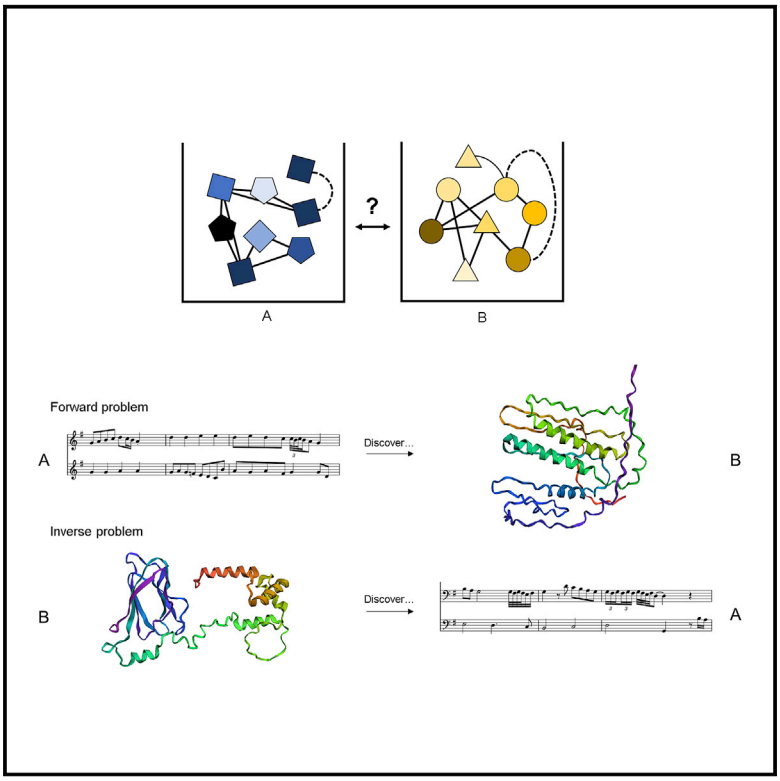
Article

Patterns

Unsupervised cross-domain translation via deep learning and adversarial attention neural networks and application to music-inspired protein designs

Graphical abstract



Authors

Markus J. Buehler

Correspondence

mbuehler@mit.edu

In brief

In this paper we report a method that allows us to discover how patterns in disparate domains can be reversibly related using a computationally rigorous approach, the AttentionCrossTranslation model. The algorithm discovers cycle- and self-consistent relationships and offers a bidirectional translation of information across disparate knowledge domains. The approach is used to discover a mapping between musical data contained in J.S. Bach's Goldberg Variations created in 1741 and protein sequence data sampled more recently.

Highlights

- Patterns in disparate domains are reversibly related
- We propose the AttentionCrossTranslation model, a deep attention neural network
- The model discovers cycle- and self-consistent relationships
- We apply the method to translations between music and proteins, and vice versa

Buehler, 2023, Patterns *4*, 100692
March 10, 2023
https://doi.org/10.1016/j.patter.2023.100692

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Unsupervised cross-domain translation via deep learning and adversarial attention neural networks and application to music-inspired protein designs

Markus J. Buehler[1,2,3,4,5,*]
[1]Laboratory for Atomistic and Molecular Mechanics (LAMM), Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[2]Center for Computational Science and Engineering, Schwarzman College of Computing, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[3]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[4]Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
[5]Lead contact
*Correspondence: mbuehler@mit.edu
https://doi.org/10.1016/j.patter.2023.100692

THE BIGGER PICTURE Human creativity has advanced the way we understand the world, including scientific and artistic modalities. However, until now, the convergent use of disparate knowledge bases has remained elusive, especially connecting art and science. To address this challenge, we report a method to achieve such translations using deep learning, whereby salient relationships are discovered in an unsupervised fashion, without knowledge of pairing or domain knowledge, thereby expanding the concept of bio-inspiration to encompass vast swaths of human knowledge. The method is demonstrated in the reversible, bidirectional translation of musical data (based on Bach's Goldberg Variations) to protein sequences, discovered in a fully autonomous manner. The general method has broader applications for other discovery platforms in a variety of engineering, scientific, cultural, artistic, and environmental data, opening many possibilities for further experimental and computational studies of pattern engineering.

Concept: Basic principles of a new data science output observed and reported

SUMMARY

Taking inspiration from nature about how to design materials has been a fruitful approach, used by humans for millennia. In this paper we report a method that allows us to discover how patterns in disparate domains can be reversibly related using a computationally rigorous approach, the AttentionCrossTranslation model. The algorithm discovers cycle- and self-consistent relationships and offers a bidirectional translation of information across disparate knowledge domains. The approach is validated with a set of known translation problems, and then used to discover a mapping between musical data—based on the corpus of note sequences in J.S. Bach's Goldberg Variations created in 1741—and protein sequence data—information sampled more recently. Using protein folding algorithms, 3D structures of the predicted protein sequences are generated, and their stability is validated using explicit solvent molecular dynamics. Musical scores generated from protein sequences are sonified and rendered into audible sound.

INTRODUCTION

Taking inspiration from nature about how to design materials has been a popular and fruitful approach for millennia.[1,2] A persistent challenge, however, is the translation of insights across disparate domains—such as using patterns or mechanisms found in the construction of language to inform novel design paradigms in polymer design, or their use in the generation of proteins where we seek to arrange amino acid building blocks.[3] Earlier work proposed a materiomics approach based on category theory[4,5];

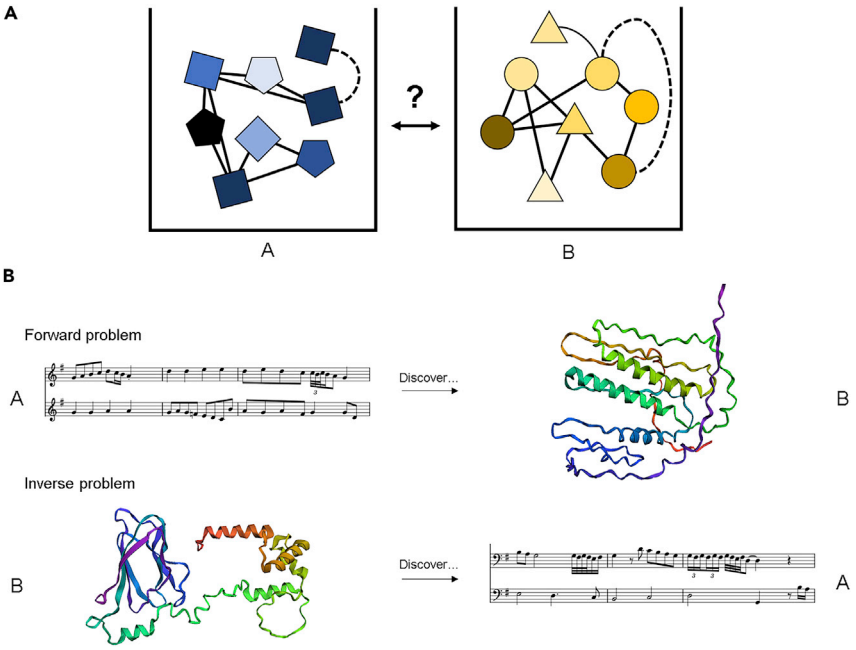


Figure 1. Summary of the AttentionCrossTranslation model that discovers cycle- and self-consistent relationships and offers a bidirectional translation of information across disparate knowledge domains

(A) This paper focuses on the translation of information across disparate domains, A and B. In (B) the task is visualized for the translation of music to proteins (top), and vice versa (bottom). Such translational problem cannot be solved using conventional methods as they lack the knowledge of translational principles and any knowledge of pairings between input and output. In this paper we develop and apply an unsupervised deep learning method that can discover relationships between such datasets using cycle-consistent transformer neural networks.

however, these concepts had limited real-world applicability since they are difficult to construct and require *a priori* knowledge of the entire set of relationships between building blocks. But, for many problems we do not know these—and they are, in fact, precisely what we seek to discover. In such scenarios we hypothesize that construction principles exist in each domain, but we do not know any closed-form mathematical expression that delineates how they can be translated across the domains. To provide a specific example—one that will be solved in this paper—we understand that music follows certain patterns in the way notes are arranged (depending on the style of music, the era of composition, etc.).[6–19] On the other hand, we also understand that protein design follows certain patterns in the way their amino acid building blocks are arranged, including for the determination of mechanical properties.[20–25] But what is the intersectionality between patterns in music and proteins, and how would we translate music to proteins, and vice versa? Can we identify, using an algorithmic tool, a relationship between these two representations, both directions (Figure 1)?

To provide additional descriptions of the music and protein data used in this study, we provide a deeper look into these two domains. Music typically consists of sound produced across a temporal history, where one or more notes are played in succession. In Western tradition, for instance, this information is often summarized in the staff, a set of five horizontal lines that show how the pitch changes over time (see Figure 1B under "forward problem" for an example). If music is performed, for instance, on a piano, the performer can deduce from the information in the staff what key to hit and at what time. For digital methods in music analysis and performance, each note can be assigned an identification number that designates a key on a piano, and that is associated with a certain fundamental frequency that defines its pitch. In the MIDI format commonly used for digital analyses of music there are 128 unique notes, from 0 to 127 (e.g., note number 69 is A4, concert pitch, reflecting 440 Hz and note number 81 is A5 one octave higher, reflecting 880 Hz). The effect music has on humans is the emergence of a multitude of factors, such as the sequence, timing, and volume of the notes played, and rendered more sophisticated once multiple note sequences are played jointly or in certain intersecting patterns (e.g., in classical music, we may have dozens of streams of intersecting players). Proteins are physical manifestations of certain types of genetic information. In this process, genetic sequences are information carriers that are transformed, through messenger RNA, into protein products. These strings of information are one-dimensional sequences that are read out by the ribosome through which, in a chemical process, protein segments are created by assembling amino acids (each amino acid defined by three DNA "letters") that thereby reflect the structural definitions encoded in the original genetic sequence. There are 20 naturally occurring amino acids, and by arranging these in complex patterns a myriad of distinct biological, chemical, and physical properties are obtained. More complex functional units can be obtained once multiple proteins interact; either to form filamentous materials (e.g., collagen, amyloid fibrils, actin, intermediate filaments, etc.) or when they engage with other proteins to form complexes where multiple building blocks interact synergistically to achieve a set of functional outcomes. This brief summary of the characteristics of music and protein sequences reveals that both are deep reservoirs of coded information that reveals itself at multiple scales and in multiple modalities. Furthermore, both sets of information resemble a transformation from low encoding space to high-dimensional functional space; and where optimization to achieve certain outcomes has occurred over a long period of time. In music, this has happened for at least thousands of years in humans when strictly considering what we refer to as "music" today, but more broadly when understood as the use of sound as a communication tool for billions of years). In the case of proteins, such adaptation has happened for billions of years and there are multiple length and time scales that evolutionary processes have considered—such as the immediate survival of an individual vs. the success of a species or group

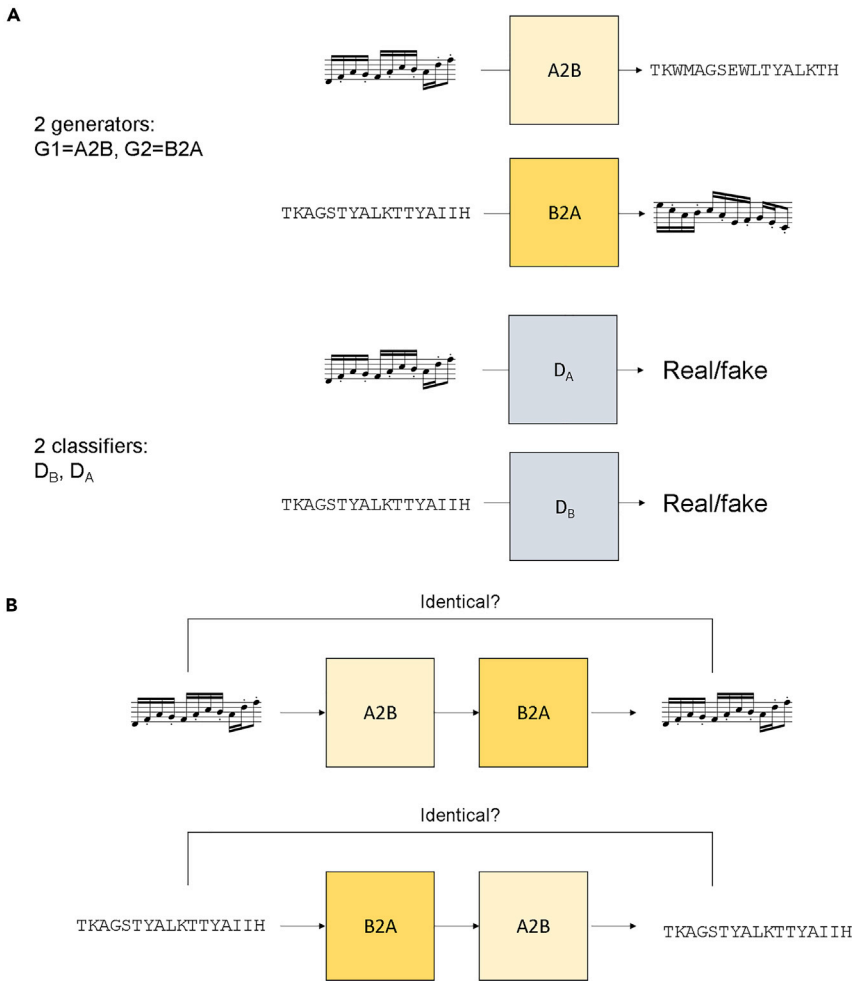


Figure 2. Overview of the translational algorithm, based on a set of four neural networks

The model consists of two generators A2B and B2A, and two classifiers D_A and D_B (A). These are used to construct a cycle-consistent neural network architecture and loss function (B). Detailed loss function definitions are provided in the experimental procedures. The generator nets translate information from one domain to another, whereas the discriminator neural nets are trained to determine whether the translations appear as "real" samples from the domains A and B, respectively.

of species. Some protein motifs have been discovered multiple times, independently, and others have been conserved into multitudes of animal families.

Earlier work[26] has attempted to map music with amino acid sequences by assigning a unique musical note to each amino acid, whereby the ordering of the association of the note-amino acid pairing was accomplished based on a calculation of the normal mode frequencies. Lower normal mode frequencies associated with certain amino acids were associated with lower musical notes and vice versa.[27] A unique association between all notes and amino acids was then developed using molecular mechanics principles, which enabled a direct translation of sequence data and even coding for higher-order structural detail. However, this approach has several shortcomings. These include the fact that mappings in the earlier work were constructed using normal mode frequencies, which may or may not have a mechanistic association with certain musical notes or progressions of multiple notes, as it is based on a choice to define an association mechanism. Furthermore, the number of unique notes or amino acids is limited by the minimum of the number of unique building blocks in each domain. Since there are only 20 unique amino acids, it was only possible to translate these into music with 20 unique notes. Conversely, only music with 20 or less unique notes could be translated into protein sequences. Moreover, the approach completely misses any larger-scale functional relationships that delineate how patterns (at different scales or hierarchical levels of information: e.g., pairs of amino acids, pairs of notes, and higher-order multiples) relate. And looking to other types of information modalities, many of these choices cannot be justified or lack complete rationale in terms of the physical world (e.g., language, or design principles, do not exist in a physical sense but are mathematically abstract entities). These are significant limitations that precluded a broader use of cross-domain translations from an engineering science perspective and are indicative of many other, similar problems that could not yet be solved using conventional methods.

A key reason why such translations have been challenging thus far is the high degree of complexity that different representations have in their own right, let alone the relationships between sets of them. We address these shortcomings and propose a fundamentally distinct way to think about this problem. First, we remove any preconceived notion about a building-block-level association between the two domains because it likely misses higher-level mechanisms of translations as well as other limitations discussed above. Also, a true translational mechanism would allow one to translate a sequence from one domain into another, and back, with cycle consistency to ensure that no information is lost during multiple forward and backward translation steps: A → B → A' = A, and B → A → B' = B. Finally, a formulation that is broadly applicable should work on any kind of data and modalities (sequence, vibrations, continuous data, field data, images, or voxels, etc.) as long as it is measurable and can be collected and made machine readable to form "buckets" of information that form the basis for the discovery algorithm (Figure 1A).

To achieve these goals, we develop a method to discover these relationships for translation directly from data, without any knowledge of paired relationships or an underlying physical law or known principle. We hypothesize that this sort of

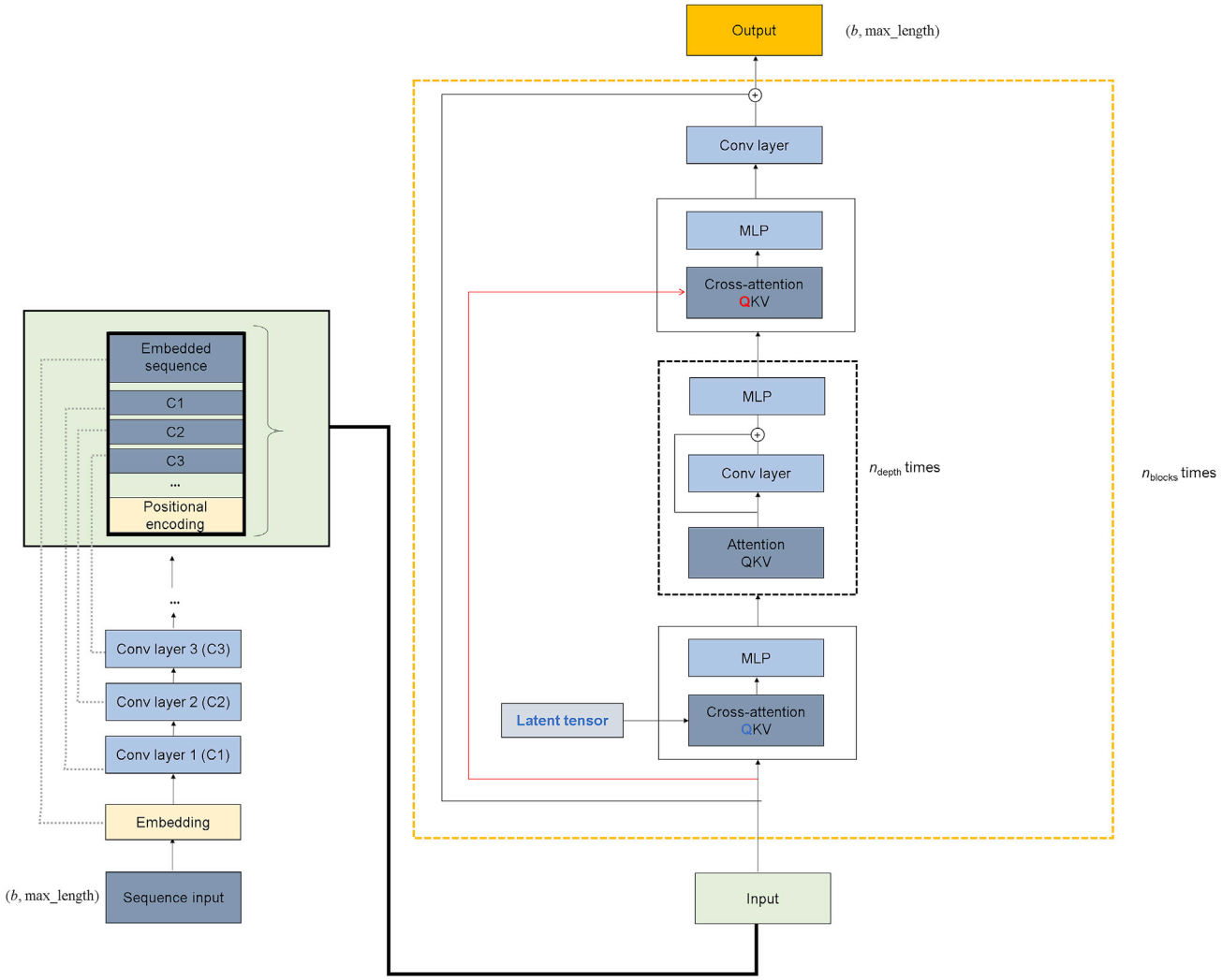


Figure 3. Architecture of the generator neural networks, from sequence input (lower left corner) to output (upper right corner), forming a hierarchical structure

The model consists of a combination of self-/cross-attention and convolutional layers, arranged in multiple blocks. The input sequence, for the problems treated here a 1D sequence, is processed via an embedding block that encodes the sequence input (left), providing the input for the hierarchical transformer encoder/decoder architecture. The sequence input and output have the same dimension to facilitate cycle-consistent losses.

problem can be addressed using the principle of self-attention and cross-attention,[28–30] and the use of cycle-consistent loss functions.[31–33] Cycle-consistent loss functions have been used successfully in earlier work to achieve translations between domains where no knowledge of input-output pairings exist, despite a relationship that is "hidden" in the data. Furthermore, a proper deep learning model must be constructed that has the learning capacity to discover extremely complex cross-domain relationships. The development of these aspects, and a validation of the approach, is presented in the next sections.

RESULTS AND DISCUSSION

We design a sophisticated deep learning framework that consists of four individual neural networks. Figure 2 shows an overview of the game-theory-based adversarial neural network approach used here, based on the set of four neural networks: two generators A2B and B2A, and two classifiers D_A and D_B (Figure 2A). These are used to construct a cycle-consistent neural network architecture and loss functions (Figure 2B, experimental procedures).[31–33] Detailed loss function definitions are provided in the experimental procedures section, along with the associated discussion for further details. Figure 3 shows the architecture of the two generator neural networks, from sequence input (lower left corner) to output (upper right corner). The model consists of a combination of self-/cross-attention and convolutional layers, arranged in multiple blocks, to facilitate a deep learning capacity for highly complex relationships within and across modalities embedded in each of the modalities. The input signal is processed via an embedding block that encodes the sequence input (left), then providing the input for the hierarchical

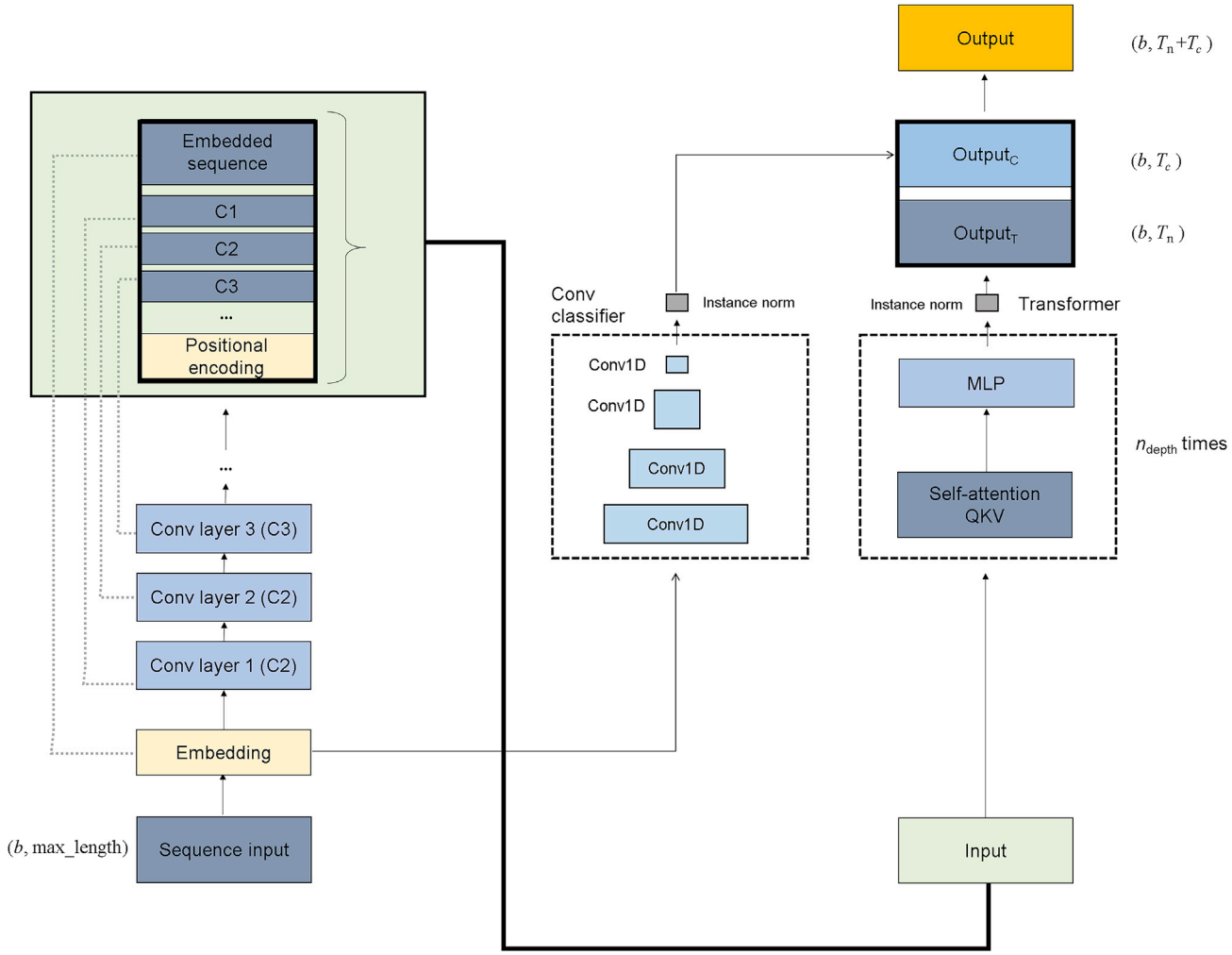


Figure 4. Architecture of the discriminator neural networks formulated as a patch-style classifier, from sequence input (lower left corner) to output (upper right corner)

The model consists of an embedding block that encodes the sequence input (left), which is then processed in a multi-layer self-attention block (right), yielding the output.

transformer encoder/decoder architecture, ultimately yielding the translated output signal. The input and output have the same dimension to facilitate cycle-consistent losses, as the method must be able to translate forward and backward, as well as cycle translations. Similarly, Figure 4 depicts the architecture of the discriminator neural networks formulated as a patch-level classifier, from sequence input (lower left corner) to output (upper right corner). The model consists of an embedding block that encodes the sequence input (left), which is then processed in a multi-layer self-attention block (right), yielding the output. It is noted that for a high-complexity problem as treated here, the design of these four neural network architectures is critical. As described in detail in the experimental procedures section, a deep multi-layer attention-based neural network architecture is designed so that it has a large capacity to discover complex relationships across the data modalities and accurately characterize their features via a patch-style classifier architecture.

We now apply this neural network architecture to a series of simple problems. The sample problems are created with the objective to probe specific aspects of the translation challenge, whereas the choices of the numerical parameters in the equations defined below are not critical. First, we consider simple problems for which we know the relationship between the two modalities with the objective to test the approach and validate its function. Then, we generalize the task and apply the approach to a dataset with unknown relationships. Figure 5 shows the details of several simple sample datasets to test the method developed here. We refer to data 1 as *x* and data 2 as *y* in the discussion of these datasets, reflecting the two domains of representation (referred to as A and B above). Figure 5A, case 1 shows the input and output data. First, data 1 are generated by picking alternating integer numbers 1 and 2, whereby each block of 1s and 2s is repeated n_i times (n_i sampled uniformly from the set {6, 7, …, 23, 24}). This sampling is repeated until we reach a sequence of length $\max_{length}$. Data 2, *y*, features a

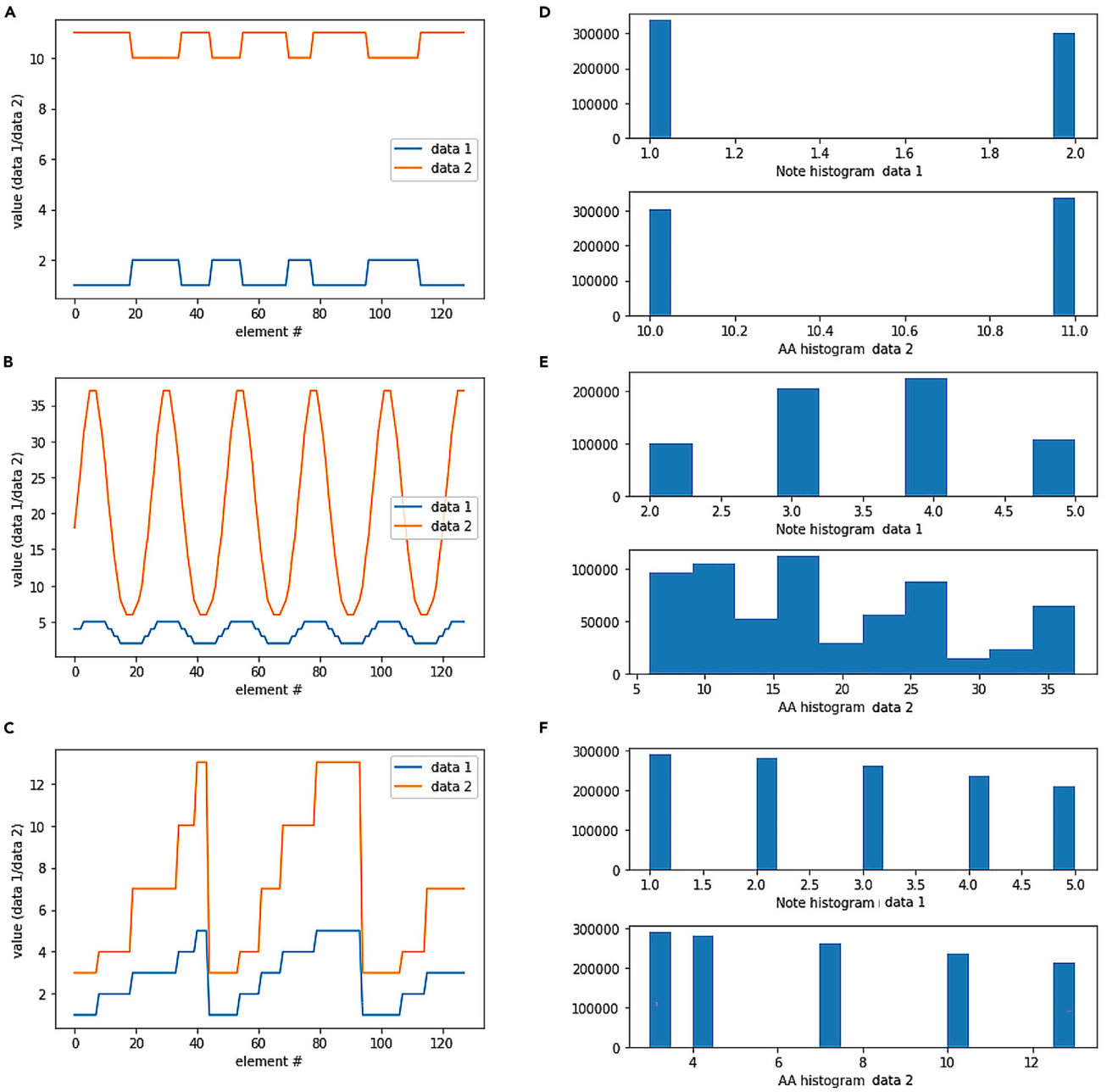


Figure 5. Simple sample datasets used to test the method developed here

(A) Case 1: input and output data (data 1, data 2, respectively) feature a simple mathematical relationship (shifted, negative multiplied token values). The input is generated based on alternating values repeated *n* times, where *n* is a random number.

(B) Case 2: a sinusoidal relationship between data 1 and data 2, realized using a simple power law that is executed element-wise.

(C) Case 3: like in case 2, input and output data (data 1, data 2, respectively) are mathematically related, but here we deal with a step function. (D), (E), and (F) show a statistical analysis of the tokens in the two datasets.

simple mathematical relationship to data 1, x, where they are shifted, negative multiplied token values:

$$y = a - x \qquad \text{(Equation 1)}$$

where $a = 12$, leading to alternating values between 10 and 11 (whereas $y = 10$ is associated with $x = 2$ and $y = 11$ is associated with $x = 1$). As shown in Figure 5D, a histogram of the data shows that both data 1 and data 2 feature the same number of tokens and the same distribution.

Figure 5B, case 2 shows the second example, a sinusoidal relationship between data 1 and data 2, realized using a simple power law that is executed element-wise:

$$x = p \sin\left(\frac{i}{q}\pi\right) \; i = 1..\max_{\text{length}} \qquad \text{(Equation 2)}$$

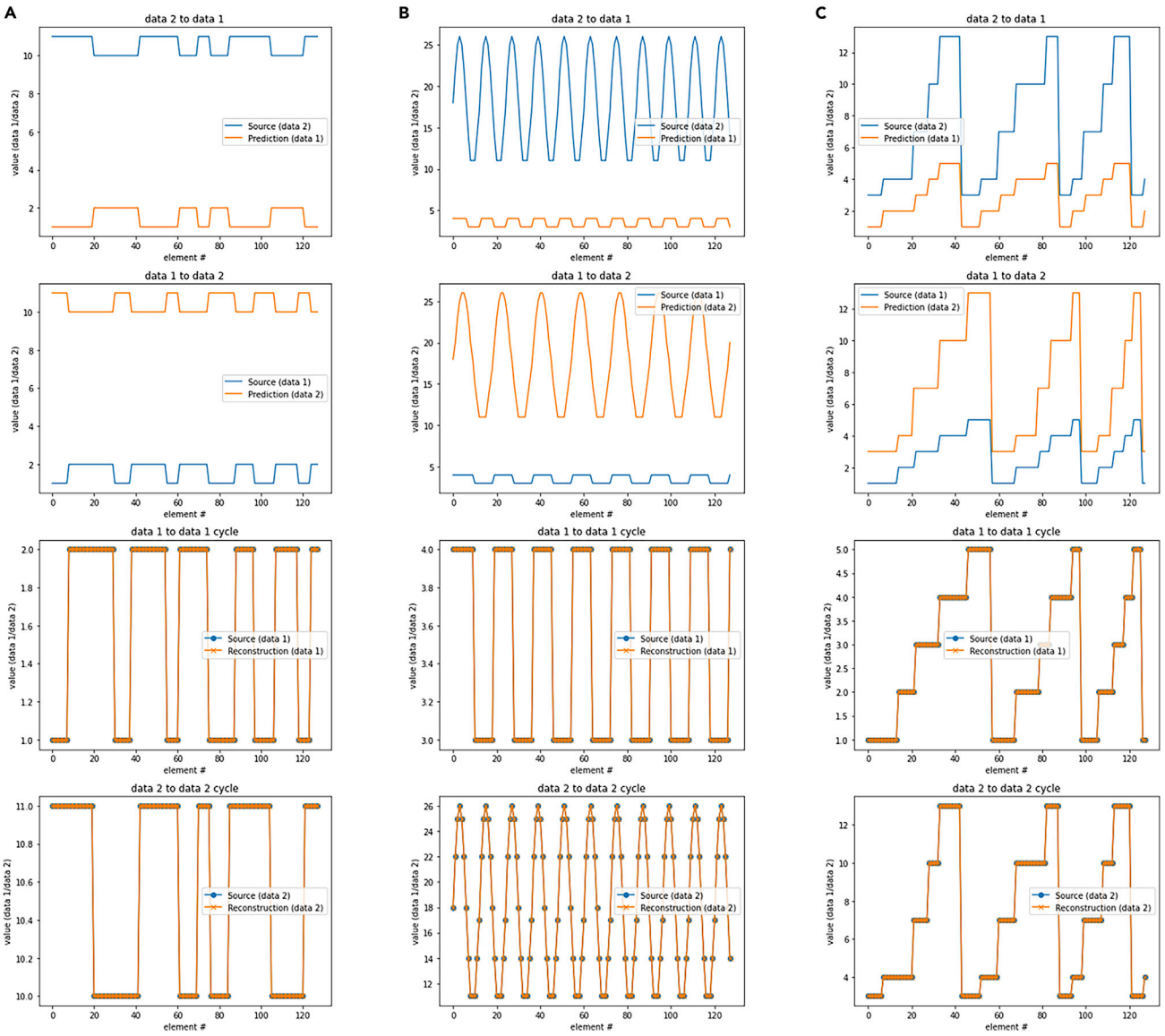


Figure 6. Results of the computational experiments based on the datasets described in Figure 5

The analysis is carried out for the three cases case 1 (A), case 2 (B), and case 3 (C). In each column, from top to bottom, the rows show: (1) sample translations from data 1 to data 2, (2) sample translations from data 2 to data 1, (3) a cycle of translation from data 1 to data 2 back to data 1, and (4) a cycle of translation from data 2 to data 1 back to data 2. The first two rows elucidate how the model designates translations. For all cases, the translation captures the mathematical relationship used in the creation of the dataset. The cycle mappings (rows 3 and 4) are conducted perfectly without error. Figure S5 shows a statistical analysis for the predictions from the test sets, revealing close agreement with the original data distributions shown in Figures 5D–5F.

and

$$y = x^2 + a \qquad \text{(Equation 3)}$$

where $a = 2$, p is sampled uniformly from the set {1, 2, 3} and q is sampled uniformly from the set {2, 3, …, 15, 16}. Now, the statistics of the input and output data is different, as shown in Figure 5E, where both the number of tokens and the distributions are distinct.

Figure 5C, case 3 is similar to case 2 where input and output data (data 1, data 2, respectively) are mathematically related, but here we deal with a repetitive step function that is generated based a periodic variation of input tokens 1, 2, 3, 4, 5, each repeated n_i times (each n_i is sampled uniformly from the set {6, 7, …, 15, 16}). Then, similar to what was defined in Equation 1, here we use:

$$y = x^{1.5} + a \qquad \text{(Equation 4)}$$

where $a = 2$. In this case, the two sets of data have the same number of tokens but a different distribution, as can be verified in Figure 5F.

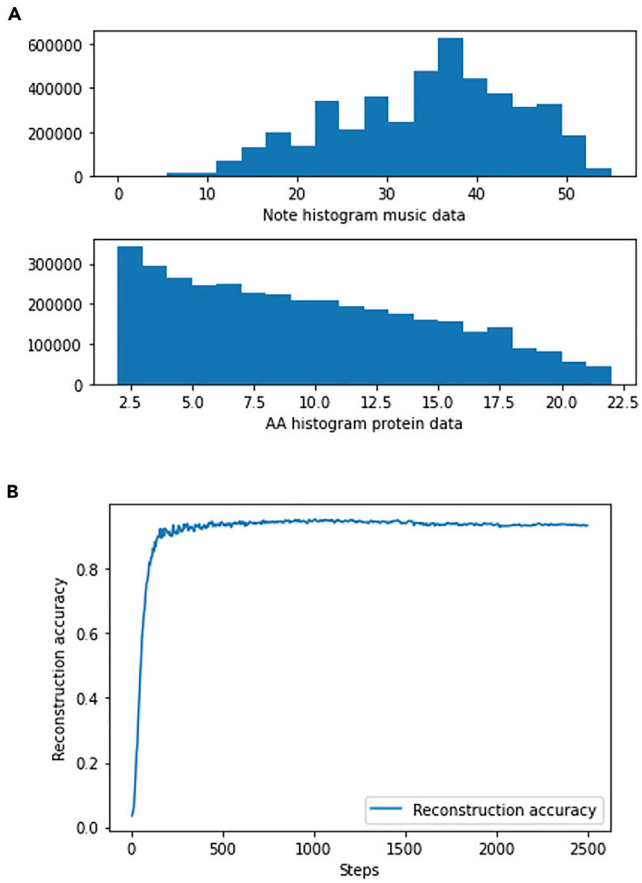


Figure 7. Statistical analysis of the note and protein datasets, showing histograms of the input tokens and reconstruction accuracy

(A) Statistical analysis of the note and protein datasets, showing histograms of the input tokens.

(B) Reconstruction accuracy, over training steps, for validation samples that were not included in the training. The reconstruction accuracy measures how accurate cyclic reconstructions are (music to protein to music and protein to music to protein). The overall accuracy reaches ~0.95.

Figure 6 depicts the results of the computational experiments based on the datasets described in Figure 5 (for the three cases case 1, Figure 6A; case 2, Figure 6B; and case 3, Figure 6C). In each column, from top to bottom, the rows show: (1) sample translations from data 1 to data 2, (2) sample translations from data 2 to data 1, (3) a cycle of translation from data 1 to data 2 back to data 1, and (4) a cycle of translation from data 2 to data 1 back to data 2. The first two rows elucidate how the model designates translations. For all cases, the translation captures the mathematical relationship used in the creation of the dataset, as defined in the equations above. The cycle mappings (rows 3 and 4) are conducted perfectly, without error. For further validation from a statistical perspective, Figure S1 shows a statistical analysis for the predictions from the test sets, revealing close agreement with the original data distributions shown in Figures 5D–5F.

As a first step toward a more complex, unknown translational problem, we consider the case presented in Figure S2. Shown therein is a similar translation problem as those depicted in Figures 5 and 6. However, in this case, we use data from two randomly generated, independent distributions that do not have any known relationship but feature a similar characteristic (it can be verified visually via the graphs shown in Figure S2A). In this case, data 1 is generated by picking alternating integer numbers 1 and 2 that define the input tokens, whereby each block of 1s and 2s is repeated $n_{i,1}$ times ($n_{i,1}$ sampled uniformly from the set {6, 7, …, 23, 24}). Data 2 is constructed in the same way, except that we use alternating integer numbers 8 and 10 that define the tokens used, which are then repeated $n_{i,2}$ times ($n_{i,2}$ sampled uniformly from the set {6, 7, …, 23, 24}; noting that $n_{i,j}$ are independently sampled). Figure S2B shows the results akin to the data depicted in Figure 6, revealing the translational mechanism that the model has discovered in order to translate between the two modalities. Even though we have not used a direct mathematical relationship between x and y, the model has identified an adequate translational mechanism, similar as the one defined in Equation 1, but featuring a slight variation given the fact that the original data varies between 1 and 2 and 8 and 10, respectively:

$$y = a - 2(x - 1) \qquad \text{(Equation 5)}$$

where $a = 10$.

This result, having discovered a translational principle between two modalities of information, is important to set the stage for the next computational experiments for the music-protein dataset.

Figure 7A summarizes a statistical analysis of the music and protein datasets, showing histograms of the input tokens. Both the number of tokens and the distribution is distinct between the two datasets; but based on the experiments discussed in the previous paragraph we hypothesize that that the method can still be applied. Indeed, the model is able to identify suitable translations between these two distinct sets of information. Figure 7B shows the reconstruction accuracy, over training steps, for validation samples that were not included in the training. The reconstruction accuracy measures how accurate cyclic reconstructions are (music to protein and back to music and protein to music back to protein). This is a measure for how well the model can encode information across modalities, and how much information is lost during the translations. The overall accuracy, calculated over a validation set, reaches ~0.95, which indicates excellent performance. The behavior of the loss functions during training is depicted in Figures S1 and S4, which show learning rates for generators and classifiers over training steps.

Digging deeper into the results, Figure 8 now shows sample translations from music to protein, from protein to music, and complete reconstruction cycles, for two representative cases (all performed using a set of validation samples that were not included in the training). The statistics of the input and output, as shown by the analysis in the right panel of each case, are well reproduced. This shows that the overall reconstruction accuracy, the specific translations, and the statistics innate to each protein or musical design, perform well. For a more global analysis of the behavior, Figure 9 depicts statistics of the original data, recovered data, and generated data. The statistics

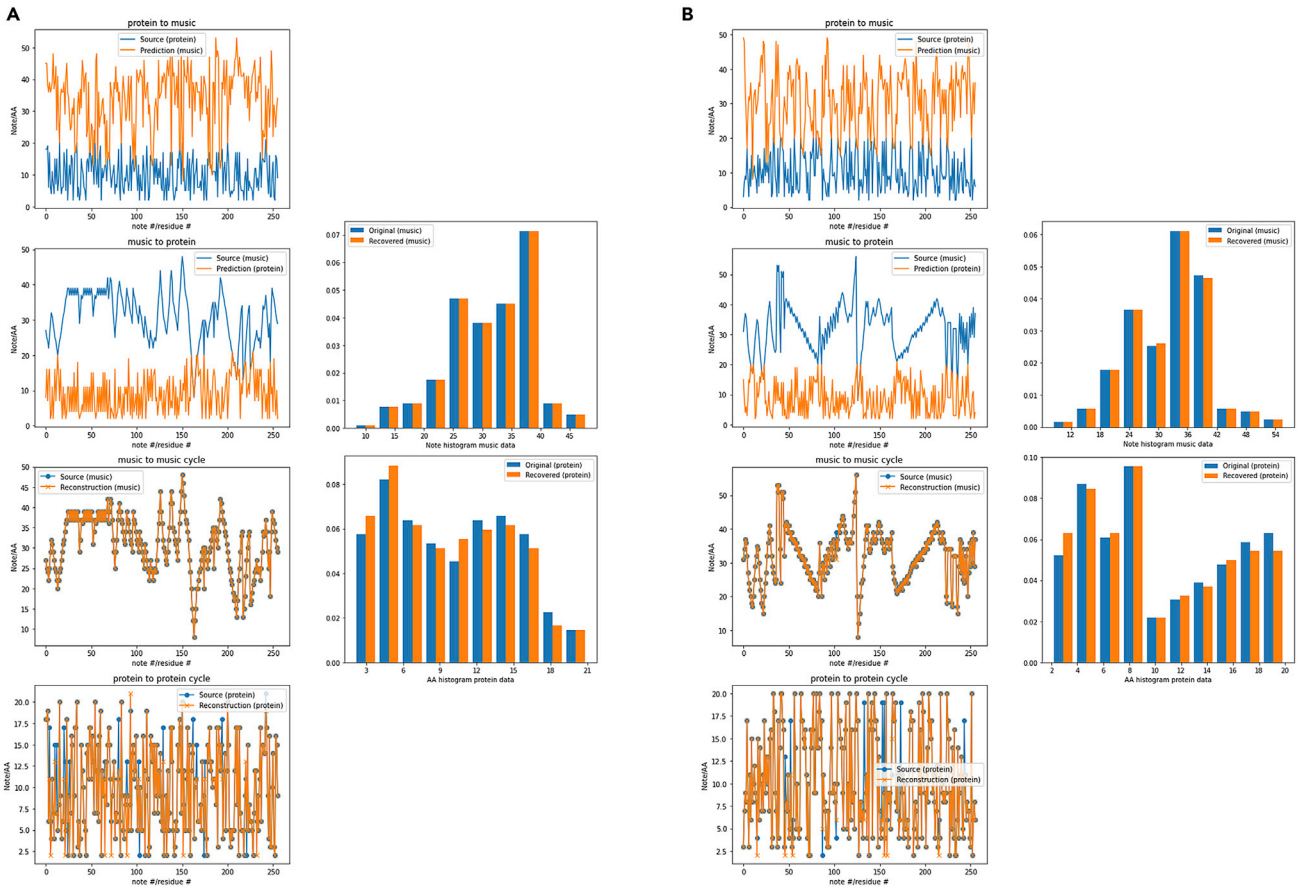


Figure 8. Sample translations from music to protein, from protein to music, and complete reconstruction cycles

The results are shown for two representative cases depicted in (A) and (B) (for validation samples that were not included in the training). The statistics of the input and output, as shown by the analysis in the right panel of each case, are well reproduced. There are additional features in (B) that are of interest, such as how continually rising or falling note sequences are translated. The context for this aspect is particularly intriguing when considering the sequence direction of proteins vs. that of music; where retrograde counterpoint is a means by which melodic fragments are inverted with respect to time. A detailed look at the data in (B) reveals that rising or falling note sequences, for instance, are not translated with regard to protein data with similar characteristics but rather encoded in more complex intricate patterns. Similar considerations apply to the data shown in Figure S6.

are calculated over 1,024 validation samples that were not included in the training. By comparing to the original statistics (blue in each case), it can be seen that the statistical properties are well preserved, both in the reconstruction (orange) and in the generation novel designs (green), in each domain. Both music data and protein data show similar performance, overall. There are additional features in Figure 8B that are of interest, such as how continually rising or falling note sequences are translated. The context for this aspect is particularly intriguing when considering the sequence direction of proteins vs. that of music; where retrograde counterpoint is a means by which melodic fragments are inverted with respect to time. A detailed look at the data in Figure 8B reveals that rising or falling note sequences, for instance, are not translated to protein data with similar characteristics but rather encoded in more complex intricate patterns.

Now that we have established that the model can appropriately conduct translations in both directions, we test the approach for a protein design application. In this application we seek to design new protein sequences that can then be explored using protein folding methods and molecular modeling. To demonstrate the concept, we pick a sequence of music from the validation set and translate it into the corresponding protein sequence. Figure 10 shows the results of this experiment. We also examine the corresponding all-atom 3D protein structure obtained based on the sequence predictions using AlphaFold 2.[34] To complement the analysis, Figure S5 displays a histogram of distributions of notes and amino acids, respectively (these are, indeed, similar to the data shown in Figure 8 for the earlier examples, with similar performance as seen in the earlier example). Similarly, Figure S6 shows translation data of the results shown in Figure 10. Audio files A1 and A2 feature the corresponding audio files (Audio A1, source music; Audio A2, prediction music generated from protein). A third audio file A3 is presented where the two sequences are played on the left and right channel, respectively, staggered by a $1/8^{th}$ note to offer clear audible recognition of each note signal, respectively. Such experiments can yield interesting insights into the type of musical structure produced when the original Bach music is mingled with the music generated from proteins. Figure S7 depicts an illustration of

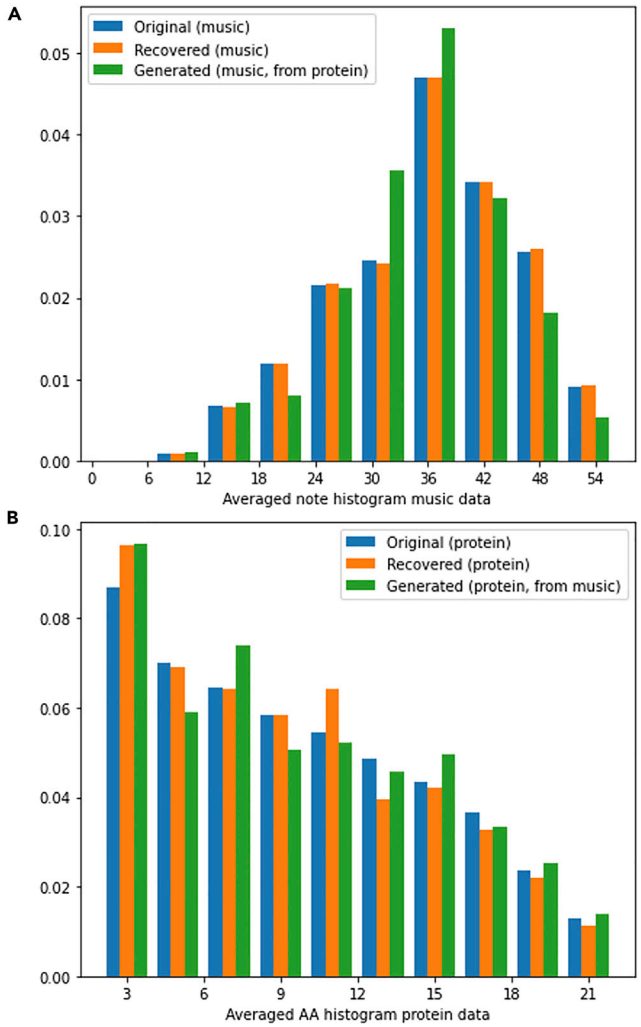


Figure 9. Statistics of the original data, recovered data, and generated data for both modalities, music and proteins

Results for (A) music and (B) proteins. The statistics are calculated over 1,024 validation samples that were not included in the training. Because we are considering a large number of samples, the overall statistical qualities of the two datasets are uncovered here, similar to what is shown in Figure 7. This is different from the analysis in the right panels of Figures 8A and 8B, where particular features of specific protein or musical data were captured.

the translation in their final receptive domains; here, protein geometry in 3D space (left) to frequency data in 2D frequency-over time space (right).

Various associated PDB files are also attached in the supplemental information. We analyzed the predicted protein sequence and did not find any hits using the basic local alignment search tool (BLAST),[35] suggesting that the designs generated were not (yet) identified by natural evolutionary principles. To test the stability of the protein generated from music data, we conduct explicit water molecular dynamics simulations (Figure S8A). Results are shown in Figure S8, revealing that the protein structure remains stable. Figure S8B shows snapshots of the protein structure across the molecular dynamics (MD) simulation, confirming that no major change occurs. Figure S8C shows per-residue root-mean-square displacement (RMSD) values comparing the final step to the first step as reference; showing that no region undergoes unfolding and that the protein simply equilibrates. Figure S8C depicts a visualization of the data shown in Figure S8D from two distinct angles for overall perspective. Finally, Figure S8E shows the overall RMSD over the entire trajectory, and Figure S8F depicts the number of H-bonds over the simulation trajectory, showing that no local unfolding occurs as the number of H-bonds remains approximately constant. Future experimental work for this and other proteins could be carried out to affirm these results in a laboratory setting. Other computational experiments may be conducted to explore how different musical passages relate with certain protein structures and functions, and whether or not they may be combined with natural protein designs.

Conclusion

In this paper we report the AttentionCrossTranslation model, which enables us to create translations between disparate domains of information modalities. Facilitated by four transformer-based neural network architectures, we conducted first a translation of simple patterns, then expanded it toward translation from musical scores to proteins, and vice versa. The method successfully learned translational relationships, as shown in Figures 6, 8, and 10, and with more results shown in Figure S7.

The general mechanism of developing translational relationships between sequence data can find applications in various science and engineering domains, including other types of sequence data (e.g., DNA), a subset of proteins, or a variety of signals sourced from natural systems including animal tracking data or evolutionary patterns. It may also help us to understand how proteins change over generations of mutations and across species, enabling the identification of the role of conserved and more highly variegated structural principles.

While we have shown that the model can correctly identify the correct principles for systems that were constructed based on mathematical principles (Figures 5 and 6), we do not know the correct translational principles for music-to-protein translations. While this can be viewed as a limitation, it does open the door for novel protein designs. We studied several of the sequences generated from Bach's music, and did not find any hits using BLAST,[35] suggesting that the designs are novel and do not exist in nature yet stable as confirmed via explicit solvent MD simulations. The spectrum of novel protein designs made accessible through the tool developed in this paper deserves future investigations, especially whether these proteins can be synthesized experimentally and what functions they would have, and what their uses may be in the medical or biological fields.

Other opportunities for future research work include the consideration of multi-dimensional features of musical data, such as timing, dynamics, and note length. These can, in principle, easily be added in the embedding dimension—simply generating a deeper tensor to be processed by the neural networks (the method can directly be used for higher-dimensional data with more than one dimension). It would be interesting to

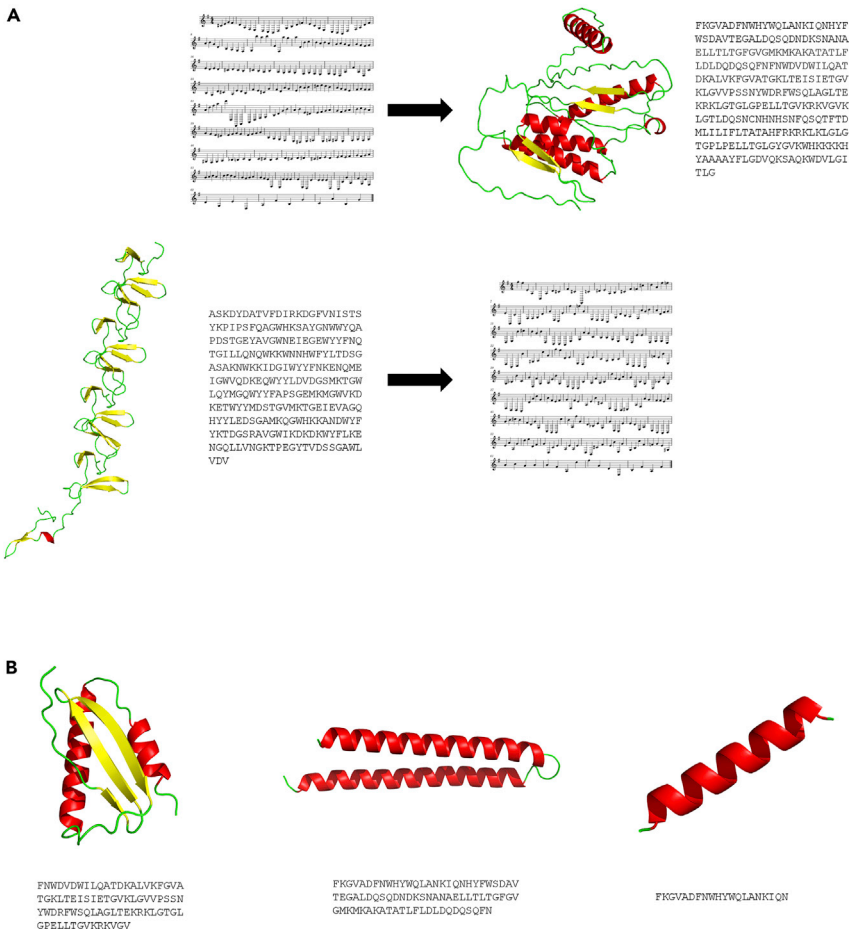


Figure 10. Molecular modeling analysis of the predicted protein structures based on musical data

(A) Predicted protein structure, based on two translations, for validation samples that were not included in the training. Figure S3 shows a histogram of distributions of notes and amino acids, respectively. Audio files A1 and A2 feature the corresponding audio files (Audio A1, source; Audio A2, prediction; and Audio A3 a staggered overlay of the data presented in A1 and A2).

(B) Predicted protein structures for part of the sequence predicted in (A). Five corresponding PDB files are provided as supplemental information.

see how protein data—which feature identical parameters for each constituting building block—are translated into music-specific features that include additional dimensions of expression. This can also include specific human characteristics; where research could explore how distinct human players provide unique features in timing and dynamism and how that translates into distinct architectural and functional relationships across manifestations. For instance, would a Bach suite played by Glenn Gould yield a different protein than a Bach suite played by Yo Ma? Such analysis may shed more light into the deeper aspects of how, and what kind of, information is carried through performance aspects as opposed to the "clinical" transcription of music in scores.

Another interesting exploration can be temporal direction. Protein data are read and transcribed from the C- to the N-terminal. While music data in a staff are also read in a certain direction (e.g., left to right) there are musical methods that reverse parts of note sequences, such as defined by the distinct mechanisms by which counterpoint is created. In particular, retrograde (a melody is played backward) and retrograde inverse (a melody is played backward but also inverted with respect to the pitch), are examples for such transformations. Since the Goldberg dataset used in this work reported here already includes all these mechanisms, including retrograde counterpoint, it is already part of the corpus of data used for discovery and future work could easily build on this (see also a preliminary discussion of this question surrounding the data depicted in Figure 8).

Future work could also explore other datasets. We already used a diverse set of unique protein sequences as defined via the NetSurf dataset; however, the musical data were limited to the corpus of concepts developed in the specific piece considered here (J.S. Bach's Goldberg Variations). It would be interesting to explore other musical datasets, different genres of music, or different pairings of data altogether. It is believed that there exist a large number of impactful applications of the concept and many follow-up studies that can take advantage of the convergent use of disparate knowledge bases including rigorous connecting principles between art and science.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Markus J. Buehler (mbuehler@MIT.EDU).

Materials availability

No materials were generated since this is theoretical and computational work.

Data and code availability

Data and codes are available on GitHub (https://github.com/lamm-mit/AttentionCrossTranslation and https://doi.org/10.5281/zenodo.7529195).

Dataset generation

The examples shown in Figures 5 and 6 are created based on synthetic datasets, where we use simple mathematical relations (delineated specifically in Figure 5) to construct known relationships between the two domains. The input and output sequences are of length max_{length} = 128 for the validation problems (Figures 5 and S2) and max_{length} = 256 for the protein-music translations.

The other results presented here are based on more complex datasets, as summarized here. The protein dataset is extracted from the NetSurf dataset of unique protein sequences.[36] We randomly draw ~15,000 sequence parts of 256 amino acid length from the entire dataset. We ensure that all sequence parts contain exactly 256 amino acids (albeit the model is set up to handle partial sequences of lesser length by masking out unused input tokens). To

Table 1. Transformer architecture details

Generator models	
Embedding dimension source data	256
Positional encoding embedding dimension	128
n_{depth}	3
n_{blocks}	3
n_{conv}	3 (kernel size 3, 6, 9)
Maximum sequence length	256
Latent dimension	512
No. of latents	64
No. of latent self-attention heads	16
No. of cross-attention heads	1
No. of input/output tokens (maximum no. of tokens in the protein data and music data)	57
Attention and feedforward dropout	0.15
Discriminator models	
Embedding dimension source data	32
Positional encoding embedding dimension	32
n_{conv}	4 (kernel size 3, 6, 9, 12)
No. of self-attention heads	8

The generator models have 80,125,207 parameters, and the discriminators 3,015,090 each.

Table 2. Loss function details and parameters

$\lambda_{CYCLE,AB}$	10
$\lambda_{CYCLE,BA}$	10
$\lambda_{ID,AB}$	1
$\lambda_{ID,BA}$	1
$\lambda_{GAN,AB}$	1
$\lambda_{GAN,BA}$	1
λ_{DISC}	0.5

represent protein data in a machine-readable manner, a tokenizer is trained to associate one-letter amino acid characters in FASTA format with tokens. The inverse of the resulting dictionary is used to convert predicted tokens into an amino acid sequence in one-letter amino acid codes.

The music dataset is extracted from J.S. Bach's Goldberg Variations.[37,38] Its choice is based on the systematic compositional technique used to generate the music, following mathematical principles and human choices. To do this, we translate all parts of the original composition to a MIDI file, then convert the MIDI file into tokenized sequences (the MIDI note number scheme is used as a tokenizer to relate the pitch of musical notes to a number (the MIDI note number is an integer between 0 and 127). We then randomly draw ~15,000 sequence segments of 256 note lengths from the entire dataset.

As described in Figures 3, 4, and 5 and the associated neural network design, embedding layers are used to process the discrete integer input into a higher-dimensional representation.

Figure 7A depicts a histogram of the music and protein data, respectfully, showing the distribution of the tokens. Music and protein data have a different number of total tokens, with a different distribution.

During training, we randomly select a batch of each of the two datasets at each step, for calculation of losses, gradients, and updating of the neural network parameters via backpropagation. This process is continued until the training is completed.

Design of the neural network architectures in the AttentionCrossTranslation model

All code is developed in PyTorch,[39] with the exception of the tokenizer that is designed and trained using TensorFlow Keras.[40] We use a cycle-consistent adversarial neural net architecture,[31–33] as shown in Figure 2. Generator A2B transforms data from modality A (e.g., music) into data in modality B (e.g., proteins) (note: the approach is illustrated for the music-protein example here but can be generally applied to any modality A, B). Conversely, generator B2A represents the opposite translation. Using adversarial training, A2B learns to generate sequences that resemble realistic proteins, and a discriminator D_B aims to distinguish between generated protein sequences $\tilde{x}_B = A2B(x_A)$ and real protein sequences x_B. At the same time, B2A learns to generate musical sequences that resemble real musical data, and the discriminator D_A aims to distinguish between generated music $\tilde{x}_A = B2A(x_B)$ and real music x_A.

With x_i and $\tilde{x}_i$ denoting the real field and approximate, predicted data, respectively,

$$A2B(x_A) = \tilde{x}_B \quad \text{(Equation 6)}$$

and

$$B2A(x_B) = \tilde{x}_A. \quad \text{(Equation 7)}$$

Classifiers D_A/D_B are trained to determine whether proteins or musical sequences are real or fake. In the adversarial training of this cycle-consistent neural network, the generator gets better and better at producing realistic data that can no longer be distinguished from real ones (and vice versa). This is achieved by training the model to feature cycle consistency in both forward and backward direction, that is:

$$B2A(A2B(x_A)) = \tilde{x}_A \approx x_A \quad \text{(Equation 8)}$$

and

$$A2B(B2A(x_B)) = \tilde{x}_B \approx x_B. \quad \text{(Equation 9)}$$

We also train the model to ensure identity loss, that is,

$$B2A(x_A) = \tilde{x}_A \approx x_A \quad \text{(Equation 10)}$$

and

$$A2B(x_B) = \tilde{x}_B \approx x_B. \quad \text{(Equation 11)}$$

Equations 10 and 11 signify that if real musical data are provided to generator B2A, the same musical data are produced (or equivalently other data modalities). Similarly, once protein data are provided to A2B, the same protein data are produced.

A set of λ parameters is introduced to weigh the relative contributions of the losses (discriminator loss, cycle consistency loss, and identity loss). We use L2 loss functions for the discriminator losses, and cross entropy loss functions for the cycle consistency loss and identity loss. The method can easily be adapted to use L2 loss functions for the data translations in cases where floating point data sequences may need to be used.

Table 1 shows overall details of the architecture of the four neural networks used in the adversarial attention architecture presented here, following the construction principles described in Figures 3 and 4. In both types of models, the embedded sequence is processed via a series of conv layers, each operating on a deeper realization of such convolution processing but concatenated to the output to yield a series of skip connections. We also add positional encoding, realized using a learnable encoding based on ordinal assignment of positional identities for each sequence element (1 … max_{length} +1) and encoded using a separate embedding layer.

Generator neural network

Following the schematic displayed in Figure 3, the generator neural network is constructed based on a series of perceiver attention blocks whereby the input data are processed into a latent space (see Table 1 for details) via

cross-attention, then processed using a series of self-attention steps. Each attention step consists of the attention operation and a conv layer, followed by layer normalization[41] and a feedforward MLP block; each repeated n_{depth} times. Skip connections are added between each attention step and the feedforward block. The perceiver architecture is arranged in a series of n_{blocks} to realize hierarchical processing.

Discriminator neural network

With the schematic displayed in Figure 4, the discriminator neural network is constructed based on the same input processing as the generator models. However, the output is processed in parallel using a self-attention block (each repeated n_{depth} times) and a series of conv layers that downsample the embedded input sequence. The results of the self-attention operation and the conv downsampling are both normalized using instance normalization[42] and then concatenated, yielding the final output that features a smaller dimension in the vein of a patch classifier.[43] The discriminator is designed to feature a combination of convolutional and attention mechanisms; to capture both patterns and hierarchical structuring but also very long-range relationships.[44] Since we are dealing with sequence data, all conv operations used are 1D convolutional layers.

Cycle consistency losses are weight by λ_{CYCLE}, and the identity loss by λ_{ID} (for details see Table 2). This strategy ensures that the model not only learns how to generate data that "look like" the required output, but specifically requires that the mapping in both forward and backward direction is satisfied. This is critical for the problem solved here.

Training and validation

We use an Adam optimizer[45] with $\beta = (0.5, 0.999)$. Training is done with a batch size of 4. The learning rate history (warmup steps and exponential decay) is visualized in Figure S2.

Protein folding

Protein folding is used to predict 3D structures of the predicted sequences. We use Alpha Fold 2[34] to conduct these experiments, using the Colab-Fold implementation.[46] We use the pdb70 template set, and MMseqs2 (UniRef and Environmental), using three cycles. The highest prediction is used for the analysis in this paper. The lDDT score is generally much higher for the original protein sequences (reflecting the fact that the proteins have been sourced from naturally occurring sequences). The designed protein sequences tend to have a low lDDT score due to the fact that there are no natural analogs; however, all samples folded well and arranged into well-packed and well-arranged structures.

Music and audio analysis

The audio files are rendered using Ableton Live,[47] whereas audible signals are created using a physical model of a harp-like instrument using the *Tension model* in Ableton Live, with subtle reverb added. The melodic spectrogram showed in Figure S7 is generated using Sonic Visualizer.[48]

MD analysis and protein data analysis and visualization

We use NAMD Scalable Molecular Dynamics with GPU support[49] with a suitable CHARMM protein force field to test the stability of the folded protein predictions. We use TIP3 explicit water solvent, add ions to achieve charge neutrality, and use the CHARMM 36 force field.[50] All preprocessing is done in visual molecular dynamics (VMD),[51] and visualization is performed in VMD and PyMol.[52] We choose an NPT ensemble with periodic boundary conditions, zero pressure, and T = 298 K for 5 ns, executed after 1,000 steps of energy minimization.

Software versions and hardware

We use Python 3.8.12, PyTorch 1.10 with CUDA (CUDA version 11.6), and an 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.patter.2023.100692.

ACKNOWLEDGMENTS

We acknowledge support from ARO (W911NF1920098 and W911NF-21-1-0178), NIH (U01EB014976 and 1R01AR077793), and ONR (N00014-19-1-2375 and N00014-20-1-2189). Additional support is provided by USDA (2021-69012-35978). Support from the MIT-IBM Watson AI Lab is acknowledged.

AUTHOR CONTRIBUTIONS

M.J.B. conceived the study; developed and trained the neural network and associated data analysis, including the protein models; and wrote the paper.

DECLARATION OF INTERESTS

The author declares no competing interests.

Received: November 29, 2022
Revised: January 2, 2023
Accepted: January 24, 2023
Published: February 14, 2023

REFERENCES

1. Wegst, U.G.K., Bai, H., Saiz, E., Tomsia, A.P., and Ritchie, R.O. (2015). Bioinspired structural materials. Nat. Mater. *14*, 23–36. https://doi.org/10.1038/nmat4089.
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*. https://doi.org/10.1016/j.biomaterials.2012.06.054.
3. Cranford, S.W., and Buehler, M.J. (2012). Biomateriomics (Springer Netherlands).
4. 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.
5. Giesa, T., Spivak, D.I., and Buehler, M.J. (2012). Category theory based solution for the building block replacement problem in materials design. Adv. Eng. Mater. *14*, 810–817. https://doi.org/10.1002/adem.201200109.
6. Schuijer, M. (2008). Analyzing Atonal Music : Pitch-Class Set Theory and its Contexts (University of Rochester Press).
7. Duncan, A. (1991). Combinatorial music theory. J. Audio Eng. Soc. *39*, 427–448.
8. Hudson, R. (2006). Regions and place: music, identity and place. Prog. Hum. Geogr. *30*, 626–634.
9. Benson, D.J. (2007). Music: A Mathematical Offering (Cambridge University Press).
10. Koelsch, S., Rohrmeier, M., Torrecuso, R., and Jentschke, S. (2013). Processing of hierarchical syntactic structure in music. Proc. Natl. Acad. Sci. USA *110*, 15443–15448. https://doi.org/10.1073/pnas.1300272110.
11. Browne, R., and Xenakis, I. (1973). Formalized music: thought and mathematics in composition. Notes *30*, 67. https://doi.org/10.2307/896037.
12. Higgins, K.M. (2012). The Music between Us: Is Music a Universal Language? (The University of Chicago Press).
13. Giesa, T., Spivak, D.I., and Buehler, M.J. (2011). Reoccurring patterns in hierarchical protein materials and music: the power of analogies. BioNanoScience *1*, 153–161. https://doi.org/10.1007/s12668-011-0022-5.
14. Wong, J.Y., McDonald, J., Taylor-Pinney, M., Spivak, D.I., Kaplan, D.L., and Buehler, M.J. (2012). Materials by design: merging proteins and music. Nano Today *7*, 488–495. https://doi.org/10.1016/j.nantod.2012.09.001.
15. Xenakis, I. (1971). Formalized Music Thought and Mathematics in Composition.

16. Widmer, G., Rocchesso, D., Välimäki, V., Erkut, C., Gouyon, F., Pressnitzer, D., Penttinen, H., Polotti, P., and Volpe, G. (2007). Sound and music computing: research trends and some key issues. J. New Music Res. *36*, 169–184. https://doi.org/10.1080/09298210701859222.

17. Tymoczko, D. (2011). A Geometry of Music : Harmony and Counterpoint in the Extended Common Practice (Oxford University Press).

18. Rohrmeier, M. (2011). Towards a generative syntax of tonal harmony. J. Math. Music *5*, 35–53. https://doi.org/10.1080/17459737.2011.573676.

19. Huang, C.-Z.A., Cooijmans, T., Roberts, A., Courville, A., and Eck, D. (2019). Counterpoint by convolution. In Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, pp. 211–218.

20. Pearce, R., and Zhang, Y. (2021). Deep learning techniques have significantly impacted protein structure prediction and protein design. Curr. Opin. Struct. Biol. *68*, 194–207. https://doi.org/10.1016/j.sbi.2021.01.007.

21. 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.

22. Bertaud, J., Qin, Z., and Buehler, M.J. (2009). Amino acid sequence dependence of nanoscale deformation mechanisms in alpha-helical protein filaments. J. Strain Anal. Eng. Des. *44*, 517–531. https://doi.org/10.1243/03093247JSA533.

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

24. 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.

25. Buehler, M., Keten, S., and Ackbarow, T. (2008). Theoretical and computational hierarchical nanomechanics of protein materials: deformation and fracture. Prog. Mater. Sci. *53*, 1101–1241. https://doi.org/10.1016/j.pmatsci.2008.06.002.

26. Yu, C.H., Qin, Z., Martin-Martinez, F.J., and Buehler, M.J. (2019). A self-consistent sonification method to translate amino acid sequences into musical compositions and application in protein design using AI. ACS Nano *13*, 7471–7482.

27. Qin, Z., and Buehler, M.J. (2019). Analysis of the vibrational and sound spectrum of over 100,000 protein structures and application in sonification. Extreme Mech. Lett. *29*, 100460. https://doi.org/10.1016/j.eml.2019.100460.

28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Neural information processing systems foundation), pp. 5999–6009.

29. Chaudhari, S., Mithal, V., Polatkan, G., and Ramanath, R. (2019). An attentive survey of attention models. J. ACM *37*.

30. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. Nature *538*, 471–476. https://doi.org/10.1038/nature20101.

31. Zhu, J.Y., Park, T., Isola, P., and Efros, A.A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision 2017, pp. 2242–2251. https://doi.org/10.1109/ICCV.2017.244.

32. Buehler, M.J. (2022). Prediction of atomic stress fields using cycle-consistent adversarial neural networks based on unpaired and unmatched sparse datasets. Mater. Adv. *3*, 6280–6290. https://doi.org/10.1039/D2MA00223J.

33. Buehler, E.L., and Buehler, M.J. (2022). End-to-end prediction of multimaterial stress fields and fracture patterns using cycle-consistent adversarial and transformer neural networks. Biomed. Eng. Adv. *4*, 100038. https://doi.org/10.1016/J.BEA.2022.100038.

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. 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.

36. Klausen, M.S., Jespersen, M.C., Nielsen, H., Jensen, K.K., Jurtz, V.I., Sønderby, C.K., Sommer, M.O.A., Winther, O., Nielsen, M., Petersen, B., and Marcatili, P. (2019). NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. Proteins *87*, 520–527. https://doi.org/10.1002/PROT.25674.

37. Kellner, H.A. (1981). The mathematical architecture of Bach's Goldberg variations. Engl. Harpsichord Mag. *2*, 183–189.

38. Booth, C. (2014). Bach's use of the single-note ornament in the Goldberg variations. Early Music *42*, 259–272. https://doi.org/10.1093/em/cau030.

39. Paszke, A., Gross, S., Bradbury, J., Lin, Z., Devito, Z., Massa, F., Steiner, B., Killeen, T., and Yang, E. (2019). PyTorch : An Imperative Style (High-Performance Deep Learning Library).

40. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., and Isard, M. (2016). Tensorflow: a system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265–283.

41. Ba, J.L., Kiros, J.R., and Hinton, G.E. (2016). Layer Normalization. https://doi.org/10.48550/arxiv.1607.06450.

42. Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. https://doi.org/10.48550/arxiv.1607.08022.

43. Hou, L., Samaras, D., Kurc, T.M., Gao, Y., Davis, J.E., and Saltz, J.H. (2015). Patch-based convolutional neural network for whole slide tissue image classification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016, pp. 2424–2433. https://doi.org/10.48550/arxiv.1504.07947.

44. 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.

45. Kingma, D.P., and Lei Ba, J. (2015). Adam: a method for stochastic optimization. Int. Conf. Learn. Representations *1–15*.

46. 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.

47. Ableton live digital audio workstation, https://www.ableton.com/en/live/.

48. Cannam, C., Landone, C., and Sandler, M. (2010). Sonic visualiser. In Proceedings of the international conference on Multimedia - MM '10 (ACM Press), p. 1467. https://doi.org/10.1145/1873951.1874248.

49. Phillips, J.C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R.D., Kalé, L., and Schulten, K. (2005). Scalable molecular dynamics with NAMD. J. Comput. Chem. *26*, 1781–1802. https://doi.org/10.1002/jcc.20289.

50. Best, R.B., Zhu, X., Shim, J., Lopes, P.E.M., Mittal, J., Feig, M., and MacKerell, A.D. (2012). Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone φ, ψ and side-chain χ_1 and χ_2 dihedral angles. J. Chem. Theory Comput. *8*, 3257–3273. https://doi.org/10.1021/ct300400x.

51. Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD: visual molecular dynamics. J. Mol. Graph. *14*, 33-8–27-8.

52. Schrödinger, L.L.C. (2015). The PyMOL Molecular Graphics System version 1.8.