Visualizing Deep Graph Generative Models and Target-Specific Generation Experiments For Drug Discovery

Karan Yang
Cornell Tech
New York, New York City, USA
ky393@cornell.edu

Chengxi Zang
Weill Cornell Medicine
New York, New York City, USA
chz4001@med.cornell.edu

Fei Wang
Weill Cornell Medicine
New York City, New York, USA
few2001@med.cornell.edu

ABSTRACT

Drug discovery aims at designing novel molecules with specific desired properties for clinical trials. Driven by big chemical data and AI, deep generative models show great potential to accelerate the drug discovery process. Existing work investigates different deep generative frameworks for molecular generation. However, less attention has been paid to visualization tools that enable a rapid demo and evaluation of a model's results. Additionally, accurate Drug-Target binding prediction is crucial for drug discovery. Yet, binding affinity data is unavailable for novel target proteins, such as the SARS-CoV-2 viral proteins involved in the recent COVID-19 pandemic. It is apparent that the generation of novel molecules with high binding affinity to a novel target protein is a challenging and non-trivial exercise, but also an essential one. In this study, we attempt to tackle the two problems outlined above in the realm of de novo drug design. First, we propose a visualization framework which provides interactive visualization tools to visualize molecules generated during the encoding-and-decoding process of deep graph generative models. Also, we provide real-time molecular optimization functionalities. Second, we propose an end-to-end de novo drug design approach to generate novel molecules with high binding affinity to a specific target protein. We have conducted some initial experiments to leverage the power of MoFlow[11] (a generative model) and the pre-trained drug-target binding affinity prediction models from DeepPurpose[8]. Our work tries to empower black-box AI-driven drug discovery models with some visual interpretive abilities. We believe our initial exploration of generating target-specific novel drug molecules will provide valuable insights for AI-based approaches to timely combat future unforeseen pandemics.

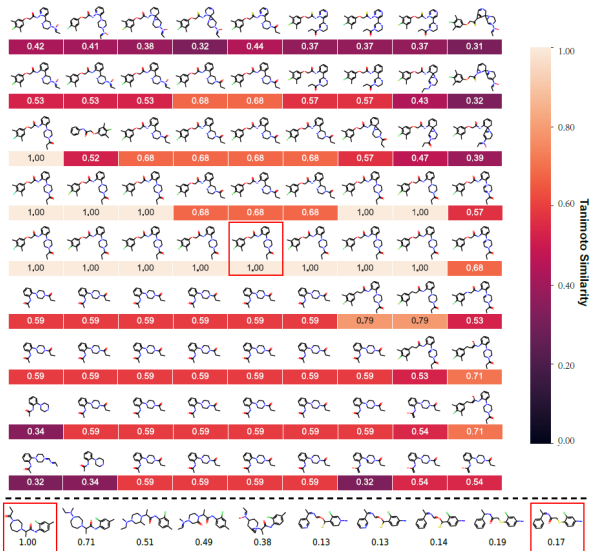


Figure 1: Visualization of learned latent space by MoFlow[11]. Top: Visualization of the grid neighbors of a seed molecule in the center, which serves as the baseline for measuring similarity. Bottom: Interpolation between two seed molecules and the left one is the baseline molecule for measuring similarity. Seed molecules are highlighted in red boxs and they are randomly selected from ZINC250K.

KEYWORDS

De novo drug discovery; Deep graph generative model; Molecular graphs generation; Visualization tools; Drug-target interaction

1 INTRODUCTION

1.1 Interactive visualization framework

Deep graph generative models are promising to accelerate the long and costly drug discovery process by exploring large chemical space in a data-driven manner. These models first learn a continuous latent space by encoding molecular graphs and then generate novel and optimized molecules by decoding from the learned latent space guided by some targeted properties [6, 9, 11]. We illustrate these encoding-and-decoding pipelines in Figure 1.

Despite their promising results, deep graph generative models are challenging to work with and the training process and results lack transparency and interpretability, in general. These challenges naturally lead us to search for a visualization tool that will help to interpret each step of how the model generates novel chemical structures in its latent space, to decompose the analysis of how the interpolation between molecules are formed and to gain insights over how the optimized functional compounds are yielded. Furthermore, a three-dimensional and interactive molecule visualization tool is needed to develop useful insights - as it is very important and informative to visually examine how the generated molecule's atoms are positioned relative to each other in 3D space.

While there are some good visualization tools designed for the deep generative models - such as, "Glow visualization demo" in Figure 3, we have not seen effective visualization tools tailored to the deep generative models for the molecular graph.

In this paper, we propose to build a web dashboard that provides three-dimensional and interactive molecular visualization tools designed specifically for the deep generative models for the

molecular graphs. Our dashboard proposes an interface to operate on chemical structures directly with following functionalities:

- Visualize the generated molecules with different levels of atom outlines, ambient occlusion, and more with the various viewing parameters(angles, rotation, zoom, highlights).
- Visualize the chemical similarity between each neighboring molecule and the centering molecule.
- Visualize the linear interpolation between two molecules to show their changing trajectory in the latent space.
- Visualize the optimized molecules under different objective properties.
- Query generated novel molecules based on seed molecules at real-time with the deep graph generative model running backend.

Our dashboard works seamlessly with Python-based open source chemoinformatics and machine learning packages such as RDKit [1] and PyTorch/TensorFlow.

1.2 Target-specific novo drug generation

The application of deep learning in drug discovery mainly into three different categories: (1) Drug properties prediction. (2)De Novo drug design. (3) Drug-target interaction (DTI) prediction. In this paper,

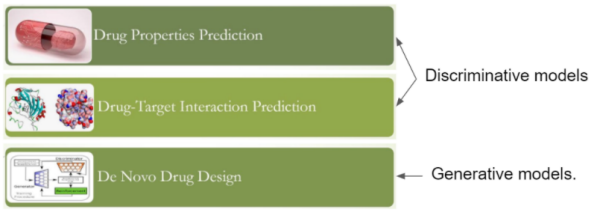


Figure 2: Three areas that deep learning can facilitate drug discovery.

we focus on DTI and de novo drug design. Specifically, we aspire to integrate both tasks and to leverage the power of MoFlow model and the pre-trained state-of-art drug-target binding affinity prediction models to generate de novo molecules for unseen target protein. The motivation for our end-to-end research design is to develop an approach that reduces both the high cost structure and the time consuming nature of physical based simulation for novel drug design - while yielding a design approach with a higher success rate. It is desirable (and urgent) to have a timely and adaptive approach to generate novel drug-candidates. Drug-Target Binding Affinity (DTBA)DTBA indicates the strength of the interaction, or binding, between a drug and its target (Ma W. et al., 2018). This binding is an essential feature of the SARS-CoV-2 viral proteins involved in the recent COVID-19 pandemic. However, such binding affinity data is unavailable. This deficiency highlights the need for a timely and adaptive approach to generate novel drug-candidate(s) - especially for new unseen target protein.

The current MoFlow model is one of the first flow-based models which not only generate molecular graphs at one-shot by invertible mappings, but also have a validity guarantee. It is a powerful and desirable feature to generate novel molecules with optimized drug-related properties. However, the MoFlow model has a drawback in that it lacks the functionality to generate novel molecules for a specific target protein sequence.

Therefore, we propose a way to overcome this deficiency via a modified model architecture - by introducing a drug target binding affinity prediction model and using the predicted binding scores as an additional property to be optimized. We discuss and review in detail the results from our initial experiments on this approach to train on the ZINC dataset and the Sar-cov-2 3CL Protease.

2 RELATED WORK.

2.1 Visualization frameworks

Interactive interfaces and visualizations of the sampled data from the chemical latent space of deep learning models have been designed and developed to help people understand what models have learned and how they make predictions. Many of those visualization tools require significant workarounds to pre-existing graph types. Here, We discuss three main visualization tools for the deep graph generative models.

RDKit-Neo4j project. This is a development of extension for neo4j graph database for querying knowledge graphs storing molecular and chemical information[4]. The project's task is to enable identification of entry points into the graph via exact or substructure/similarity searches. The intention is to use chemical structures as limiting conditions in graph traversals originating from different entry points. Neo4j is coded in CYPHER which is quite complex.

Tableau 3D project. In a tableau project[2], the Caffeine molecule example uses a dual axis chart. One axis draws the atoms while the other axis draws the bonds between them. The project relies purely on the background image, mark size and z-order to achieve the 3D look. One issue with Tableau evolves from its dual axis structure, as z-order does not always work well - as Tableau sorts within each axis independently, so in this setup atoms are always drawn above bonds but each group is sorted within themselves.

"Glow visualization". One interesting online visualization tool for the "Glow", a reversible generative model, is shown in the figure 3[3].

However, a limitation of the "Glow visualization" is that it is built upon javascript and only works on the images. Our final goal would be to build an alternative, but similar, interactive visualization tool that is designed specifically for drug discovery purposes and for the generation of the molecular structure under varying parameters.

2.2 AI-based Drug-Target prediction models

Drug-Target Binding Affinity (DTBA). DTBA indicates the strength of the interaction or binding between a drug and its target (Ma W. et al., 2018). The advantage of formulating drug-target prediction as a binding affinity regression task, is that it can be transformed from regression to either binary classification by setting specific thresholds or to ranking problem (He et al., 2017)[7]. This enables different generalization options. To predict drug-target binding affinities (DTBA) are of great value in early stages of drug discovery.

[1] https://github.com/rdkit/rdkit

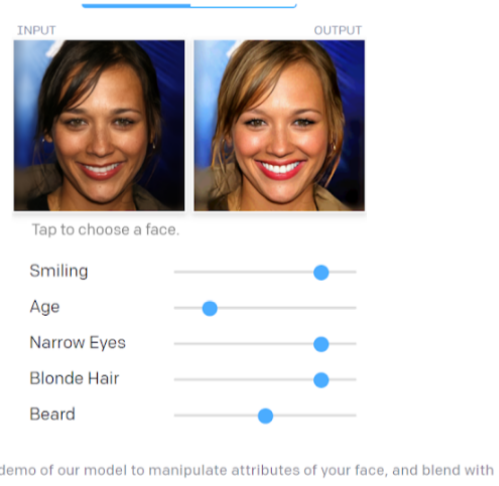


Figure 3: An interactive demo of the Glow model to manipulate attributes of the face, and blend with other faces.[3]

DeepPurpose. DeepPurpose ia a Deep Learning Based Molecular Modeling and Prediction Toolkit on Drug-Target Interaction Prediction, Compound Property Prediction, Protein-Protein Interaction Prediction, and Protein Function prediction (using PyTorch).[8] DeepPurpose uses an encoder-decoder framework for drug-target interaction prediction. DeepPurpose will take the drug's simplified molecular-input line-entry system (SMILES) string and target amino acid sequence pair as input, and output a score indicating the binding activity of the drug target pair.

3 METHOD

3.1 MOFLOW MODEL-A Deep Generative Model of Molecular Graphs

As an initial experiment to build our broad web dashboard, we have underway a process to build a web dashboard specifically for the MoFlow model for molecular graph generation proposed by Zang and Wang[11]. MoFlow is one of the first flow-based models which not only generates molecular graphs at one-shot by invertible mappings but also has a validity guarantee. MoFlow consists of a variant of Glow model for bonds, a novel graph conditional flow for atoms given bonds, and then combining them with post-hoc validity correction. MoFlow achieves many new state-of-the-art performance on molecular generation, reconstruction and optimization. We summarize the inference and generation procedure of the MoFlow in Algorithm 1 and Algorithm 2 respectively.

3.2 Visualization Implementation

Our approach is to integrate **Pytorch, rdkit and Dash/Dash bio** to build a web dashboard that provides three-dimensional and interactive molecule visualization tools. This framework would be able to combine pytorch - an open source machine learning framework, Rdkit - an Open-Source Cheminformatics Software, and Dash - a Python framework for building web applications. Written on top of

Algorithm 1: Exact Likelihood Inference (Encoding) by MoFlow

Input: $f_{\mathcal{A}|\mathcal{B}}$: graph conditional flow for atoms, $f_{\mathcal{B}}$: glow for bonds, A: atom matrix, B: bond tensor, $P_{\mathcal{Z}_*}$: isotropic Gaussian distributions.
Output: Z_M:latent representation for atom M, $\log P_{\mathcal{M}}(M)$: logarithmic likelihood of molecule M.

$Z_B = f_{\mathcal{B}}(B)$
$\log P_{\mathcal{B}}(B) = \log P_{\mathcal{Z}_{\mathcal{B}}}(Z_B) + \log \mid \det(\frac{\partial f_{\mathcal{B}}}{\partial B}) \mid$
$\hat{B}$ = graphnorm(B)
$Z_{A|B} = f_{\mathcal{A}|\mathcal{B}}(A|\hat{B})$
$\log P_{\mathcal{A}|\mathcal{B}}(A|B) = \log P_{\mathcal{Z}_{\mathcal{A}|\mathcal{B}}}(Z_{A|B}) + \log \mid \det(\frac{\partial f_{\mathcal{A}|\mathcal{B}}}{\partial A}) \mid$
$Z_M = (Z_{A|B}, Z_B)$
$\log P_{\mathcal{M}}(M) = \log P_{\mathcal{B}}(B) + \log P_{\mathcal{A}|\mathcal{B}}(A|B)$
Return: Z_M, $\log P_{\mathcal{M}}(M)$

Algorithm 2: Molecule Generation (Decoding) by the Reverse Transformation of MoFlow

Input: $f_{\mathcal{A}|\mathcal{B}}$: graph conditional flow for atoms, $f_{\mathcal{B}}$: glow for bonds, Z_M:latent representation of molecule M or sampling from a prior Gaussian, validity-correction: validity correction rules.
Output: M: a molecule

$(Z_{A|B}, Z_B) = Z_M$
$B = f_{\mathcal{B}}^{-1}(Z_B)$
$\hat{B}$ = graphnorm(B)
$A = f_{\mathcal{A}|\mathcal{B}}^{-1}(Z_{A|B}|\hat{B})$
M = validity-correction(A, B)
Return: M

Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly customized user interfaces in pure Python. To summarize, the main components of our visualization tools for the molecular generation are:

- Use Dash as the web dashboard UI interface design. .
- Callback functions that take an dataset and the best model's parameters as input and connect to the model's encoder and decoder(Algorithm 1 and 2 in figure..)
- Use Rdkit functions to convert molecular data to 2D view and to "xyz" file for the 3D view.
- Use Dash Bio-speck[1] to display the output generated by the model's decoder in an interactive Molecule 3D View

3.3 Molecule Generation Constrained by Target-Specific Binding Affinity

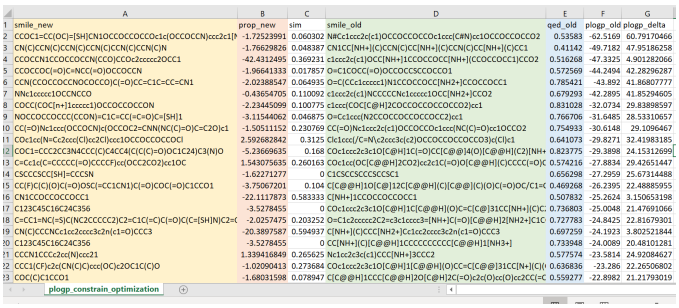
3.3.1 ***Datasets and Model.*** The target sequence we use is the SARS-CoV2 3CL Protease main protease. We choose the MPNN-CNN- BindingDB from DeepPurpose as the pre-trained predictor. Here we try MPNN for drug and CNN for target. This model is pretrained on BindingDB Kd dataset[2]. The drug encoder, MPNN Gilmer et al.[5] is a message-passing graph neural network that operate on the compound molecular graph.After obtaining embedding vectors for each atom and edge, a readout function (mean/sum) is used to obtain a (molecular) graph-level embedding vector.The target encoder: CNN Krizhevsky et al. [2012] is a multi-layer 1D

[2]https://www.bindingdb.org/bind/info.jsp

convolutional neural network. The target amino acid is decomposed to each individual character and is encoded with an embedding layer and then fed into the CNN convolutions. It follows a global max pooling layer.

3.3.2 ***Implementation.*** As depicted in Figure 5, our Implementation of Molecule Generation Constrained by Target-Specific Binding Affinity has two main components:

Step 1: Generate molecules with desired properties(e.g. Plop, QED) using MoFlow - a generative model. We output the generated molecules in a csv file (as below) for further analysis.



	smile_new	prop_new	sim	smile_old	qed_old	plogp_old	plogp_delta
2	CCOC1=CC(OC)=[SH]CN1OCCOCCOCCOc1c(OCCOCCN)ccc2c1[N	-1.72523991	0.060302	NNCc1ccc2c(c1)OCCOCCOCCOc1ccc(CNN)cc1OCCOCCOCCO2	0.53583	-62.5169	60.79170466
3	CN(C)CCN(C)CCN(C)CCN(C)CCN(C)CN(C)N	-1.76629826	0.048387	CN1CC[NH+](C)CCN(C)CC[NH+](C)CCN(C)CC[NH+](C)CC1	0.41142	-49.7182	47.95186258
4	CCOCCN1COCCOCCN(CCO)CCOc2ccccc2OCC1	-42.4312495	0.369231	c1ccc2c(c1)OC[NH+]1CCOCCOCC[NH+](CCOCCOCC1)CCO2	0.516268	-47.3325	4.901282066
5	CCOCCOC(=O)C=NCC(=O)OCCOCCN	-1.96641333	0.017857	O=C1COCC(=O)OCCOCCSCCOCCO1	0.572569	-44.2494	42.28296287
6	CCN(CCOCCOCNCOCCO)C(=O)CC=C1C=CC=CN1	-2.02388547	0.064935	O=C(CC1ccccc1)N1CCOCCOCC[NH2+]CCOCCOCC1	0.785421	-43.892	41.86807777
7	NNc1ccccc1OCCNCCO	-0.43654705	0.110092	c1ccc2c(c1)NCCCCCNc1ccccc1OCC[NH2+]CCO2	0.679293	-42.2895	41.85294605
8	COCC(COC[n+]1ccccc1)OCCOCCOCCON	-2.23445099	0.100775	c1ccc(COC[C@H]2COCCOCCOCCOCCO2)cc1	0.831028	-32.0734	29.83898597
9	NOCCOCCOCCC(CCON)=C1C=CC(=C=O)C=[SH]1	-3.11544062	0.046875	O=C1ccc(N2CCOCCOCCOCCOCC2)cc1	0.766706	-31.6485	28.53310657
10	CC(=O)Nc1ccc(OCCOCN)c(OCCOC2=CNN(NC(C)=O)C=C2O)c1	-1.50511152	0.230769	CC(=O)Nc1ccc2c(c1)OCCOCCOc1ccc(NC(C)=O)cc1OCCO2	0.754933	-30.6148	29.1096467
11	COc1cc(N=C2ccc(C)cc2C)cc1OCCOCCOCCOCl	2.592682842	0.3125	COc1ccc(/C=NNc2ccc3c(c2)OCCOCCOCCOCCO3)cC)c1	0.641073	-29.8271	32.41983185
12	COC1=CCC2CC3N4CCC(/C)C4CC4(C(C)C)=O)OC1C24)C3(N)O	-5.23669635	0.168	COc1ccc2c3c1O[C@H]1C(=O)CC[C@@]4(O)[C@@H](C2)[NH+	0.823775	-29.3898	24.15312699
13	C=Cc1c(C=CCCCC(=O)CCCC)cc(OCC2CO2)cc1OC	1.543075635	0.260163	COc1cc(OC[C@@H]2CO2)cc2c1C(=O)O[C@@H](C)CCCC(=O)C	0.574216	-27.8834	29.42651447
14	CSCCCSCC[SH]=CCCSN	-1.62271277	0	C1CSCSCSCCSCSCSC1	0.656298	-27.2959	25.67314488
15	CC(F)C(C)(O)C(=O)OSC(=CC1CN1)C(=O)COC(=O)C1CCO1	-3.75067201	0.104	C[C@@H]1O[C@]12C[C@@H](C)[C@@](C)(O)C(=O)OC/C1=C	0.469268	-26.2395	22.48885955
16	CN1CCOCCOCCOCC1	-22.1117873	0.583333	C[NH+]1CCOCCOCCOCC1	0.507832	-25.2624	3.150653198
17	C123C45C16C24C356	-3.5278455	0	COc1ccc2c3c1O[C@H]1C[C@@H](O)C=C[C@]31CC[NH+](C)C	0.736803	-25.0048	21.47691066
18	C=CC1=NC(=S)C(NC2CCCCC2)C2=C1C(=C)C(=O)C(C=[SH]N)C2=C	-2.0257475	0.203252	O=C1c2ccccc2C2=c3c1cccc3=[NH+]C(=O)[C@@H]2[NH2+]C1C	0.727783	-24.8425	22.81679301
19	CN(C)CCCNCc1cc2cccc3c2n(c1=O)CCC3	-20.3897587	0.594937	C[NH+](C)CCC[NH2+]Cc1cc2cccc3c2n(c1=O)CCC3	0.697259	-24.1923	3.802521844
20	C123C45C16C24C356	-3.5278455	0	CC[NH+](C)[C@@H]1CCCCCCCCC[C@@H]1[NH3+]	0.733948	-24.0089	20.48101281
21	CCCN1CCCc2cc(N)ccc21	1.339416849	0.265625	Nc1cc2c3c(c1)CCC[NH+]3CCC2	0.577574	-23.5814	24.92084627
22	CCC3(CF)c2c(CN(C)C)ccc(OC)c2OC1C(C)O	-1.02090413	0.273684	COc1ccc2c3c1O[C@H]1[C@@H](O)CC=C1C[C@@]31CC[NH+](C)(	0.636836	-23.286	22.26506802
23	COC(C)C1CCO1	-1.68031598	0.078947	C[C@@H]1CCC[C@@H]2O[C@H]2C(=O)c2c(O)ccc(O)cc2CC(=C	0.559277	-22.8982	21.21793019

plogp_constrain_optimization

Figure 4

Step 2: For the Drug-Target Binding Affinity prediction, we pass the generated molecules and a target protein as input and use a pre-trained predictor from DeepPurpose to rapidly produce the binding affinity score.

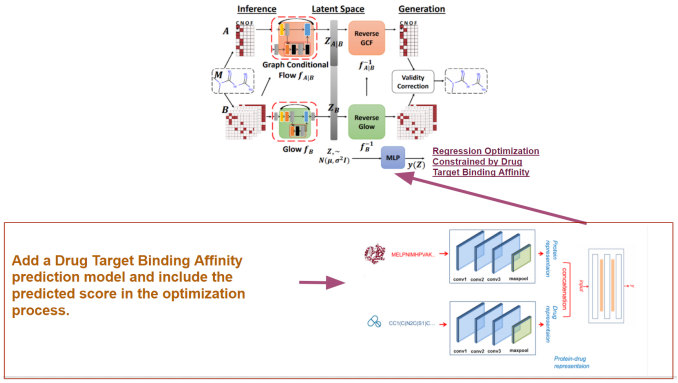


Figure 5

4 RESULTS

4.1 Dashboard

Here we present some of the main features and highlights of our dashboard. On the left side of the dashboard, The users can navigate to 4 types of visualiation tasks. Currently, we use the default model : Moflow and default dataset: ZINC

We built an efficient web dashboard with a dropdown menu, allowing people to select a drug's name and display the 2D and 3D interactive view. This dashboard allows users to manipulate attributes to explore the latent space (Note: we can rotate the 3D molecule). Figure 6-9 are sample demos of our dashboard designs and functionalities.

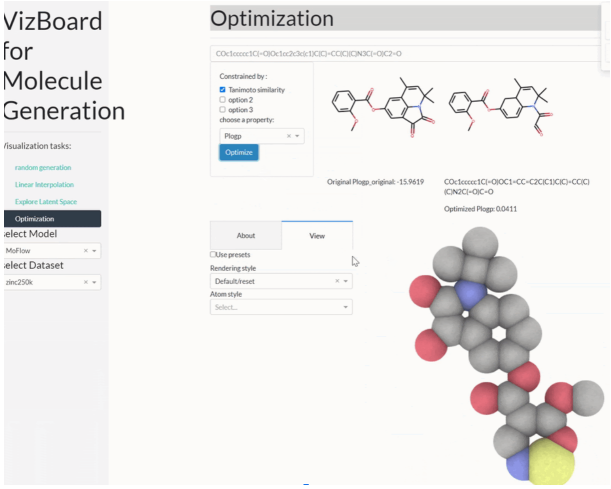


Figure 6: Visualization of the optimized molecules under different objective properties.

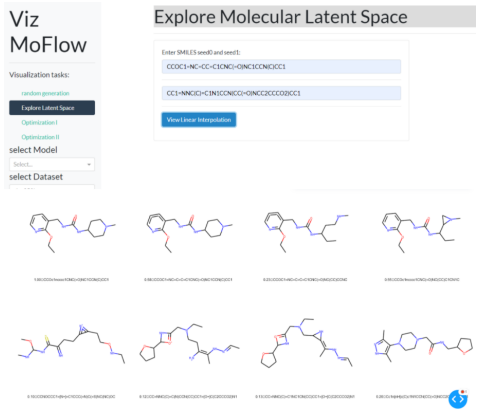


Figure 7: Visualization of the linear interpolation between two molecules to show their changing trajectory in the latent space.

4.2 Target-Specific Molecule Generation

Virtual screening Result for SARS-CoV2 3CL Protease. Here we present our initial experiments using the molecules generated from the MoFlow model as lists of drugs to pair with the target protein - SARS-CoV2 3CL Protease.

As shown in Figure 10, we output a table that lists the ranking of the drug candidates by the predicted binding affinity score.

Figure 11 displays the virtual screening results in detail. Also, we can find the graph structure and SMILEs of the corresponding drug molecules

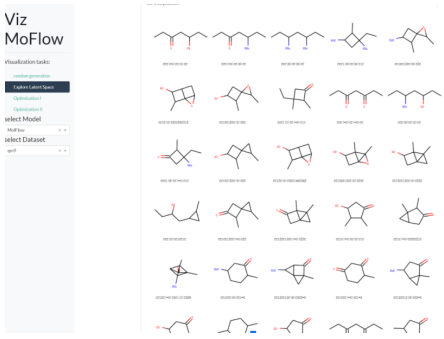


Figure 8: Visualization of the chemical similarity between each neighboring molecule and the centering molecule

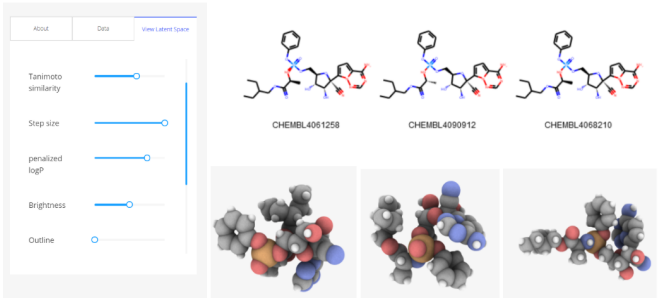


Figure 9: Demo of 2D and 3D interactive visualizations for similar chemical structures in the latent space

0	| Rank | Drug Name | Target Name | Binding Sco...
1	+------+----------+------------+------------...
2	| 1 | Drug 15 | Target 15 | 92.27 ...
3	| 2 | Drug 18 | Target 18 | 92.27 ...
4	| 3 | Drug 24 | Target 24 | 92.27 ...
5	| 4 | Drug 35 | Target 35 | 92.27 ...
6	| 5 | Drug 39 | Target 39 | 92.27 ...
7	| 6 | Drug 10 | Target 10 | 7.41 ...
8	| 7 | Drug 36 | Target 36 | 7.31 ...
9	| 8 | Drug 13 | Target 13 | 6.91 ...
10	| 9 | Drug 21 | Target 21 | 6.79 ...
11	| 10 | Drug 20 | Target 20 | 6.49 ...
12	| 11 | Drug 23 | Target 23 | 6.26 ...
13	| 12 | Drug 27 | Target 27 | 6.26 ...

Figure 10: Ranking of the drug candidates by the predicted binding affinity score

5 DISCUSSION AND FUTURE WORK

We found our approach - which uses Dash / Dash bio - to be ideal for the drug discovery purpose. First, it ties modern UI elements like dropdowns, sliders, and graphs directly to the analytical Python code. Dash isn't just for dashboards. Users will have full control over the look and feel of the applications. The 3D view feature from Dash bio can illustrate the shapes of proteins and provide insights into the way that they bind to other molecules. We can change the level of atom outlines, ambient occlusion, and more with the various viewing parameters. Further, our approach allows for the scroll wheel to control zoom for the molecule. Dash fires

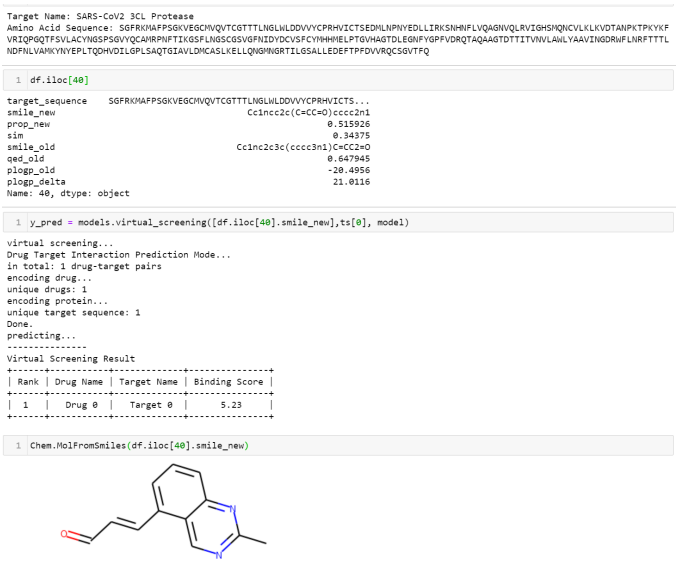


Figure 11: Virtual screening results in detail

Python callback functions when you click on an atom, rotate the molecule, or change the structure. Also, you can highlight individual atoms (like a protein's active site).[10] These properties render our approach especially valuable when communicating the mechanics of biomolecular processes.

We are still in the process of searching for a strong DBTA predictor. We plan to conduct many more experiments using various pre-trained DBTA prediction models from DeepPurpose - as well as from other DBTA prediction frameworks. To achieve better results, we would like to train our own DBTA predictor if it's necessary using the BindingDB dataset.

Furthermore, we need to add an evaluation metric and selection scheme, such as Docking analysis with Target Structure, to filter the generated molecules with the most desired properties.

6 CONCLUSION

Deep generative models have demonstrated strong potentials on efficient and effective drug molecule design with desired properties. However, such complex deep learning models for drug discovery are hard to train and hard to understand. In order to provide meaningful solutions to the drug discovery challenge, it is important to have robust data visualization tools for the generative models. Our approach holds the promise for researchers to explore and build on the deep learning models' results (with less effort) and thereby enable them to focus on evaluation, error analysis and developing valuable insights from the visualizations.

We believe that our end-to-end de novo drug design approach to generate novel molecules with high binding affinity to a specific target protein will provide a timely, efficient and adaptive approach to generate novel drug-candidate especially for new unseen target proteins. Our initial explorations on these tasks offer some proof-of-concepts for AI-based approaches to tackle drug target interaction prediction and novel drug design in a simultaneous process.

REFERENCES

[1] [n.d.]. Speck Examples and Reference. https://dash.plotly.com/dash-bio/speck

[2] BORA BERAN. 2015. *Going 3D with Tableau.* https://boraberan.wordpress.com/2015/12/18/going-3d-with-tableau/

[3] Prafulla Dhariwal and Durk Kingma. 2019. *Glow: Better Reversible Generative Models.* https://openai.com/blog/glow/

[4] evgerher and sarmbruster. 2019. *RDKit-Neo4j project.* https://github.com/rdkit/neo4j-rdkit

[5] Riley PF Vinyals O Dahl GE Gilmer J, Schoenholz SS. 2017. eural Message Passing for Quantum Chemistry. In: International Conference on Machine Learning. *p. 1263–1272. Available from: http://proceedings.mlr.press/v70/gilmer17a.html.* (2017).

[6] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* 4, 2 (2018), 268–276.

[7] Heidemeyer M. Ban F. Cherkasov A. He, T. and M. Ester. 2017. SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *J. Cheminform. 9:24. doi: 10.1186/s13321-017-0209-z* (2017).

[8] Kexin Huang, Tianfan Fu, Lucas Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. 2020. DeepPurpose: a deep learning library for drug-target interaction prediction and applications to repurposing and screening. *arXiv:2004.08919* (2020).

[9] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364* (2018).

[10] MODERN.DATA. 2019. *Dash has gone full R.* r-craft.org/r-news/dash-has-gone-full-r/

[11] Chengxi Zang and Fei Wang. 2020. MoFlow: An Invertible Flow Model for Generating Molecular Graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*