1038/s41596-021-00628-9.

85. Schrodinger (2015). The PyMOL molecular graphics system, version 1.8. https://pymol.org/2/.

86. Rego, N., and Koes, D. (2015). 3Dmol.js: molecular visualization with WebGL. Bioinformatics *31*, 1322–1324. https://doi.org/10.1093/BIOINFORMATICS/BTU829.