BioChem

MDPI

Article

De Novo Drug Design Using Artificial Intelligence Applied on SARS-CoV-2 Viral Proteins ASYNT-GAN

Ivan Jacobs [1,*] and Manolis Maragoudakis [2]

[1] AI4U, 7321 Steinsel, Luxembourg
[2] Department of Informatics, Ionian University, 49100 Corfu, Greece; mmarag@ionio.gr
* Correspondence: ivan.jacobs@ai4u.ai

Abstract: Computer-assisted de novo design of natural product mimetics offers a viable strategy to reduce synthetic efforts and obtain natural-product-inspired bioactive small molecules, but suffers from several limitations. Deep learning techniques can help address these shortcomings. We propose the generation of synthetic molecule structures that optimizes the binding affinity to a target. To achieve this, we leverage important advancements in deep learning. Our approach generalizes to systems beyond the source system and achieves the generation of complete structures that optimize the binding to a target unseen during training. Translating the input sub-systems into the latent space permits the ability to search for similar structures, and the sampling from the latent space for generation.

Keywords: de novo drug design; synthetic molecule structure generation; deep learning

Citation: Jacobs, I.; Maragoudakis, M. De Novo Drug Design Using Artificial Intelligence Applied on SARS-CoV-2 Viral Proteins ASYNT-GAN. *BioChem* **2021**, *1*, 36–48. https://dx.doi.org/10.3390/biochem1010004

Received: 29 September 2020
Accepted: 4 November 2020
Published: 5 April 2021

1. Introduction

An outbreak of the novel coronavirus, SARS-CoV-2, has caused worldwide social and economic disruption. The scientific community has limited knowledge of the molecular details of the infection. Currently, there is a lack of common adaption of antiviral drugs, and are no vaccines for prevention.

The time and effort to create and market a drug or vaccine that can treat a certain infection can span over decades and millions of investments. Throughout the process of drug discovery, one of the main challenges is to identify a molecular structure that can attach itself to a target. The quality of the binding has a direct influence on the side effects and effectiveness of the treatment. Multiple approaches exist to find or create these structures. On successful structure identification, the compounds' structure is further built around it.

Computer-assisted de novo design permits to reduce efforts and obtain natural-product-inspired bioactive molecules. Three types of in silico drug-target interaction (DTI) prediction methods have been proposed in the literature: molecular docking [1], similarity-based [2], and deep learning-based models [3]. Molecular docking is a simulation-based on predefined human rules aiming to optimize the conformation of ligand and the target protein. The currently applied docking methods have shown multiple limitations, in particular unsatisfactory scoring of biological activities. Investigations have shown that the performance of docking approaches is dependent on the type of receptors and performs better on hydrophobic vs. hydrophilic pockets [4]. Additionally, in practice, the binding pose rarely has enough information to reproduce affinities. For this reason, more computationally expensive methods, such as molecular mechanics generalized born and surface area continuum solvation (MM/GBSA), need to be applied to re-score compounds during virtual screening [5].

Proposed machine learning DTI approaches as, KronRLS [6] and SimBoost [7], concentrate on binding affinity prediction. Where the compound and protein are transformed into their Simplified Molecular Input Line Entry System (SMILES) and sequence representation, respectively, and used to predict the probability of high-affinity score. By applying this transformation, the ligands and proteins are considered in their 1D representations by the model. This results in the loss of all relative information regarding the interactions of the two in 3D space. Additionally, the underlying complexity of the protein sequence is completely ignored, as each letter in the sequence is encoded as a single unique id. However, each letter represents an amino acid that by itself is a sequence of atoms and bonds that have a particular 3D formation and possible reactions between them and the ligand.

Transformer architectures [8] are used to leverage the advances of Natural Language Processing (NLP) and work with the SMILES representations of compounds. Additional to the above-mentioned limitations, following this approach, the compounds must be created manually or searched in the chemical space. The chemical space has been estimated to be in the order of 10^{63} organic compounds of size up to 30 atoms [9], which infers that iterating over the full space in search of a potential hit is computational expansive and highly inefficient.

Reinforcement Learning techniques [10] have successfully been applied by splitting compounds into meaningful molecule fragments and adding a molecule fragment at a time to produce a score generated by human-defined rules. The model, however, is not gaining insights from the training data, but rather by humanly predefined rules for scoring.

To address these shortcomings, we propose the generation of synthetic small and sophisticated molecule structures that optimize the binding affinity to a target (ASYNT-GAN). To achieve this, we leverage three important achievements in machine learning: attention, deep learning on graphs, and generative adversarial networks. Similar to question answer in NLP, we generate a molecular architecture based on an existing target that functions as context. By exploring the latent space, created by the model, we propose a novel way of searching for candidate compounds suitable for binding.

2. Experimental Section

2.1. Data

Our method is learned from a collection of systems comprised of proteins and ligands, small molecules, used in drug compounds. The proteins are split into chains. Per chain, we extract the proteins, ligands, and their respective binding bonds and transform them using PyMol to their cartoon representation that is then exported as a point cloud, i.e., every point of the structure is encoded with its x, y, z coordinates.

Point cloud learning has attracted increasing attention, due to its wide applications in many areas, such as computer vision, autonomous driving, and robotics [11]. An increasing number of methods are proposed to address various problems related to 3D processing, including 3D shape classification, 3D object detection, and tracking, 3D point cloud segmentation, 3D point cloud registration, 6-DOF pose estimation, and 3D reconstruction [12–14]. During training, we use as input the proteins and a sample Gaussian distribution or a limited number of points sampled from the point cloud of the ligand. All data and scripts needed to reproduce the experiments are provided in the projects GitHub repository [15].

The systems are protein structures from the protein data bank RCSB [16]. The Protein Data Bank archives information about the 3D shapes of proteins, nucleic acids, and complex assemblies that helps students and researchers understand all aspects of biomedicine and agriculture, from protein synthesis to health and disease. As a member of the wwPDB, the RCSB PDB curates and annotates PDB data. The ligands that we consider as valid are the ligands that are referenced as Chemical Component with a Drug Bank [17] identifier.

2.2. Methods

2.2.1. Encoder-Decoder

We first learn the transformation of an input molecule into the latent space using an encoder-decoder architecture with attention. Encoder-decoder architecture has been extensively used in deep learning thanks to its excellent performance for various problems.

Consider the encoder and decoder blocks in Figure 1. The encoder block maps a given input into the latent space, whereas the decoder block takes this latent space map as an input, processes it, and produces an output as similar as possible the input. By doing so, the encoder learns to produce latent features that are of high importance to the essence of the input. The latent space learned by this approach follows a distributed representation, but observes the vector arithmetic phenomenon. We show that by following this approach, we can sample from the latent space for generation purposes and find similar structures by their proximity in the latent space. We identify regions of interest and resample points from those regions. The resampled points are then used together with the initial activations to generate the final output by a run through a second encoder-decoder architecture that may share weights with the first one.

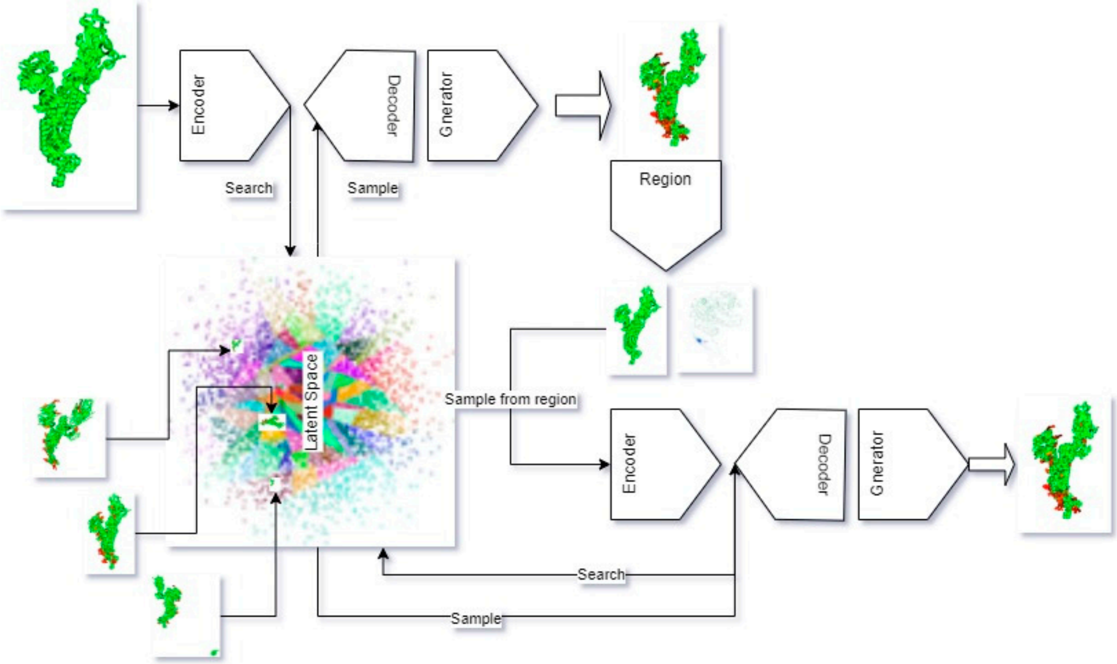


Figure 1. The architecture of the proposed solution. The model consists of an Encoder-Decoder architecture that translates the inputs into the latent space and a Generator that produces the 3D structure of the system. We propose a stacked Generator architecture that takes the first output and calculates regions of interest. We use the regions of interest to resample and generate a second output that is concatenated to the first to produce the prediction of the 3D structure of the molecular system.

The output is the coordinates of the molecules in 3D space. This approach permits the generation of molecule sequences with specific attributes.

2.2.2. Attention Based Generator

We use U-Net architecture, as shown in Figure 2, U-Net [18] decorated with residual blocks and attention gates to encode the point cloud coordinates of the target protein and a PointNet [19] decoder for constructing the ligand structure fitting the binding in 3D space. PointNet architecture permits the direct consumption of point clouds that respects the permutation invariance of the inputs. It similarly helps us work with the 3D input without the need to translate it into an unnecessary voluminous representation of 3D voxel grids.

U-Net architectures are used for segmentation when applied to classification tasks, e.g., U-Net used for Biomedical Image Segmentation [18]. U-Net's have been successfully applied on generative task and have proven to improve the generation process by improving the capacity to synthesize globally and locally coherent shapes and to learn the translation of input from a source domain into a target domain in the absence of paired examples. [20,21]

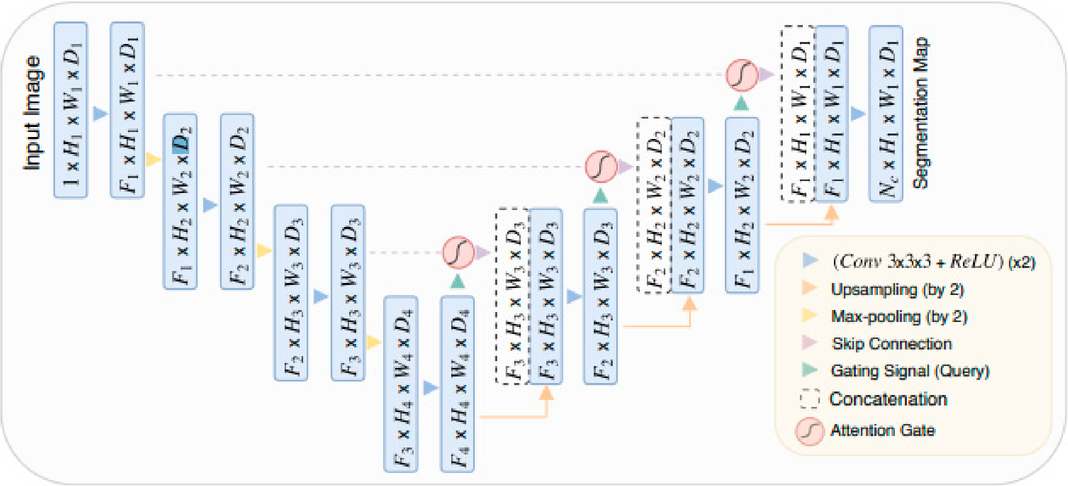

Input Image
$1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_4 \times W_4 \times D_4$
$F_4 \times H_4 \times W_4 \times D_4$
$N_c \times H_1 \times W_1 \times D_1$
Segmentation Map
(Conv 3x3x3 + ReLU) (x2)
Upsampling (by 2)
Max-pooling (by 2)
Skip Connection
Gating Signal (Query)
Concatenation
Attention Gate

Figure 2. Attention U-Net: The inputs are down sampled to a common shape using a convolutional layer followed by a max-pooling. The attention with the previous layer is calculated using the attention gate, and the result is concatenated to the result of the up sampling.

Attention gates are used during the up sampling where the down sampled inputs from the ligand and the protein are concatenated and run through an attention gate. The produced activations are concatenated with previous up sampled activations. The concatenated result is up sampled by the following layer, as seen in Figure 3.

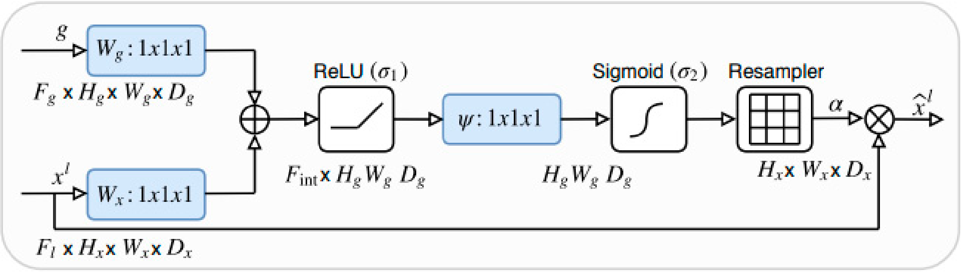

g
$W_g : 1x1x1$
$F_g \times H_g \times W_g \times D_g$
x^l
$W_x : 1x1x1$
$F_l \times H_x \times W_x \times D_x$
ReLU (σ_1)
$F_{int} \times H_g W_g D_g$
$\psi : 1x1x1$
Sigmoid (σ_2)
$H_g W_g D_g$
Resampler
$H_x \times W_x \times D_x$
α
$\hat{x}^l$

Figure 3. Attention Gate (AG) [18]: The input features (x^l) are scaled by the coefficients (α) computed in AG. Regions are selected by analyzing the activations generated by the gating signal (g), which is collected from a coarser scale.

The decoder takes in the latent representation produced from the up sapling in the U-Net and produces the point cloud coordinates for the ligand. The decoder is a Point Net residual network with up sampling capacity in residual blocks Figure 4.

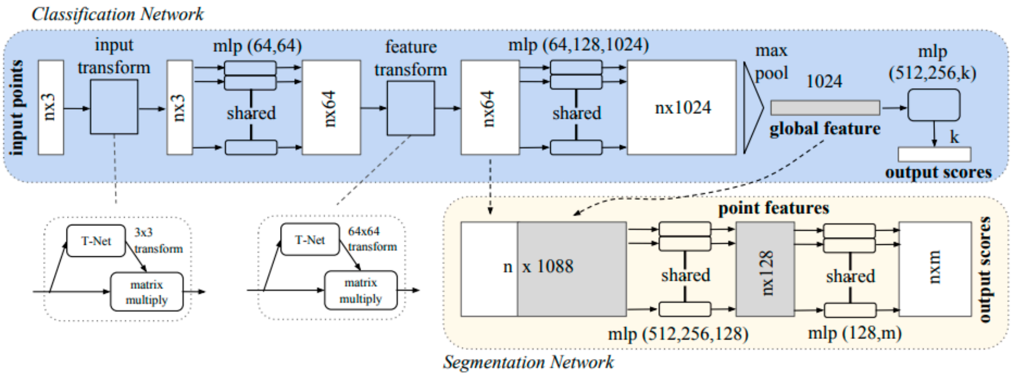


Figure 4. PointNet [19] Architecture takes points in 3D space as input applies feature transformation and applies aggregation with a max-pool layer.

During training, we sample coordinates from the target protein and train the network to produce point samples of the ligand that will produce the best binding affinity in 3D space.

The encoder is trained with a 2048 randomly sampled points from the protein and 64 randomly sampled points from a ligand or a 64 point sampled from the boundary with Gaussian-decaying probabilities. This is done with the purpose of simulating use cases where the ligand or part of it is known and provided as input.

The decoder is conditioned to generate the ligands structure based on the target structure as an input. We train the encoder and decoder with the L_2 loss from the Chamfer distance [22] (CD) that produces the sum of closest point distances, with an additional latent regularization loss to constrain the latent space of the learned embeddings.

We use a symmetric version of the CD, calculated as the sum of the average minimum distance from the point set A to point set B and vice versa. The average minimum distance from one point set to another is calculated as the average of the distances between the points in the first set and their closest point in the second set, and is, thus, not symmetrical.

The loss is given as:

$$Ch(X,Y) = \sum_{x \in X} min_{y \in Y} \|x - y\|_2^2$$

$$d_{CD}(X,Y) = Ch(X,Y) - Ch(Y,X)$$

$$\mathcal{L}_{end}(\theta_e, \theta_d) = \frac{1}{|P\|\,B|} \sum_{i \in P} \sum_{j \in B} L_c\big(d_{CD}\big(D_{\theta_d}\big(x_{i,j}, E_{\theta_e}(g_i)\big), t_{i,j}\big)\big) + \lambda \|E_{\theta_e}(g_i)\|_2$$

where P is the set of all training molecules in each mini-batch, B is the set of point samples sampled per target, $\mathcal{L}_c(.,.)$ is the $\mathcal{L}2$ loss, E_{θ_e} is the encoder parameterized by trainable parameters θ_e, D_{θ_d} is the decoder parameterized by trainable parameters θ_d, and g_i is the sampled point cloud for the *i*-th binding ligand structure.

2.2.3. Metrics

For our experiments, we evaluate the generative quality with CD [22]. We estimate CD using 1024 randomly sampled points on the ground truth and generated systems. We have tested the Chamfer Distance [22] on a series of viral Proteins of the Severe acute respiratory syndrome coronavirus 2.

2.3. *Similarity Search*

We translate the inputs, protein, and ligands, into the latent space. We can use the properties of the encoder to index systems or part of systems and perform a search for similar systems.

The embeddings of all the systems are inserted into an index and searched for similarities using Approximate nearest neighbor [23].

An approximate nearest neighbor search algorithm can return points, whose distance from the query is at most c times the distance from the query to its nearest points.

The appeal of this approach is that, in many cases, an approximate nearest neighbor is almost as good as the exact one. In particular, if the distance measure accurately captures the notion of user quality, then small differences in the distance should not matter.

The search in latent space can be done during training or during inference. During training, if the ligand is partially known, its latent representation can be used to look for candidates instead of sampling from the Gaussian-decaying probabilities. The latent space has a structure that can be explored as shown in Figure 5, such as by interpolating between points and performing vector arithmetic between points. For instance, we can use the best match from the approximate nearest neighbor search as a starting point for a walk through the latent space.

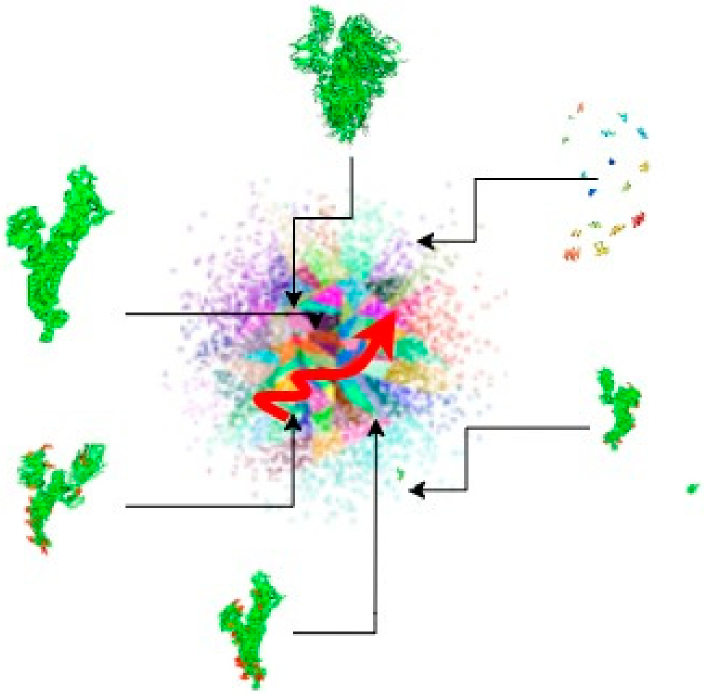


Figure 5. Approximate nearest neighbor [10] search returns points, whose distance from the query is at most c times the distance from the query to its nearest points.

2.4. Progressive Training

We introduce the notion of progressive training. During training, we progressively reduce the number of sampled points from the ligands. We start by sampling 1042 points from the point cloud of the ligand, and gradually reduce to zero. When reduced to zero, we sample from Gaussian-decaying probabilities as an input to the up sampling part of the attention-based U-Net. We observe an overall stabilization and faster convergence of the generator.

2.5. Stacked Generators

We introduce the notion of stacked generators. The points generated from the first generator layer are used as attention regions for the second generator layer. Points from the target protein are sampled from these regions, as shown in Figure 6, and used as input for the next Generator Layer.

We trained the generator layers with shared weights and separately. In both cases, we noticed a significant increase in accuracy in the second generative layer, as well as a stabilization of the overall loss during the training, as shown in Figures 7 and 8.

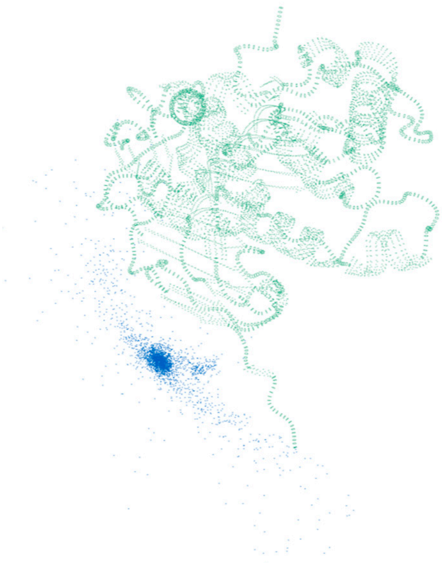


Figure 6. Shows the regions of interest identified to resample data from those regions. The data is concatenated to the activations from the first generator and used as input by the stacked generator.

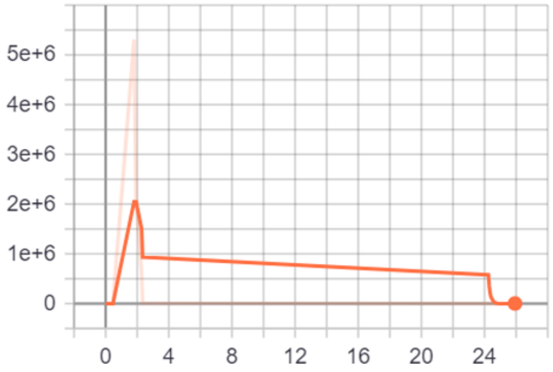


Figure 7. Depicts the evolution of the loss and stabilization using the stacked progressive generator approach. The y-axis indicates the values produced by the loss function and x-axis the training step for the respective loss value.

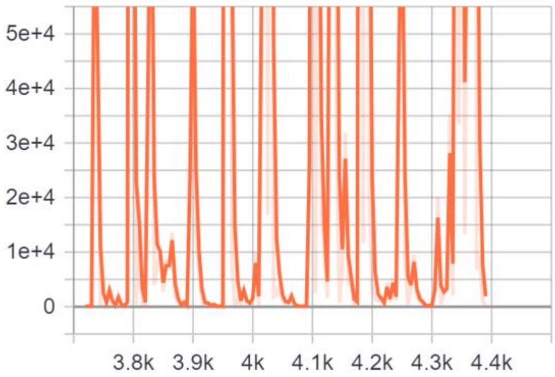


Figure 8. Depicts the loss of the evolution without using the stacked generator network approach. The y-axis indicates the values produced by the loss function and x-axis the training step for the respective loss value.

2.6. Interpolation

We experimented with an interpolation approach that takes in the attention grids from the Attention U-Net and interpolates the input points. The interpolated points are fed into

Residual Network [24] consistent with PointNet Dense Layers. This approach has shown promising results in converging fast in coordinates of ligands.

This approach would be a good fit when systems are split into meaningful sub-systems, and generation is done in particular 3D sub-spaces. As shown in Figure 9, the interpolation produces less noise and in contrast to Figure 10. However, not enough points are generated.

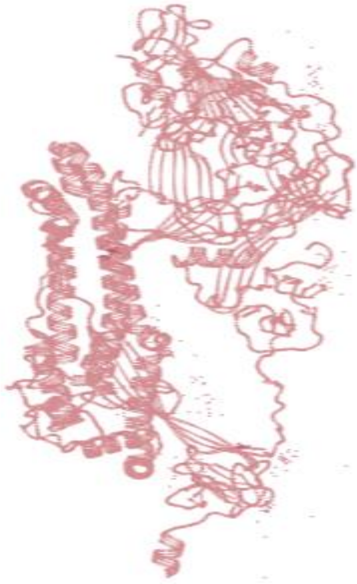


Figure 9. Generated using the interpolation approach produces points with less noise.

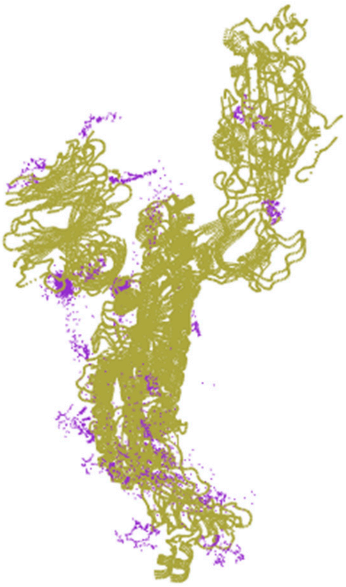


Figure 10. Generated using the stacked generator approach is noisier, but produces more details.

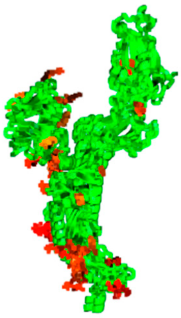
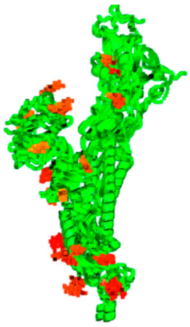
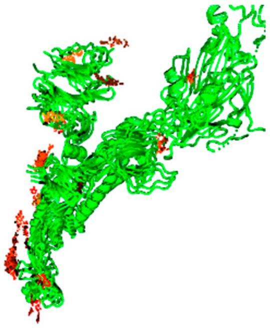
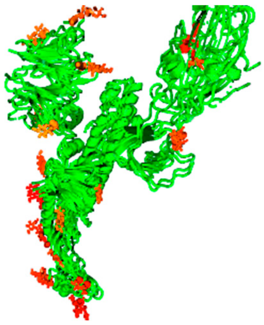
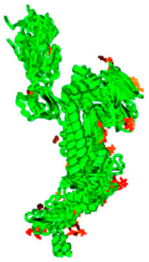
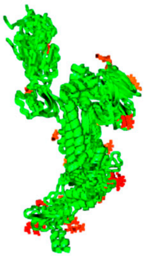
3. Results

We quantitatively and qualitatively compare performances in Tables 1 and 2, respectively. Given our solution is trained to learn a latent representation of ligands; the learned representation does generalize to systems and chains beyond the source system. Visually, as shown in Table 2, our solution achieves a good generation of complete structure that optimizes the binding molecules in the system (e.g., ligand and protein), but performs poorly in terms of generating point clouds without noise, as shown in Figure 11.

Table 1. Quantitative representation of the metric applied to the results produced.

Protein	Chain	Chamfer Distance
6VYB	B	49.71
6VYB	A	90.33
6VW1	A	5.95
6VW1	B	16.87
6WPT	A	78.67
6WPT	B	29.55
6WPT	C	58.04
6WPT	E	48.25
6VXX	A	69.02
6VXX	B	107.20

Table 2. Visual representation of the produced results.

Input	Prediction	Ground-Truth
		
		
		

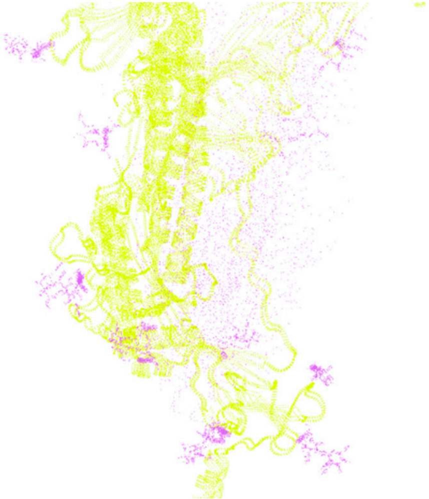


Figure 11. Depicts the amount of noise in the generated structure.

4. Discussion and Conclusions

The generation of synthetic small and more sophisticated molecule structures that optimize the binding affinity to a target (ASYNT-GAN) through encoding a protein and generating a system comprised of a ligand and a protein. Experiments show that ASYNT-GAN is able to generate ligand structures for proteins unseen during training. Translating the input sub-systems into the latent space permits the reachability for similar structures and the sampling from the latent space for a generation. Topics for future work include ways of integrating the search capabilities in the training process, explore alternatives for sampling and generating points ASYNT-GAN from regions of interest, provide for the ability to generate alternative variants of proteins to predict mutations.

Current approaches require a biochemist to manually seek or create existing potential ligand that can bind to a target. The selected target is then manually placed in potential binding pockets and aligned with the target to evaluate the reactions. The selection of the potential pockets is made based on the scientists' knowledge and potential literature discussing the particular protein. The binding affinity is evaluated using simulation techniques, e.g., docking. This process is repeated until the best fit is found.

In contrast to the above-described approach, ASYNT-GAN can directly produce ligand structures represented in a point cloud in their respective binding pockets in the alignment that optimizes binding affinity. The search approach described in Section 2.3 produces a list of indexed ligands.

A concrete example of the usage of ASYNT-GAN is as follows. Protein 6VYB [25] is converted to point cloud and analyzed chain per chain, as shown in Figure 12. We represent the ligand as a sample from a Gaussian distribution, i.e., we are unaware of potential binding ligand and wish the model to generate it. Using the point cloud of the protein and the sample from the Gaussian distribution, ASYNT-GAN generates the point clouds of the ligands, as shown in Figure 13, in multiple shades of orange. ASYNT-GAN's prediction provides the areas where the ligand will bind with the higher affinity, the alignment, and the ligand in point cloud representation. ASYNT-GAN's prediction also provides the translation of the generated ligand into the latent space, which permits the search for a list of candidates in the index of ligands, as described in Section 2.3. Additionally, we can perform vector arithmetic between points in latent space, which have meaningful and targeted effects, e.g., we can combine the predicted ligands with a chain from another protein and start a lookup from that point in the latent space. The results can similarly be used to initiate a search and walk through the latent space to provide a broader list of candidate ligands and proteins, as shown in Figure 14.

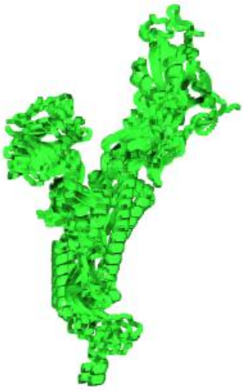


Figure 12. Depicts the point cloud of Chain A of 6VYB [25] SARS-CoV-2 spike ectodomain structure.

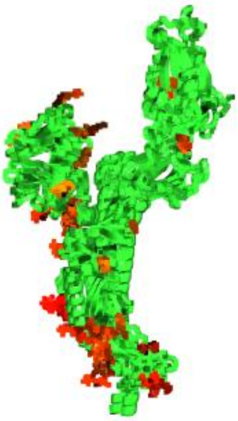


Figure 13. Generated ligands predictions are the points in multiple shades of orange.

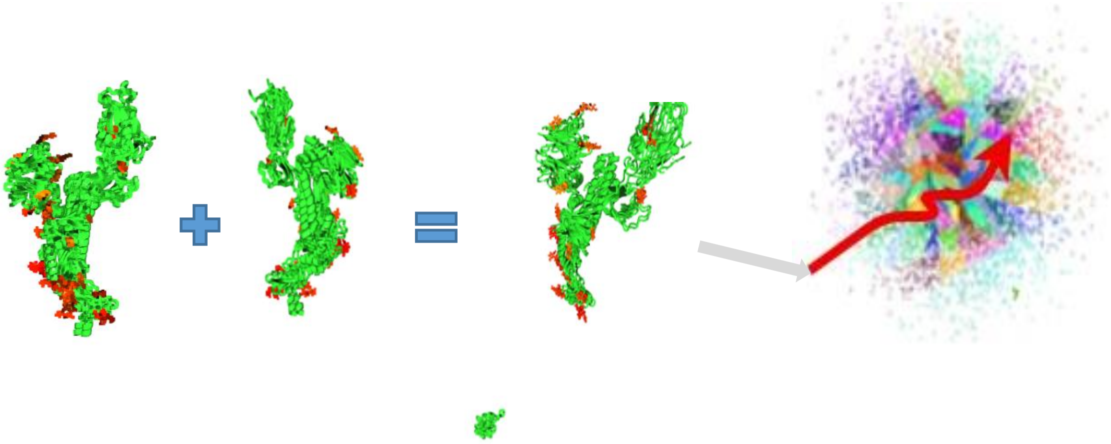


Figure 14. Shows the ability to perform vector arithmetic for search and walk through latent space.

It is important to note that the latent space representations hold important insights that the model learned throughout the training where its task is to understand the structures and generate new structures based on shape and interactions. This results in a latent space that holds latent features that have high value in terms of shape and interactions between the elements. Additionally, they are comparable and combinable that permits sophisticated search strategies.

This is in contrast to current search approaches, which do not take interactions and folding in 3D space into account, and do not permit any arithmetic between multiple elements where sequences are compared—resulting into a best match that is a sequence with the most overlapping elements with the query sequence.

In the presented experiments, we have taken the approach to generate ligands into 3D space that bind into the target protein. The same approach can be taken for any interaction between elements in 3D space, e.g., amino acid interaction with other amino acids in a chain, protein interaction with other proteins, sub-systems interaction in a macro molecule, atom, and bonds interaction with other atom and bonds.

Author Contributions: Conceptualization, I.J.; methodology, I.J.; software, I.J.; validation, M.M.; formal analysis, I.J. and M.M.; investigation, I.J.; resources, I.J.; data curation, I.J.; writing—original draft preparation, I.J.; writing—review and editing, I.J. and M.M.; visualization, I.J.; supervision, M.M.; project administration, I.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Berry, M.; Fielding, B.; Gamieldien, J. Chapter 27—Practical Considerations in Virtual Screening and Molecular Docking. In *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*; Tran, Q.N., Arabnia, H., Eds.; Morgan Kaufmann Publishers: Boston, MA, USA, 2015; pp. 487–502.
2. Palma, G.; Vidal, M.-E.; Raschid, L. *Drug-Target. Interaction Prediction Using Semantic Similarity and Edge Partitioning*; Springer: Cham, Switzerland, 2014; Volume 8796, p. 146.
3. Shiloh-Perl, L.; Giryes, R. introduction to deep learning. *arXiv* **2020**, arXiv:2003.03253.
4. Xu, W.; Lucke, A.J.; Fairlie, D.P. Comparing sixteen scoring functions for predicting biological activities of ligands for protein targets. *J. Mol. Graph. Model.* **2015**, *57*, 76–88. [CrossRef] [PubMed]
5. Yang, T.; Wu, J.C.; Yan, C.; Wang, Y.; Luo, R.; Gonzales, M.B.; Dalby, K.N.; Ren, P. Virtual screening using molecular simulations. *Proteins Struct. Funct. Bioinform.* **2011**, *79*, 1940–1951. [CrossRef] [PubMed]
6. Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: Deep drug–target binding affinity prediction. *Bioinformatics* **2018**, *34*, 821–829. [CrossRef] [PubMed]
7. SimBoost: A Read-Across Approach for Predicting Drug–Target Binding Affinities Using Gradient Boosting Machines. Available online: https://www.researchgate.net/publication/316235177_SimBoost_a_read-across_approach_for_predicting_drug-target_binding_affinities_using_gradient_boosting_machines (accessed on 27 September 2020).
8. Shin, B.; Park, S.; Kang, K.; Ho, J.C. Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction. Available online: http://arxiv.org/abs/1908.06760 (accessed on 27 September 2020).
9. Organic Chemistry. Available online: https://global.oup.com/ukhe/product/organic-chemistry-9780199270293?cc=lu&lang=en& (accessed on 27 September 2020).
10. Tang, B.; He, F.; Liu, D.; Fang, M.; Wu, Z.; Xu, D. AI-aided design of novel targeted covalent inhibitors against SARS-CoV-2. *BioRxiv* **2020**. [CrossRef]
11. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. Available online: http://arxiv.org/abs/1912.12033 (accessed on 13 October 2020).
12. Elbaz, G.; Avraham, T.; Fischer, A. 3D Point Cloud Registration for Localization Using a Deep Neural Network Auto-Encoder. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2472–2481. [CrossRef]
13. Zeng, A.; Yu, K.; Song, S.; Suo, D.; Ed, W., Jr.; Rodriguez, A.; Xiao, J. Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. Available online: http://arxiv.org/abs/1609.09475 (accessed on 13 October 2020).
14. Han, X.-F.; Laga, H.; Bennamoun, M. Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef] [PubMed]
15. Source Code GitHub. ai4u-ai/ASYNT-GAN.ai4u. Available online: https://github.com/ai4u-ai/ASYNT-GAN (accessed on 28 September 2020).
16. Bank, R.P.D. RCSB PDB: Homepage. Available online: https://www.rcsb.org/ (accessed on 27 September 2020).
17. Wishart, D.S.; Feunang, Y.D.; Guo, A.C.; Lo, E.J.; Marcu, A.; Grant, J.R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Res.* **2018**, *46*, D1074–D1082. [CrossRef] [PubMed]
18. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Available online: http://arxiv.org/abs/1505.04597 (accessed on 27 September 2020).

19. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Available online: http://arxiv.org/abs/1612.00593 (accessed on 27 September 2020).
20. Schönfeld, E.; Schiele, B.; Khoreva, A. A U-Net Based Discriminator for Generative Adversarial Networks. Available online: http://arxiv.org/abs/2002.12655 (accessed on 19 October 2020).
21. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Available online: http://arxiv.org/abs/1703.10593 (accessed on 19 October 2020).
22. Hajdu, A.; Hajdu, L.; Tijdeman, R. Approximations of the Euclidean distance by chamfer distances. Available online: http://arxiv.org/abs/1201.0876 (accessed on 27 September 2020).
23. Andoni, A.; Indyk, P.; Razenshteyn, I. Approximate Nearest Neighbor Search in High Dimensions. Available online: http://arxiv.org/abs/1806.09823 (accessed on 27 September 2020).
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. Available online: http://arxiv.org/abs/1512.03385 (accessed on 27 September 2020).
25. Bank, R.P.D. RCSB PDB—6VYB: SARS-CoV-2 Spike Ectodomain Structure (Open State). Available online: https://www.rcsb.org/structure/6VYB (accessed on 19 October 2020).