Generating Novel Protein Sequences with Self-Supervised Learning

Sayantan Paul
Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Texas, USA
sayantan.paul@students.tamuk.edu

Wojciech M. Karłowski
Computational Biology
Adam Mickiewicz University
Poznań, Poland
wkarlowski@amu.edu.pl

Hani Z. Girgis
Electrical Engineering and Computer Science
Texas A&M University-Kingsville
Texas, USA
hani.girgis@tamuk.edu

***Abstract*—Generative artificial intelligence (AI) represents a major technological advancement, with applications spanning image synthesis to large language models capable of human-like interaction. Inspired by these developments, we designed a suite of generative deep learning models specifically for protein sequence generation. Leveraging the UniProt database, we evaluated several architectures—including autoencoders with dense and convolutional layers, as well as their variational counterparts. Among these, the convolutional variational autoencoder achieved the lowest reconstruction error. Notably, we found that the condensation level—defined as the size of the latent representation—had a strong impact on sequence identity: greater condensation reduced identity but increased sequence diversity. We identified 80% condensation as a critical threshold that enabled the generation of diverse yet biologically meaningful protein variants. These findings underscore the potential of generative AI to advance protein engineering.**

I. Introduction

Designing proteins with specific functionalities is one of the most intricate and transformative challenges in modern biology. Over the past five decades, significant advances in protein engineering have led to breakthroughs in the development of novel enzymes, therapeutic agents, and biosensors, revolutionizing industries from medicine to biotechnology. Proteins derive their functional characteristics from the intricate folding of their linear amino acid chains into complex secondary and tertiary structures. These structures dictate their unique activity and remain a critical focus for engineers aiming to tailor proteins for specific purposes. However, the acquisition of high-resolution structural data is notoriously expensive and labor-intensive, limiting the pace of innovation.

To address these challenges, we employ generative models that produce protein sequences resembling natural ones, without relying on three-dimensional structural information. Our approach uses similarity measures to guide the generation process, ensuring that the resulting sequences preserve key functional properties—such as stability and activity—comparable to their natural counterparts. By generating variable-length sequences, our models effectively capture patterns across both local and long-range residue interactions within the primary sequence.

II. Related Work

Self-supervised learning has revolutionized our ability to analyze and generate complex biological sequences by leveraging the rich, unlabeled data provided by high-throughput sequencing.

Early work in self-supervised sequence modeling adapted natural language processing (NLP) techniques to the biological domain. Methods based on the skip-gram framework—exemplified by dna2vec—treat biological k-mers as “words” and their surrounding tokens as contextual information [1]. Similarly, document embedding methods such as Seq2vec extend this concept to whole sequences, producing continuous vector representations that encode both local motifs and global sequence patterns [2]. These embedding strategies lay the groundwork for generating novel sequences via sampling from the learned latent spaces.

More recent approaches have shifted toward models that directly generate sequences. Variational Auto-Encoders (VAEs) [3], [4], for instance, develop continuous latent spaces from which biologically plausible sequences can be generated. Architectures such as BindVAE employ a Dirichlet formulation to discover latent “topics” (e.g., motif or binding signatures) from chromatin accessibility data, producing new sequences that preserve functional characteristics even when high-resolution structural information is absent [5].

Transformer architectures have significantly advanced sequence generation through self-supervised learning objectives. DNABERT, for instance, adapts masked language modeling to biological sequences—using k-mer-based tokenization—to learn deep contextual embeddings that capture both local interactions and long-range dependencies [6]. Complementary to this, contrastive learning approaches (such as those implemented in Self-GenomeNet) exploit the inherent symmetry of reverse complements in DNA. By applying a contrastive loss that brings matching subsequences closer together in embedding space while pushing non-matching pairs apart, these models produce robust representations that generalize across varying sequence lengths and regulatory contexts [7].

Additionally, GeneBERT extends the BERT architecture by incorporating multiple sources of information—such as k-mer, positional, and segment embeddings—to produce unified representations applicable to the design of regulatory genomic elements [8]. Transformer-based models have demonstrated strong performance on a range of tasks, including mutation effect prediction and the generation of high-fitness protein variants.

While originally developed for raw audio generation, the WaveNet architecture provides valuable insights applicable to biological sequence generation. WaveNet's use of dilated causal convolutions allows it to capture long-range temporal dependencies without requiring explicit recurrent connections [9]. Such architectures can be adapted to biological sequences—where capturing dependencies over long ranges is critical for preserving structural motifs and functional domains—to generate sequences that are both coherent and biologically realistic. WaveNet's ability to model complex data distributions suggests promising avenues for generating biological sequences within a self-supervised learning framework.

Recent work on protein sequence modeling [10] provides a comprehensive overview of generative approaches in this domain. Their review discusses how methods like VAEs, generative adversarial networks, and autoregressive models have been applied to generate and optimize protein sequences. The authors highlight experimentally validated cases, such as luciferase generation using VAEs, malate dehydrogenase design via generative adversarial networks, and single-domain antibody creation with autoregressive models. These efforts not only expand sequence space into new structural domains but also optimize functional properties, as demonstrated in applications like signal peptide design for industrial enzymes and metal-binding motif generation for metalloproteins. In sum, these advances illustrate the transformative potential of deep generative methods in understanding protein fitness landscapes and designing sequences with tailored functionalities.

Collectively, these self-supervised frameworks—from embedding-based methods to advanced generative architectures—offer a versatile toolkit for biological sequence generation. They allow researchers to traverse vast sequence spaces to identify novel proteins or nucleic acid sequences with tailored functionalities, all while bypassing the need for costly experimental structure determination.

III. Methods

A. Overview

We devised a comprehensive comparison methodology aimed at generating novel protein sequences, beginning with the design and evaluation of foundational models built from basic neural network layers. Initial experiments focused on auto-encoder architectures, exploring variations with dense layers, convolutional layers, and recurrent layers. To enhance model performance, we systematically optimized key parameters, such as the number of neurons, filter counts, kernel sizes, and the extent of condensation in the representation layer. Additionally, we designed two Variational Auto-Encoders (VAEs) specifically for their ability to generate multiple variants of the same sequence. Overall, we experimented with four generative networks: (i) dense auto-encoder, (ii) convolutional auto-encoder, (iii) dense VAE, and (iv) convolutional VAE.

B. Dataset

Our dataset includes 571,864 protein sequences from UniProt [11] and millions of bacterial Open Reading Frames (ORFs). UniProt sequences shorter than 50 or longer than 1,024 amino acids were excluded, yielding 540,957 sequences. To obtain bacterial ORFs, we downloaded 2,421 representative bacterial genomes from the Genome Taxonomy Database (GTDB) [12], using 2,000 genomes for model development and validation, and setting aside 421 for final blind testing, though these were not utilized in the current study as this specific work is still in progress. We extracted ORFs using the `getorf` program from EMBOSS [13], then aligned them to UniProt proteins using BLAST [14]. ORFs with at least 75% alignment coverage were labeled as confirmed, while those with at most 50% coverage were labeled unconfirmed. This process yielded 3,634,984 confirmed ORFs; we selected an equal number of unconfirmed ORFs, matching the length distribution of the confirmed set. We combined confirmed ORFs and UniProt sequences to form a functional (positive) set and partitioned it into training (70%; 757,338 sequences), validation (20%; 216,382), and testing (10%; 108,194).

C. Sequence Representation

Each amino acid is assigned a unique index and then represented by a 3-dimensional vector using an already-trained embedding layer — a specialized type of neural network layer that learns how to map categorical values to real-valued vectors while capturing similarities among them. Amino acids with similar properties are represented by vectors that are close together in this 3-dimensional space, while dissimilar ones are mapped farther apart. The embedding vectors are learned by a neural network trained to classify functional protein sequences from non-functional ones (to be described elsewhere). Next, we detail the various neural network architectures employed for generating novel protein sequences.

D. Concept of masking padded inputs

To ensure the model effectively learned amino acid relationships despite varying protein sequence lengths, input sequences were padded to a fixed length, with 0 as the padding token and its embedding vector explicitly set to 0. A masking layer followed the input layer, marking padded vectors as False and actual tokens as True in a boolean format. The resulting mask tensor was applied in a custom layer, multiplying it with the input tensor to preserve zeroed padded values across subsequent layers, preventing their influence on training. This technique improved training efficiency, yielding lower loss values and enhanced similarity metrics between predicted and original sequences.

E. Stacked Convolutional Auto-encoder

In Figure 1(a), we outline the architecture of the stacked convolutional auto-encoder. This model architecture begins with a masking layer applied to the input to handle padded tokens. This is followed by two stacked 1D convolutional layers in the encoder, where the second layer has twice the number of filters and a stride of 2 for downsampling. A 1D max-pooling layer processes the mask to match the reduced sequence length, and a custom masking layer ensures padded positions remain zeroed. The next layer is a dense layer that acts as the condensation layer to produce a latent representation of the sequence. In the decoder, a Conv1DTranspose layer upsamples the encoded features back to the original sequence length. The output is then passed through a flatten layer, a dense layer for aggregation, and a reshape layer to restore the sequence to its final dimensions. A final masking step is applied to maintain exclusion of padded inputs during reconstruction.

F. Stacked Dense Auto-encoder

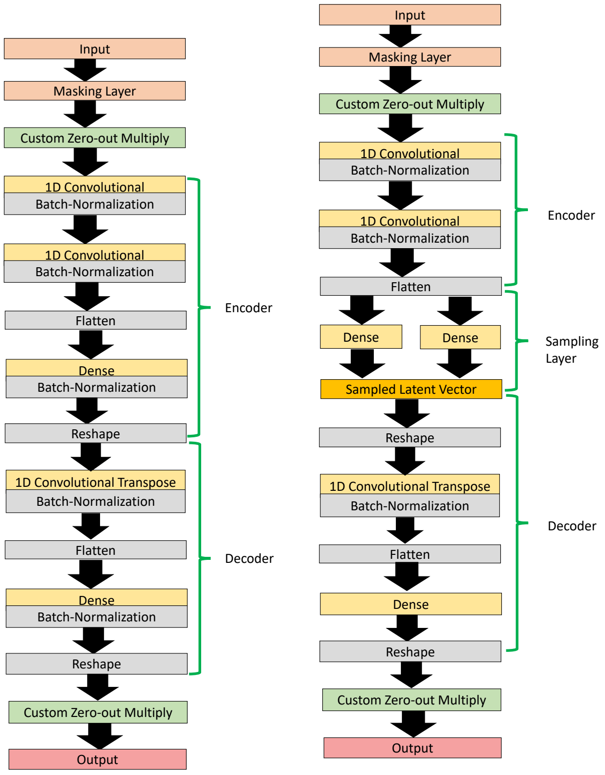
We developed a neural network featuring multiple hidden layers, known as stacked auto-encoders. The input data consists of protein sequences represented as time steps, with 3-dimensional embeddings as features. A masking layer was applied immediately after the input layer to differentiate padded inputs. The mask tensor produced by this layer was then utilized in a custom layer to multiply the input tensor by the mask tensor. This multiplication was performed both before feeding the inputs to the first dense layer in the encoder. The next layer is a dense layer that acts as the condensation layer to produce a latent representation of the sequence. The auto-encoder was designed with a symmetrical architecture, where the first dense layer in the encoder and the last dense layer in the decoder consistently maintained a neuron count equal to three times the maximum sequence length. Finally, the output of the final dense layer is reshaped to match the shape of the input sequence.

G. Stacked Recurrent Auto-encoder

The recurrent auto-encoder architecture consists of stacked Gated Recurrent Unit (GRU) layers, each followed by a Batch Normalization (BN) layer to stabilize and accelerate training. A custom masking layer is applied before the first GRU layer in the encoder to zero out padded positions in the input. In the decoder, after reconstruction, a reshape layer is applied, followed by another custom masking layer to ensure padded inputs remain excluded. This recurrent architecture is optimized for sequential protein data, capturing long-range dependencies efficiently but becomes computationally intensive for sequences longer than 100 amino acids.

H. Variational Auto-encoder with Convolutional layers

In Figure 1(b), we outline the architecture of our variational auto-encoder, which incorporates convolutional layers. The process begins with an input masking layer that flags padded positions within the input data. Following this, an extract mask layer retrieves a mask indicating the valid timesteps. This mask is then applied by a masked zeroing layer, which sets padded positions to zero, and an assert masked zeros layer confirms the integrity of this masking. The data then passes through two convolutional blocks. The first convolutional block consists of a 1D convolutional layer with a kernel size of 3 or 5, followed by batch normalization and ReLU activation. The second convolutional block includes another 1D convolutional layer, this time with twice the number of filters and a stride of 2 for downsampling. This block also incorporates max pooling with a pool size of 2, which helps downsample the mask to align with the output of this layer. Throughout these blocks, masked zeroing layers and assert masked zeros layers are used to maintain masking enforcement. The output from these convolutional blocks is then flattened and fed into two parallel dense layers. These layers are responsible for producing the mean and log-variance vectors that define the latent space. The latent space serves as a compressed representation where key features and patterns of the high-dimensional data are encoded. A sampling layer then draws latent vectors from this space. These vectors are reshaped and passed through a 1D transpose convolutional block for upsampling. Finally, the result is flattened, passed through a dense layer, and then reshaped to match the dimensions of the original output.



(a) Convolutional Auto-encoder Network

(b) Convolutional Variational Auto-encoder Network

Fig. 1. Comparison of two network architectures.

I. Variational Auto-encoder with Dense layers

We designed a variational auto-encoder (VAE) using dense layers. We applied an input masking layer to exclude padded values right after the input. Then, we incorporated custom layers—an extract mask layer, a masked zeroing layer, and an assert masked zeros layer—to ensure the integrity of these masked regions throughout the entire process. For the encoder, we flattened the input and passed it through dense layers with ReLU activation and batch normalization. The coding layer then computed both the mean and log variance of the latent space, enabling variational sampling. In the decoder, we reconstructed the data from latent space representations, reshaping the output to its original dimensions. By combining the encoder and decoder within the VAE framework, we utilized a sampling layer for latent variable generation and trained the model using the Adam optimizer.

IV. Results & Discussion

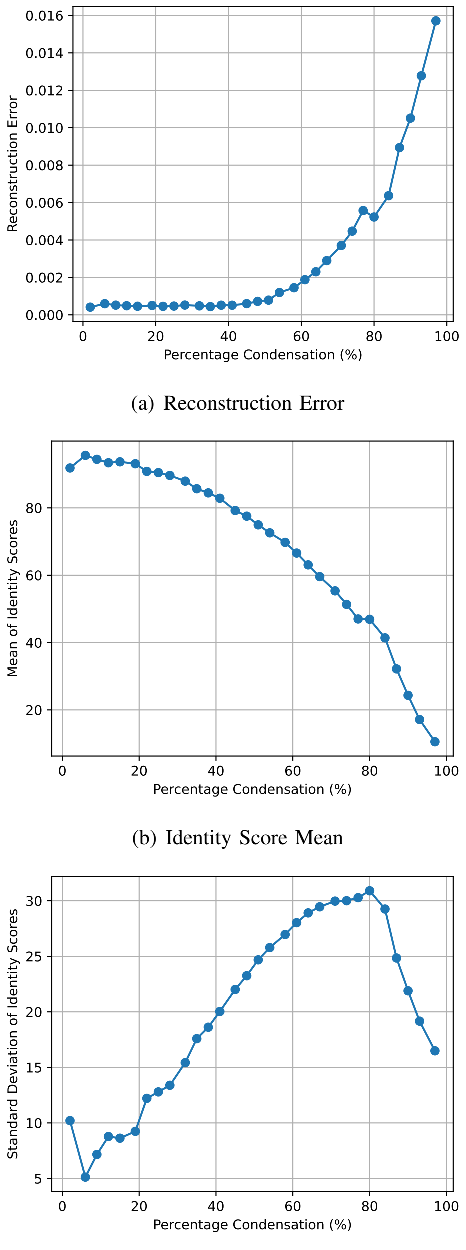
We trained models designed for sequence lengths ranging from 50 to 1024 amino acids, aiming to assess their performance and practical feasibility. During this process, we identified limitations on our computers with standard CPU and GPU; such limitations rendered recurrent models ineffective for sequences exceeding 50 amino acids. As a result, we focused on four alternative architectures: (i) an auto-encoder with dense layers, (ii) an auto-encoder with convolutional layers, (iii) a variational auto-encoder with dense layers, and (iv) a variational auto-encoder with convolutional layers.

A. Experiment to determine better model

In the first experiment, 100 sequences were sampled from the validation dataset. The range of lengths of these sequences varied from 50 to 512. These sequences were used as inputs to the four trained models, and the loss metric for each model was recorded. The loss metric is the mean absolute error; recall that each amino acid is represented by a vector of three dimensions. The metric measures the average difference between a predicted vector in the generated sequence and the true vector at the same position in the original sequence. The convolutional Variational Auto-Encoder (VAE) resulted in the lowest validation loss (0.0015), followed by the dense VAE (0.0021), the convolutional auto-encoder (0.0022), and finally the dense auto-encoder (0.0031). We found that variational models outperform regular autoencoders, and convolutional architectures work better than dense ones. A key factor is the condensation level—the size of the hidden layer that captures the sequence representation. This can be seen as the model's memory: the smaller the size, the lower the memory capacity. Our goal is to compress protein sequences as much as possible while preserving their essential patterns, ultimately enabling the generation of meaningful natural variants.

B. Experiment to study the effects of condensation levels

We conducted a second experiment to examine the effects of condensation levels by calculating (i) the average reconstruction error, (ii) the average identity, and (iii) the standard deviation across multiple condensation levels. We expanded the validation set to 1,000 sequences, with lengths ranging from 50 to 1,024. For each validation sequence, the network generated a variant. This experiment focused on convolutional VAEs, as previous results showed they outperform dense models in reconstruction while also enabling the generation of diverse variants from the same input protein sequence. We then computed the identity score between the original and regenerated sequences. Finally, we calculated the average and standard deviation across all 1,000 sequences.



(a) Reconstruction Error

(b) Identity Score Mean

(c) Identity Score STD

Fig. 2. An experiment to study the effects of different condensation levels. (a) The loss metric for the convolutional variational auto-encoder. (b) The average identity scores for the generated sequences. (c) The standard deviation depicting variability of the generated sequences.

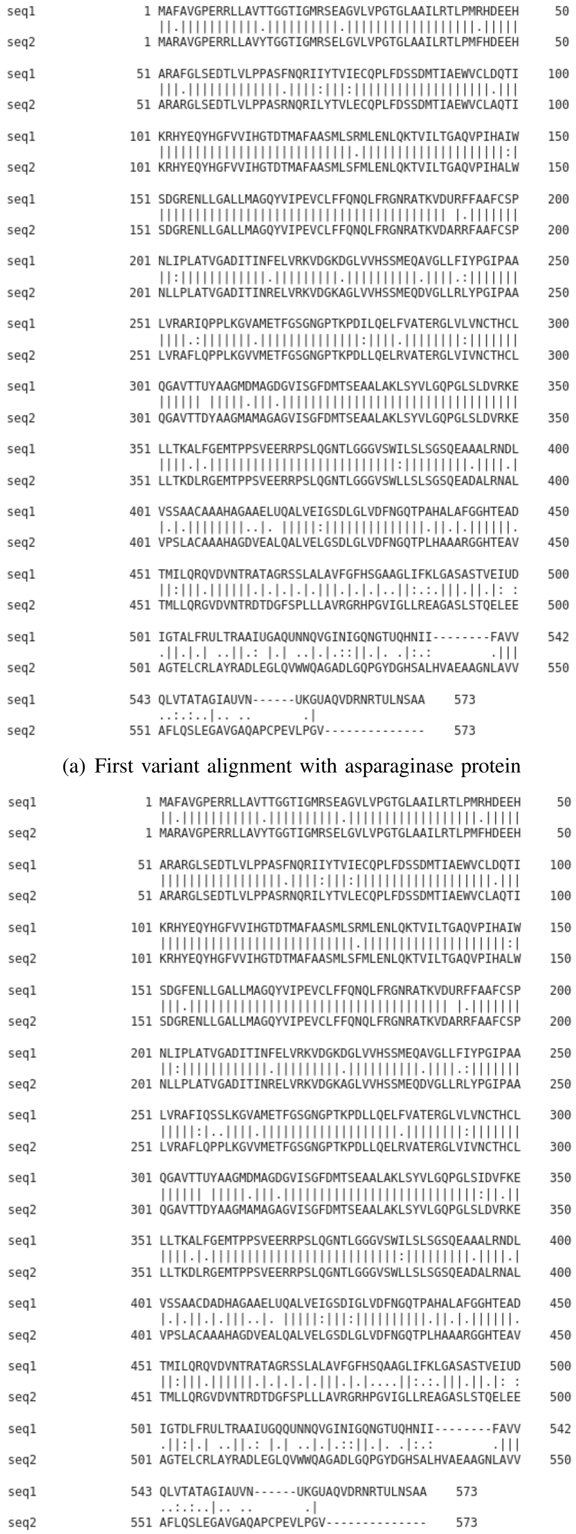
The results from the second experiment are given in Figure 2. When the size of the condensation layer increased, the memory capacity of the network also increased, and the reconstruction error decreased (Figure 2(a)). As the condensation level decreased (i.e., the layer size increased), the mean identity score generally exhibited an upward trend, as expected (Figure 2(b)). For example, when the size of the condensation layer was 1500 (50% condensation), the mean identity score of the generated sequences was about 74%. When the size of the layer was increased to 1800 (60% condensation), the mean identity score was about 82%. Observing the associated standard deviations and mean identity scores, we find that variability increases as condensation increases, up to a certain threshold. Notably, at 80% condensation, where the mean identity score is 46%, the model began to generate meaningful sequence variants. These results suggest a threshold below which the model tends to generate less meaningful outputs, and beyond which it begins to produce coherent, functional variants, underscoring the critical role of condensation in enabling the generation of novel and biologically relevant protein sequences.

C. Case study using natural protein

In this case study, we investigated the generation of protein variants for Q86U10 (ASPG) which is a multifunctional human enzyme that exhibits asparaginase activity alongside lipid-modifying functions, making it a rare native human candidate linked to therapeutic asparaginase functions. The reviewed Uniprot entry for Q86U10 (ASPG) is available on the Uniprot website, confirming its sequence length of 573 amino acids.

We performed this case study with the convolutional variational auto-encoder comprising 1500 neurons to generate variants of the natural Q86U10 (ASPG) sequence. We then conducted pairwise sequence alignment to evaluate the similarity between the original protein and its reconstructed variants. Our analysis revealed the following identity scores: Variant 1 exhibited an identity score of 77.2% compared to the native asparaginase sequence. Variant 2 displayed an identity score of 76.3% relative to the original protein. These findings demonstrate the ability of the convolutional variational auto-encoder to generate meaningful sequence variations while retaining a significant degree of structural similarity to the native protein. The pairwise alignment of this natural protein with two variants reconstructed by the convolutional variational auto-encoder is shown in (Figure 3)

To evaluate the biological plausibility and structural integrity of the protein variants generated by our model, we conducted a domain-level comparison between the original (reference) protein sequence and a set of 25 synthetically generated variants. The reference sequence corresponds to the human asparaginase enzyme with a sequence length of 573 amino acids, retrieved from the UniProtKB database. All variants generated showed an identity score between 76–77%. Domain annotation was performed using the hmmscan utility from the HMMER3 package (v3.3) [15], employing the Pfam-A HMM library (current release) [16]. This allowed us to identify and compare conserved protein domains present in both the reference and the synthetic sequences. We parsed the resulting domain tables to determine the presence, absence, or alteration of known functional domains in each variant relative to the reference. For each variant, we computed the percentage



```
seq1          1 MAFAVGPERRLLAVTTGGTIGMRSEAGVLVPGTGLAAILRTLPMRHDEEH     50
                ||.|||||||||||.||||||||||.||||||||||||||||||.|||||
seq2          1 MARAVGPERRLLAVYTGGTIGMRSELGVLVPGTGLAAILRTLPMFHDEEH     50

seq1         51 ARAFGLSEDTLVLPPASFNQRIIYTVIECQPLFDSSDMTIAEWVCLDQTI    100
                |||.|||||||||||||.||||:|||:|||||||||||||||||||.|||
seq2         51 ARARGLSEDTLVLPPASRNQRILYTVLECQPLFDSSDMTIAEWVCLAQTI    100

seq1        101 KRHYEQYHGFVVIHGTDTMAFAASMLSRMLENLQKTVILTGAQVPIHAIW    150
                |||||||||||||||||||||||||||.||||||||||||||||||||:|
seq2        101 KRHYEQYHGFVVIHGTDTMAFAASMLSFMLENLQKTVILTGAQVPIHALW    150

seq1        151 SDGRENLLGALLMAGQYVIPEVCLFFQNQLFRGNRATKVDURFFAAFCSP    200
                |||||||||||||||||||||||||||||||||||||||| |.|||||||
seq2        151 SDGRENLLGALLMAGQYVIPEVCLFFQNQLFRGNRATKVDARRFAAFCSP    200

seq1        201 NLIPLATVGADITINFELVRKVDGKDGLVVHSSMEQAVGLLFIYPGIPAA    250
                ||:||||||||||||.|||||||||.||||||||||.||||.:|||||||
seq2        201 NLLPLATVGADITINRELVRKVDGKAGLVVHSSMEQDVGLLRLYPGIPAA    250

seq1        251 LVRARIQPPLKGVAMETFGSGNGPTKPDILQELFVATERGLVLVNCTHCL    300
                ||||.:|||||||.||||||||||||||:||||.||||||||:|||||||
seq2        251 LVRAFLQPPLKGVVMETFGSGNGPTKPDLLQELRVATERGLVIVNCTHCL    300

seq1        301 QGAVTTUYAAGMDMAGDGVISGFDMTSEAALAKLSYVLGQPGLSLDVRKE    350
                |||||| |||||.|||.|||||||||||||||||||||||||||||||||
seq2        301 QGAVTTDYAAGMAMAGAGVISGFDMTSEAALAKLSYVLGQPGLSLDVRKE    350

seq1        351 LLTKALFGEMTPPSVEERRPSLQGNTLGGGVSWILSLSGSQEAAALRNDL    400
                ||||.|.||||||||||||||||||||||||||:|||||||||.||||.|
seq2        351 LLTKDLRGEMTPPSVEERRPSLQGNTLGGGVSWLLSLSGSQEADALRNAL    400

seq1        401 VSSAACAAAHAGAAELUQALVEIGSDLGLVDFNGQTPAHALAFGGHTEAD    450
                |.|.||||||||..|. |||||:||||||||||||||.||.|.||||||.
seq2        401 VPSLACAAAHAGDVEALQALVELGSDLGLVDFNGQTPLHAAARGGHTEAV    450

seq1        451 TMILQRQVDVNTRATAGRSSLALAVFGFHSGAAGLIFKLGASASTVEIUD    500
                ||:|||.||||||.|.|.|.|.|||.|.|.|..||:.:.|||.||.|: :
seq2        451 TMLLQRGVDVNTRDTDGFSPLLLAVRGRHPGVIGLLREAGASLSTQELEE    500

seq1        501 IGTALFRULTRAAIUGAQUNNQVGINIGQNGTUQHNII--------FAVV    542
                .||.|.| ..||.: |.| ..|.|.::||.|. .|:.:        .|||
seq2        501 AGTELCRLAYRADLEGLQVWWQAGADLGQPGYDGHSALHVAEAAGNLAVV    550

seq1        543 QLVTATAGIAUVN------UKGUAQVDRNRTULNSAA    573
                ..:.:..|.. ..       .|
seq2        551 AFLQSLEGAVGAQAPCPEVLPGV--------------    573
```

(a) First variant alignment with asparaginase protein

```
seq1          1 MAFAVGPERRLLAVTTGGTIGMRSEAGVLVPGTGLAAILRTLPMRHDEEH     50
                ||.|||||||||||.||||||||||.||||||||||||||||||.|||||
seq2          1 MARAVGPERRLLAVYTGGTIGMRSELGVLVPGTGLAAILRTLPMFHDEEH     50

seq1         51 ARARGLSEDTLVLPPASFNQRIIYTVIECQPLFDSSDMTIAEWVCLDQTI    100
                |||||||||||||||||.||||:|||:|||||||||||||||||||.|||
seq2         51 ARARGLSEDTLVLPPASRNQRILYTVLECQPLFDSSDMTIAEWVCLAQTI    100

seq1        101 KRHYEQYHGFVVIHGTDTMAFAASMLSRMLENLQKTVILTGAQVPIHAIW    150
                |||||||||||||||||||||||||||.||||||||||||||||||||:|
seq2        101 KRHYEQYHGFVVIHGTDTMAFAASMLSFMLENLQKTVILTGAQVPIHALW    150

seq1        151 SDGFENLLGALLMAGQYVIPEVCLFFQNQLFRGNRATKVDURFFAAFCSP    200
                |||.|||||||||||||||||||||||||||||||||||| |.|||||||
seq2        151 SDGRENLLGALLMAGQYVIPEVCLFFQNQLFRGNRATKVDARRFAAFCSP    200

seq1        201 NLIPLATVGADITINFELVRKVDGKDGLVVHSSMEQAVGLLFIYPGIPAA    250
                ||:||||||||||||.|||||||||.||||||||||.||||.:|||||||
seq2        201 NLLPLATVGADITINRELVRKVDGKAGLVVHSSMEQDVGLLRLYPGIPAA    250

seq1        251 LVRAFIQSSLKGVAMETFGSGNGPTKPDLLQELFVATERGLVLVNCTHCL    300
                |||||:|..||||.|||||||||||||||||||.||||||||:|||||||
seq2        251 LVRAFLQPPLKGVVMETFGSGNGPTKPDLLQELRVATERGLVIVNCTHCL    300

seq1        301 QGAVTTUYAAGMDMAGDGVISGFDMTSEAALAKLSYVLGQPGLSIDVFKE    350
                |||||| |||||.|||.|||||||||||||||||||||||||||:||.||
seq2        301 QGAVTTDYAAGMAMAGAGVISGFDMTSEAALAKLSYVLGQPGLSLDVRKE    350

seq1        351 LLTKALFGEMTPPSVEERRPSLQGNTLGGGVSWILSLSGSQEAAALRNDL    400
                ||||.|.||||||||||||||||||||||||||:|||||||||.||||.|
seq2        351 LLTKDLRGEMTPPSVEERRPSLQGNTLGGGVSWLLSLSGSQEADALRNAL    400

seq1        401 VSSAACDADHAGAAELUQALVEIGSDIGLVDFNGQTPAHALAFGGHTEAD    450
                |.|.||.|.|||..|. |||||:|||:||||||||||.||.|.||||||.
seq2        401 VPSLACAAAHAGDVEALQALVELGSDLGLVDFNGQTPLHAAARGGHTEAV    450

seq1        451 TMILQRQVDVNTRATAGRSSLALAVFGFHSQAAGLIFKLGASASTVEIUD    500
                ||:|||.||||||.|.|.|.|.|||.|.|....||:.:.|||.||.|: :
seq2        451 TMLLQRGVDVNTRDTDGFSPLLLAVRGRHPGVIGLLREAGASLSTQELEE    500

seq1        501 IGTDLFRULTRAAIUGQQUNNQVGINIGQNGTUQHNII--------FAVV    542
                .||:|.| ..||.: |.| ..|.|.::||.|. .|:.:        .|||
seq2        501 AGTELCRLAYRADLEGLQVWWQAGADLGQPGYDGHSALHVAEAAGNLAVV    550

seq1        543 QLVTATAGIAUVN------UKGUAQVDRNRTULNSAA    573
                ..:.:..|.. ..       .|
seq2        551 AFLQSLEGAVGAQAPCPEVLPGV--------------    573
```

(b) Second variant alignment with asparaginase protein

Fig. 3. A case study showcasing two novel variants of the asparaginase protein. We used the convolutional variational auto-encoder with a condensation layer of 1500 neurons (75% condensation). In both panels seq1 indicates the variant protein sequence and seq2 indicates the original asparaginase protein sequence. (a) The first variant has an identity score of 77.2% with the original protein. (b) The second variant has an identity score of 76.3% with the original protein.

similarity based on shared domain content with the original sequence and documented any domains that were missing or newly introduced.

To assess the conservation of functional and structural features in model-generated variants of the human ASPG enzyme (Q86U10), we compared their domain architectures to the reference sequence, which comprises eight domains including multiple ankyrin repeats (Ank, Ank 2–5, Ank KRIT1) and two enzymatic domains (Asparaginase, Asparaginase C) For checking domain conservation we ran scans using Hmmer [15] for both the original protein sequence and its 25 variants. Among the 25 predicted variants analyzed, most retained core functional domains, particularly Asparaginase and Asparaginase C, across all sequences. The majority (16 out of 25) showed 87.5% domain similarity with the reference, typically lacking only the Ank KRIT1 domain. A smaller group (7 variants) had 75.0% similarity, missing both Ank 3 and Ank KRIT1, while a few (3 variants) exhibited the lowest similarity at 62.5%, due to the additional loss of Ank 5. Notably, one variant included an extra DUF2064 domain not found in the reference, suggesting possible structural divergence. These results indicate that the model can generate variants preserving the enzymatic identity of the asparaginase enzyme.

V. Conclusion

We implemented and evaluated two autoencoders and two variational autoencoders (VAEs) for generating variants of natural protein sequences, experimenting with dense, convolutional, and recurrent layers. Overall, our findings demonstrate that VAEs offer a clear advantage over traditional autoencoders for generating biologically meaningful variants of natural protein sequences. Their ability to learn a continuous latent space enables the capture of essential biological patterns while allowing for diversity in sequence generation. Further, convolutional layers resulted in slightly better performance than their dense counterparts. We observed a strong positive correlation between the size of the condensation layer (i.e., the size of the latent representation) and the average sequence identity, indicating that larger latent spaces lead to improved reconstruction fidelity. However, this comes at the cost of reduced sequence novelty. Building on this, we identified a critical threshold in latent space size beyond which the model transitions from generating diverse but noisy sequences to producing coherent and functionally meaningful variants, suggesting effective utilization of the latent space for biologically relevant generation. A detailed case study on a multifunctional human enzyme further demonstrated the model's ability to generate protein variants that not only maintained high sequence identity but also preserved key functional domains, reinforcing the biological plausibility of the generated sequences. Together, these results underscore the promise of self-supervised generative models — particularly convolutional VAEs — in advancing protein engineering by enabling controlled generation of novel, functional sequences.

VI. Acknowledgment

Hani Z. Girgis is currently a Fulbright Scholar in residence at Adam Mickiewicz University, Poznań, Poland, supported by award number US/2024/22/SC.

References

[1] P. Ng, "dna2vec: Consistent vector representations of variable-length k-mers," 2017. [Online]. Available: https://arxiv.org/abs/1701.06279

[2] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, "Distributed representations for biological sequence analysis," 2016. [Online]. Available: https://arxiv.org/abs/1608.05949

[3] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2022.

[4] F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island, NY, USA: Manning Publications Co., 2021.

[5] M. Kshirsagar, H. Yuan, J. L. Ferres, and C. Leslie, "BindVAE: Dirichlet variational autoencoders for de novo motif discovery from accessible chromatin," *Genome Biology*, vol. 23, no. 1, p. 174, 2022. [Online]. Available: https://doi.org/10.1186/s13059-022-02723-w

[6] Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri, "Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome," *Bioinformatics*, vol. 37, no. 15, pp. 2112–2120, 02 2021.

[7] H. A. Gündüz, M. Binder, X.-Y. To, R. Mreches, B. Bischl, A. C. McHardy, P. C. Münch, and M. Rezaei, "A self-supervised deep learning method for data-efficient training in genomics," *Communications Biology*, vol. 6, no. 1, p. 928, 2023.

[8] S. Mo, X. Fu, C. Hong, Y. Chen, Y. Zheng, X. Tang, Z. Shen, E. P. Xing, and Y. Lan, "Multi-modal self-supervised pre-training for regulatory genome across cell types," 2021. [Online]. Available: https://arxiv.org/abs/2110.05231

[9] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[10] Z. Wu, K. E. Johnston, F. H. Arnold, and K. K. Yang, "Protein sequence design with deep generative models," *Current Opinion in Chemical Biology*, vol. 65, pp. 18–27, 2021, mechanistic Biology * Machine Learning in Chemical Biology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136759312100051X

[11] T. U. Consortium, "Uniprot: the universal protein knowledgebase in 2025," *Nucleic Acids Research*, vol. 53, no. D1, pp. D609–D617, 11 2024.

[12] D. H. Parks, M. Chuvochina, P.-A. Chaumeil, C. Rinke, A. J. Mussig, and P. Hugenholtz, "A complete domain-to-species taxonomy for bacteria and archaea," *Nature Biotechnology*, vol. 38, no. 9, pp. 1079–1086, 2020.

[13] P. Rice, I. Longden, and A. Bleasby, "EMBOSS: the European Molecular Biology Open Software Suite," *Trends in Genetics*, vol. 16, no. 6, pp. 276–277, Jun. 2000.

[14] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990.

[15] S. R. Eddy, "Accelerated profile hmm searches," *PLOS Computational Biology*, vol. 7, 10 2011. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1002195

[16] T. Paysan-Lafosse, A. Andreeva, M. Blum, S. R. Chuguransky, T. Grego, B. L. Pinto, G. A. Salazar, M. L. Bileschi, F. Llinares-López, L. Meng-Papaxanthos, L. J. Colwell, N. V. Grishin, R. D. Schaeffer, D. Clementel, S. C. E. Tosatto, E. Sonnhammer, V. Wood, and A. Bateman, "The pfam protein families database: embracing ai/ml," *Nucleic Acids Research*, vol. 53, no. D1, pp. D523–D534, 11 2024.