DE NOVO DESIGN OF PROTEIN TARGET SPECIFIC SCAFFOLD-BASED INHIBITORS VIA REINFORCEMENT LEARNING

Andrew D. McNaughton, Mridula S. Bontha, Carter R. Knutson, Jenna A. Pope, & Neeraj Kumar
Pacific Northwest National Laboratory
Richland, WA 99354, USA
{neeraj.kumar}@pnnl.gov

ABSTRACT

Efficient design and discovery of target-driven molecules is a critical step in facilitating lead optimization in drug discovery. Current approaches to develop molecules for a target protein are intuition-driven, hampered by slow iterative design-test cycles due to computational challenges in utilizing 3D structural data, and ultimately limited by the expertise of the chemist – leading to bottlenecks in molecular design. In this contribution, we propose a novel framework, called 3D-MolGNN$_{RL}$, coupling reinforcement learning (RL) to a deep generative model based on 3D-Scaffold to generate target candidates specific to a protein building up atom by atom from the starting core scaffold. 3D-MolGNN$_{RL}$ provides an efficient way to optimize key features by multi-objective reward function within a protein pocket using parallel graph neural network models. The agent learns to build molecules in 3D space while optimizing the activity, binding affinity, potency, and synthetic accessibility of the candidates generated for infectious disease protein targets. Our approach can serve as an interpretable artificial intelligence (AI) tool for lead optimization with optimized activity, potency, and biophysical properties.

1 INTRODUCTION

Recent advancements in machine learning (ML) and artificial intelligence (AI) have demonstrated the potential to revolutionize the drug design process by reducing the initial chemical search space in the early stages of discovery (You et al., 2018; Born et al., 2020; Li et al., 2018; Huang et al., 2020; Chenthamarakshan et al., 2020). Currently, the potential chemical space is composed of $> 10^{60}$ molecules, candidates with suitable activity against specific protein targets only narrows the search space significantly based on the critical fragments. The COVID-19 pandemic has brought a surge of interest to explore data-driven methods to better produce efficacious drug candidates (Huang et al., 2020). One of the most important factors in identifying new drug candidates is that the molecules possess optimized properties that allow them to effectively bind with a required target while also having minimal off-target effects. If we consider molecular generation as a controlled and dynamic step-by-step process, it is possible to produce end products that possess these optimized properties. This approach allows us to formulate *de novo* drug design as a reinforcement learning (RL) problem and utilize algorithms that best learn a molecule's representation space based on the core moiety and its spatial interaction with the protein target. As an alternative, RL would provide a platform to create a highly-efficient inverse molecular design AI-system capable of producing novel high-performance molecules with domain-targeted properties.

1.1 RELATED WORKS

Our current work presents a novel approach to address the problem of generating molecules optimized for specific 3D protein targets starting from core functional groups so called scaffold. In recent years, RL-based works to generate molecules have been introduced that attempt to tackle the problem in different ways. A method by Popova et al. (2018) utilizes a fragment-replacement-based approach to optimize existing SMILES strings for specific drug-like properties, while another method by

Ståhl et al. (2019) generates entirely new SMILES. Several current RL methods rely on the use of variational autoencoders (VAEs), which learn the latent space representation of 2D molecules to find the best representation (Lim et al., 2018; Jin et al., 2018; Liu et al., 2018; Sattarov et al., 2019; Joo et al., 2020). However, none of these methods take into consideration the 3D protein structure during the generation phase. Some of these methods do test the drug-likeness post-generation against some given target protein, but the target is never directly involved in the RL loop. When designing drug candidates that target specific proteins, learning how the molecule interacts with the protein's structure is invaluable to the generative process and can greatly assist in accelerating the automation of medicinal chemistry.

Recently, Born et al. (2020) proposed a deep reinforcement learning framework called PaccMann^{RL} for designing antiviral candidates binding against given protein targets using 2D protein sequence data. The generative model is composed of two pre-trained VAEs, namely a protein-VAE and a SELFIES-VAE, for mapping protein and drug molecules to a multi-modal latent space. The critic model is composed of two predictive models for predicting binding affinity of the protein-ligand pair and toxicity for the ligand. These predictions are used to formulate a multi-modal reward function to penalize the generative model (Born et al., 2021). This method still uses VAEs to generate 2D molecules, but it combines the latent space of the SMILES with the embedding of a specific protein to generate SMILES that consider target protein information

1.2 Problem Formulations and Proposed Method

We propose a new method, known as 3D-MolGNN$_{RL}$, that not only incorporates the protein structure into the RL loop, but considers the 3D structures of the generated compounds by constructing them atom-by-atom in 3D space. To the best of our knowledge, this is a process that hasn't been explored as most methods simply rely on 2D representations of the molecules in the form of a SMILES string. To overcome the sequence-based representation of the target and the drug, we leverage a 3D-scaffold-based molecular generation actor over a 2D SMILES generation actor. The 3D-MolGNN$_{RL}$ method uses our previously formulated 3D-scaffold-based generative model Joshi et al. (2021) to make atom-wise placements on the molecule starting from a desired scaffold.

Our method considers a similar approach to that of the actor-critic method demonstrated in the PaccMann^{RL} model. In our approach, we utilize a scaffold-based generative model, 3D-Scaffold (Joshi et al., 2021) to produce valid 3D compounds as the actor model. For the critic model, we utilize parallel graph neural networks as a binding probability predictor (GNN$_P$ (Knutson et al., 2021)) to evaluate whether the generated compound actively binds with a target protein. This method takes the 3D structure of the protein pocket and the ligand, and predicts the probability of their interaction without any prior knowledge of the intermolecular interactions. The GNN$_P$ model uses both the residue and 3D atomic-level representation of a protein as well as the 3D atomic and bond level representation of the molecule.

2 Experiments and Dataset

To demonstrate our methods ability to produce molecules for a given target protein, this work was centered around creating drug-like molecules that interacted with the SARS-CoV-2 M_{pro}. To train the model, We used a dataset of non-covalent inhibitors from the BindingDB dataset (Gilson et al., 2016) and FDA approved drugs FDA. These compounds were filtered based on biophysical and cheminformatics properties like IC_{50}, molecular weight, atom type, and functionality and have almost 10K unique scaffolds. To represent the definition of scaffold, we used Murcko scaffolds Bemis & Murcko (1996). A similar dataset was used in the prior 3D-Scaffold model (Joshi et al., 2021). High-throughput experimentation and our computer aided drug design results suggest that compounds containing the functional group piperazine are potent and have a higher affinity towards the M_{pro} target (Clyde et al., 2021; Joshi et al., 2021). However, there were no piperazine-containing molecules in the initial dataset used for training the 3D-scaffold model, but we were still able to generate some molecules containing piperazine using the trained model. We curated a smaller subset of the BindingDB dataset that possessed affinity towards protease-like targets and combined them with the piperazine dataset. Finally, we filtered these compounds based on their ability to bind with M_{pro} by doing an initial pass through the GNN$_P$ model. Since our motive is to achieve better binding affinity, potency and easily synthesizable compounds for the M_{pro} target, the initial screening helped us to

choose proper compounds for training the RL models so that they learned to produce similar, or better compounds.

3 METHODS

In this section, we discuss in detail the architecture and implementation of the RL approach. In particular,we go into detail about how the critic interprets the output from the actor.

3.1 3D-MOLGNN$_{RL}$ FRAMEWORK

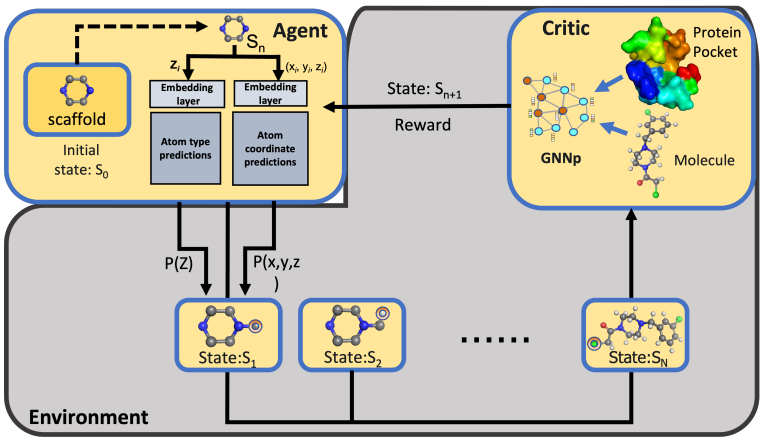


Figure 1: Schematic RL workflow highlighting the interaction of Actor and Critic in our 3D-MolGNN$_{RL}$ model. The agent starts building the molecule from a starting scaffold (state S_0) and subsequently builds the molecule by choosing an atom based on the reward assigned by the critic for that intermediate state. The Agent block serves as the Actor, generating molecules to pass into the Critic block to assess the performance.

The schematic representation of 3D-MolGNN$_{RL}$ framework is shown in the figure 1. The RL workflow begins from training a 3D-scaffold molecule generator followed by optimizing the partially built candidates towards the target of interest at every step of training. The neural network used in the 3D-Scaffold framework for *de-novo* drug candidate design can be broken into two major blocks: feature learning and atom placement as shown in Figure 1. In the feature learning block, the embedding and interaction layers of SchNet (Joshi et al., 2021; Schütt et al., 2019; Schütt et al., 2017; Schütt et al., 2017; Schütt et al., 2018) are used to extract and update rotationally and translationally invariant atom-wise features that capture the chemical environment of an unfinished molecule. The extracted features are used to predict distributions for the type of next atom and its 3D coordinates, where the latter distribution is constructed from predictions of pairwise distances between the next atom and all preceding atoms. The whole procedure is repeated successively to build a complete molecule with the desired scaffold. The partial molecule associated with each step of this atom placement is assessed by the critic (GNN$_P$). As a part of our experimentation, we used two critics simultaneously to use the binding probability and binding affinity of the partial molecule and the target of interest along with the synthetic accessibility of the partial molecule. The binding probability is the outcome from the SoftMax layer which predicts the activity/inactivity of a protein-ligand complex. The binding affinity is measured in terms of K_i and K_d, which refer to the inhibition and dissociation constants, respectively and is given as $-log(\frac{K_i}{K_d})$.The two critics essentially use the same feature representation, but differ in terms of the data they have been trained on and also the label associated with the data. A more detailed breakdown of the layers and hyperparameters associated with the actor and critic models is included in the Appendix.

3.2 REWARD FUNCTION & PARAMETERS

The process of building the molecule starts from a desired scaffold associated with the selected core functionality. For every action *t* which is a random selection and placement of the atom to the partial molecule, the 3D-scaffold model predicts the next possible atom that could be placed close to the center of mass to the partial molecule at step $t-1$ and transitions to state s_t. At any step, let Z_{next} be the ground truth type of the next atom and $\hat{p}_{\text{type}}^{Z_{\text{next}}}$ the probability that the model assigns to that type at the current step. This probability is converted to a negative log-likelihood as $P_\Theta(s_t)^{type}$ (equation 3a). Here, $\hat{p}_j^b$ is the probability that the model assigns for the distance between position of already placed atom and ground truth of the next atom to fall into distance bin $b \in B$ at the current step. The distance based probability is calculated using Gaussian expanded ground truth distances q_j^b and probability distribution of atom placement $\hat{p}_j^b$ s shown in equation 3b. The overall idea is to train the agent to learn the latent space representation of atom type and it's possible placement in 3D-space closest to the center of mass of the molecule generated so far. The overall probability for any action *t* is the summation of type-probability and distance-probability (equation 2). The ultimate goal is to train the agent such that it learns to generate new compounds while optimizing the policy $\Pi(\Theta)$ (equation 1). The policy is defined as the difference between the action-probabilities and the reward assigned by the critic at that step.

$$\Pi(\Theta) = \sum_{s_t \in S} (P_\Theta(s_t) - R(s_t)) \tag{1}$$

$$P_\Theta(s_t) = P_\Theta(s_t)^{type} + P_\Theta(s_t)^{dist} \tag{2}$$

$$P_\Theta(s_t)^{type} = -\log\left(\hat{p}_{\text{type}}^{Z_{\text{next}}}\right) \quad \text{(3a)} \qquad P_\Theta(s_t)^{dist} = \sum_{j=1}^{N} \sum_{b \in B} q_j^b \log\left(\hat{p}_j^b\right) \quad \text{(3b)}$$

$$R_1(s_t) = \alpha.C_{BP}(B, C_t) + \beta.C_{EA}(B, C_t) + (1 - \gamma.C_{SA}(C_t)) \tag{4}$$

$$R_2(s_t) = \alpha.C_{BP}(B, C_t) + (1 - \beta.C_{SA}(C_t)) \tag{5}$$

We used two different reward functions as $R_1(s_t)$ and $R_2(s_t)$ associating with two independent experiments for the purposes of ablation/hyperparameter testing. By removing portions of the reward function, we can see which attributes provide the best reward. The first reward is a function of binding probability, binding affinity of the target and the partially built molecule in addition to the synthetic accessibility of the molecule. While the second reward function uses only the binding probability and the synthetic accessibility score. We assigned different weights to each of these components while calculating the rewards. For $R_1(s_t)$ (equation 5), we used 0.5,0.25 and 0.25 respectively for binding probability, binding affinity and synthetic accessibility. While for $R_2(s_t)$ (equation 4, we used 0.75 and 0.25 for binding probability and synthetic accessibility respectively. We trained the agent for 150 epochs and chose the best trained model to generate compounds.The Synthetic Accessibility and Binding Affinity scores are scaled to be in between 0 and 1 to match the scale of the binding probability. For more on the ablation study, including some reward functions not shown in the main text, see Appendix C.

4 RESULTS & DISCUSSION

4.1 MODEL PERFORMANCE

To ensure that we are a learning to build molecules based on the training data, we visualize the training and validation error over each epoch. Figure 2 represents the aggregated loss over the 150 epochs for our 3D-MolGNN$_{RL}$ model. As loss decreases over time, we know that our model is able to effectively learn from the training data. This result is strictly evaluating the training and validation loss through the actor model to evaluate training performance, the drug-likeness scoring of the GNN model is discussed further in the next section.

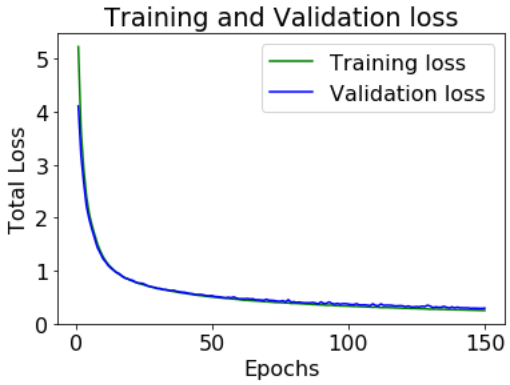


Figure 2: Plot of loss over total epochs, the loss represents how well the model learns the 3D trace of the input data. Loss is calculated as a combination of KL-Divergence and the Reward Function

4.2 Drug-likeness Metrics for Generated Compounds

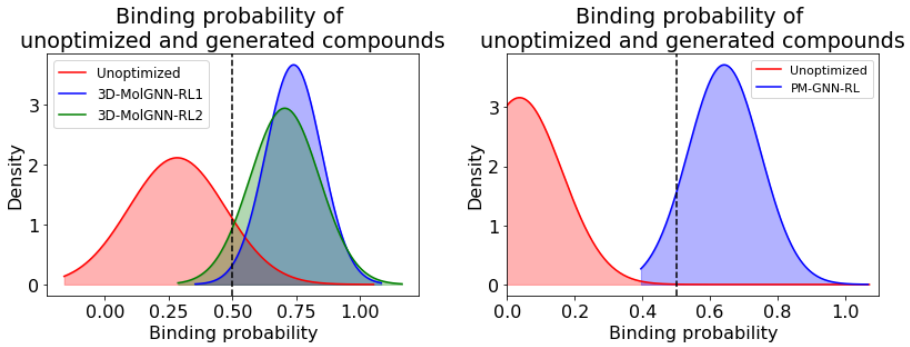
We examined the effectiveness of our RL methods using different drug-likeliness metrics: quantitative estimate of drug-likeness (QED), water solubility (logS), synthetic accessability (SA), and the octanol-water partition coefficient (logP). See the appendix for a full explanation of each metric. A desirable drug candidate would score well in each of these metrics. Since the agent for the 3D-MolGNN$_{RL}$ is a scaffold based generative model, we focused on generating compounds with piperazine as the scaffold. To ensure that only valid molecules are compared, we filtered the total list of generated compounds based on a modified Lipinksi rule. The rule suggests that for a candidate to be an acceptable drug-like compound, there should be no more than 5 hydrogen bond donors, no more than 10 hydrogen bond acceptors, no more than 5 rotatable bonds, a molecular mass between 200 and 500 Dalton, an octanol-water partition coefficient less than 5, and finally at least one aromatic ring in the structure. Once filtered, approximately 100 top compounds per method were obtained.

4.2.1 Comparison to 2D Generative Model

To show the advantages of using 3D-based generative modeling over conventional 2D-based generative models (SMILES-based), we compare our 3D-MolGNN$_{RL}$ model to that of PaccMann$_{RL}$. For better comparison of the two methods, we utilize a slightly modified version of the architecture to include our GNN critic model instead of the default PaccMann critic. This allows us to have a metric of comparison between a 2D and 3D generative model. For purposes of discussion, the modified PaccMann$_{RL}$ model will be referred to as PM-GNN$_{RL}$.

We first wanted to demonstrate that the RL implementation of the methods were essential in producing molecules with a high binding probability. To accomplish this, we generated molecules for the protein target M_{pro} (PDB:6WQF), both with and without passing the reward from the critic back to the agent in each method. We designated here the molecules generated without RL optimization as *unoptimized*. To fairly compare the optimized vs unoptimized methods, we ensured that the unoptimized models were trained and tested on the same datasets used for training the respective RL models. The comparison of binding probabilities of the generated compounds towards the M_{pro} target from the 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ is shown in figure 3a and 3b, respectively. From this figure, it is evident that the RL mechanism has improved both agent's performance in producing candidates with high binding probability. The unoptimized molecules generated by the agent in 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ show very low binding probability towards M_{pro} as the predictions have a low mean probability. On the other hand, when using biased compounds during RL optimization, we see a significant increase in high binding probability compounds, showing the effects of incorporating RL with the generative models.

To further assess the novelty and properties of generated compounds, we looked at the four different drug-likeness properties listed above to serve as comparison metrics. Figure 4 compares these metrics for both 3D-MolGNN$_{RL}$ reward functions with known active molecules for M_{pro}. A full description of each metric is given in the appendix (section A). The first metric, QED, represents a quantification



(a) 3D-MolGNN-RL optimized vs unoptimized (b) PM-GNN-RL optimized vs unoptimized

Figure 3: Comparison of generated compounds before and after optimization by 3D-MolGNN$_{RL}$. The unoptimized molecules were generated using the respective agents for each method over the same data but without involvement of the critic. Results visualized as the normalized probability density function of the data

of the desirability of the drug (Bickerton et al., 2012). We can see that in Figure 4a, the RL methods all produce higher scoring molecules than the current experimentally determined M$_{\text{pro}}$ actives with the mean of the PM-GNN$_{RL}$ having a slightly higher increase in mean.

We next look at the distributions of water solubility calculated using the ESOL method (Delaney, 2004). Figure 4b shows that our methods score better than the current M$_{\text{pro}}$ actives while also showing that these methods produce molecules with desirable solubility in water. The mean of the 3D-MolGNN$_{RL}$'s $R_1(s,t)$ model has the highest improvement relative to the mean of the M$_{\text{pro}}$ actives while both the $R_2(s,t)$ and PM-GNN$_{RL}$ models improve less.

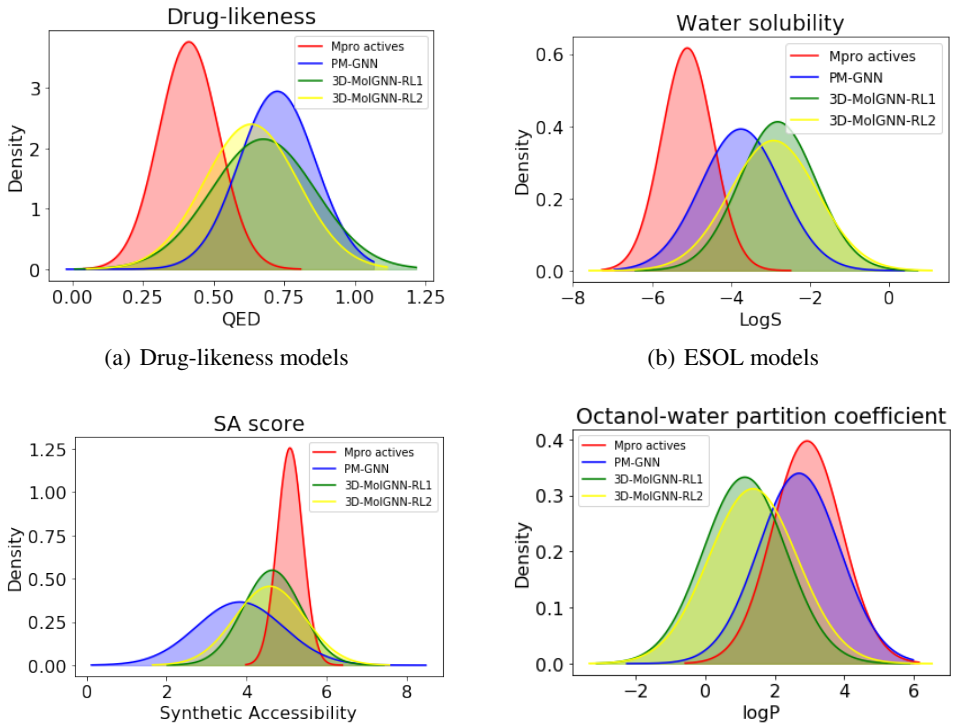
The next metric, synthetic accessibility (SA), is a term included in each of the reward functions of the RL methods tested. Figure 4c illustrates the distributions obtained by each method where a SA score closer to 1 is preferred. The two 3D-MolGNN$_{RL}$ models have a consistent range of SA scores that center around 5, which indicates consistently above average scores. $R_1(s,t)$ shows an 8.78% improvement and $R_2(s,t)$ shows a 9.92% improvement. The PM-GNN$_{RL}$ spans a much wider range of SA scores, but the mean is centered around 4. These three models only slightly edge out the existing M$_{\text{pro}}$ actives.

The last metric is the octanol-water partition coefficient, or more simply known as log$P$. Looking at Figure 4d, we can see how each method produces molecules spanning the range of -2 to 6. The 3D-MolGNN$_{RL}$ model using the $R_1(s,t)$ reward yields a 61.21% improvement to the M$_{\text{pro}}$ actives in logP value, whereas using the $R_2(s,t)$ reward yields a 53.08% improvement. In this metric, the PM-GNN$_{RL}$ metric produces compounds very similar to the M$_{\text{pro}}$ actives.

Table 1: Details about the drug-likeness metrics proposed in this work. We look at the 3D-MolGNN$_{RL}$ model's two separate reward functions $R_1(s,t)$ and $R_2(s,t)$. For each method, the top 3 candidates are chosen and their metric scores listed.

	QED			**logS**			**SA**			**logP**		
Rank	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
3D-MolGNN$_{RL}$ R1	0.49	0.59	0.41	-3.32	-2.49	-2.23	4.81	4.81	4.59	1.56	0.82	-0.14
3D-MolGNN$_{RL}$ R2	0.32	0.44	0.82	-2.53	-3.82	-2.90	3.55	4.84	3.93	0.71	2.29	1.68
PM-GNN$_{RL}$	0.45	0.39	0.74	-3.14	-3.90	-3.31	2.86	2.22	2.27	1.53	2.70	1.78

To compare how each method performed on a per-molecule basis, Table 1 gives the results for the top 3 candidates produced by each of the methods. These molecules were ranked based on their predicted binding probability with the M$_{\text{pro}}$ target. We can see that molecules that are performing above average in one metric are not guaranteed to perform well in every metric. For example, the top candidates for each method are each above the mean SA for the given method, but produce below



(a) Drug-likeness models

(b) ESOL models

(c) Synthetic accessibility score models

(d) Partition coefficient models

Figure 4: Comparison of the properties of the molecules produced from 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ against the experimentally identified active compounds for M_{pro} target. Results visualized as the normalized probability density function of the data

the mean for QED. 2D and 3D snapshots of the top 3 candidates from each method are available in appendix section B.

Since our goal is achieve *de novo* design, additional metrics such as validity, uniqueness, and novelty were considered among the piperazine-based generated compounds as shown in Table 2. Initial validity was determined by processing the molecules using cheminformatics tools. The methods of calculating these properties are described in further detail in appendix subsection A.5. From table 2, we found that 3D-MolGNN$_{RL}$ R1 and 3D-MolGNN$_{RL}$ R2 produced greater than 95% valid compounds while also scoring close to 100% in uniqueness and novely from this valid set. The PM-GNN$_{RL}$ model, while not scoring quite as well in Uniqueness and Novelty, scores the highest in Validity. This shows that by incorporating more parameters into the multi-objective reward function, there is an improvement in the generation of novel drug candidates. Overall, 3D-MolGNN$_{RL}$ R1 outperformed other methods by generating compounds with high Validity, Uniqueness and Novelty.

Table 2: Table outlining the three metrics used to evaluate the compounds produced by varying reward functions and RL models. Compounds were screened and scored for an overall percentage based on validity, uniqueness, and novelty.

Model	Validity	Uniqueness	Novelty
3D-MolGNN$_{RL}$ R1	99.9%	100%	99.9%
3D-MolGNN$_{RL}$ R2	97%	99.9%	99.9%
PM-GNN$_{RL}$	100%	93%	93%

5 CONCLUSIONS

In this work, we introduced a new method to include both the 3D structure of protein target and the generated compounds to perform multi-objective lead optimization critical for drug design and discovery. We demonstrated that our novel framework 3D-MolGNN$_{RL}$, which couples RL to a deep generative model based on a 3D-Scaffold, can generate target candidates built up atom by atom that are specific to a protein pocket. 3D-MolGNN$_{RL}$ provides an efficient way to generate target specific candidates by learning to build molecules in 3D space while optimizing the binding affinity, potency, and synthetic accessibility. To accomplish this, we utilized the protein for SARS-CoV-2 M_{pro} as a target for generating optimized inhibitor candidates. We found that our model was able to generate molecules with better druglikeness, synthetic accessibility, water solubility, and hyrdophilicity than current M_{pro} active molecules. This was given by a $>$50% increase in QED, a $>$40% increase in solubility, a $>$8% improvement in SA, and a $>$50% improvement in hydrophilicity. We found that our RL integration significantly improved the types of molecules generated by the untrained agents. Throughout this work, we demonstrated that by including more parameters into the multi-objective reward function, there is an improvement in generated novel target specific candidates. This gives us confidence that our RL framework is effective at producing protein target specific hit candidates by leveraging the 3D structures of both the generated molecule and the protein pocket, a consideration not made by other molecular generation methods to date.

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Department of Energy, Office of Science, Laboratory Directed Research Funding (LDRD), Mathematics of Artificial Reasoning for Science (MARS) Initiative, at the Pacific Northwest National Laboratory. Pacific Northwest National Laboratory (PNNL) is a multiprogram national laboratory operated by Battelle for the DOE under Contract DE-AC05-76RLO 1830. This research used computational resources provided by Research Computing at the Pacific Northwest National Laboratory.

REFERENCES

Zinc database. URL http://zinc.docking.org/substances/subsets/fda/?page=1.

Guy W. Bemis and Mark A. Murcko. The properties of known drugs. 1. molecular frameworks. *J. Med. Chem.*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928. URL https://doi.org/10.1021/jm9602928.

G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, feb 2012. ISSN 1755-4330. doi: 10.1038/nchem.1243. URL http://www.nature.com/articles/nchem.1243.

Jannis Born, Matteo Manica, Joris Cadow, Greta Markert, Nil Adell Mill, Modestas Filipavicius, and María Rodríguez Martínez. Paccmann^{RL} on sars-cov-2: Designing antiviral candidates with conditional generative models. *arXiv preprint arXiv:2005.13285*, 2020.

Jannis Born, Matteo Manica, Joris Cadow, Greta Markert, Nil Adell Mill, Modestas Filipavicius, Nikita Janakarajan, Antonio Cardinale, Teodoro Laino, and María Rodríguez Martínez. Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2. *Machine Learning: Science and Technology*, 2(2):025024, jun 2021. ISSN 2632-2153. doi: 10.1088/2632-2153/abe808. URL https://iopscience.iop.org/article/10.1088/2632-2153/abe808.

Vijil Chenthamarakshan, Payel Das, Inkit Padhi, Hendrik Strobelt, Kar Wai Lim, Ben Hoover, Samuel C Hoffman, and Aleksandra Mojsilovic. Target-specific and selective drug design for covid-19 using deep generative models. *arXiv preprint arXiv:2004.01215*, 2020.

Austin Clyde, Stephanie Galanie, Daniel W Kneller, Heng Ma, Yadu Babuji, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, et al. High-throughput virtual screening and

validation of a sars-cov-2 main protease noncovalent inhibitor. *Journal of chemical information and modeling*, 2021.

John S. Delaney. ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, may 2004. ISSN 0095-2338. doi: 10.1021/ci034243x. URL https://pubs.acs.org/doi/10.1021/ci034243x.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, dec 2009. ISSN 1758-2946. doi: 10.1186/1758-2946-1-8. URL https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-1-8.

Michael K. Gilson, Tiqing Liu, Michael Baitaluk, George Nicola, Linda Hwang, and Jenny Chong. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research*, 44(D1):D1045–D1053, jan 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1072. URL https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkv1072.

Kexin Huang, Tianfan Fu, Cao Xiao, Lucas Glass, and Jimeng Sun. Deeppurpose: a deep learning based drug repurposing toolkit. *arXiv preprint arXiv:2004.08919*, 2020.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. feb 2018. URL http://arxiv.org/abs/1802.04364.

Sunghoon Joo, Min Soo Kim, Jaeho Yang, and Jeahyun Park. Generative Model for Proposing Drug Candidates Satisfying Anticancer Properties Using a Conditional Variational Autoencoder. *ACS Omega*, 5(30):18642–18650, aug 2020. ISSN 2470-1343. doi: 10.1021/acsomega.0c01149. URL https://pubs.acs.org/doi/10.1021/acsomega.0c01149.

Rajendra Prashad Joshi, Niklas Gebauer, Neeraj Kumar, and Mridula Bontha. 3d-scaffold: Deep learning framework to generate 3d coordinates of drug-like molecules with desired scaffolds. *bioRxiv*, 2021.

Carter Knutson, Mridula Bontha, Jenna A Bilbrey, and Neeraj Kumar. Decoding the protein-ligand interactions using parallel graph neural networks. *arXiv preprint arXiv:2111.15144*, 2021.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1):33, 2018. ISSN 1758-2946. doi: 10.1186/s13321-018-0287-6. URL https://doi.org/10.1186/s13321-018-0287-6.

Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of Cheminformatics*, 10(1):31, dec 2018. ISSN 1758-2946. doi: 10.1186/s13321-018-0286-7. URL https://jcheminf.biomedcentral.com/articles/10.1186/s13321-018-0286-7.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. Constrained graph variational autoencoders for molecule design. In *NeurIPS*, pp. 7806–7815, 2018.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7):eaap7885, jul 2018. ISSN 2375-2548. doi: 10.1126/sciadv.aap7885. URL https://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aap7885.

Boris Sattarov, Igor I. Baskin, Dragos Horvath, Gilles Marcou, Esben Jannik Bjerrum, and Alexandre Varnek. De Novo Molecular Design by Combining Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *Journal of Chemical Information and Modeling*, 59 (3):1182–1196, mar 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00751. URL https://pubs.acs.org/doi/10.1021/acs.jcim.8b00751.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Adv. Neural Inf. Process Syst.*, volume 30, pp. 991–1001. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.*, 8(1):13890, 2017. ISSN 2041-1723. doi: 10.1038/ncomms13890. URL https://doi.org/10.1038/ncomms13890.

K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. Schnet – a deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, 2018. doi: 10.1063/1.5019779. URL https://doi.org/10.1063/1.5019779.

K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller. Schnetpack: A deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.*, 15(1):448–455, 2019. doi: 10.1021/acs.jctc.8b00908. URL https://doi.org/10.1021/acs.jctc.8b00908.

Niclas Ståhl, Göran Falkman, Alexander Karlsson, Gunnar Mathiason, and Jonas Boström. Deep Reinforcement Learning for Multiparameter Optimization in de novo Drug Design. *Journal of Chemical Information and Modeling*, 59(7):3166–3176, jul 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.9b00325. URL https://pubs.acs.org/doi/10.1021/acs.jcim.9b00325.

Scott A. Wildman and Gordon M. Crippen. Prediction of Physicochemical Parameters by Atomic Contributions. *Journal of Chemical Information and Computer Sciences*, 39(5):868–873, sep 1999. ISSN 0095-2338. doi: 10.1021/ci990307l. URL https://pubs.acs.org/doi/10.1021/ci990307l.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, pp. 6410–6421, 2018.

A PERFORMANCE METRICS

A.1 QUANTITATIVE ESTIMATE OF DRUGLIKENESS

The QED metric represents a quantification of the desirability of the drug (Bickerton et al., 2012). The closer the score is to 1, the more desirable it is as a drug candidate.

The equation for QED is given as:

$$QED = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln d_i\right),$$

Where d_i is a series of desirability functions (d) belonging to eight widely used molecular descriptors. These are molecular weight (MW), the octanol-water partition coefficient (ALOGP), the number of hydrogen bond donors (HBD), the number of hydrogen bond acceptors (HBA), the molecular polar surface area (PSA), the number of rotatable bonds (ROTB), the number of aromatic rings (AROM), and the number of structural alerts (ALERTS).

The desirability function can be represented by a general asymmetric double sigmoidal function where d(x) is the desirability function for molecular descriptor x shown as:

$$d_i(x) = a_i + \frac{b_i}{1+\exp\left(-\frac{x-c_i+\frac{d_i}{2}}{e_i}\right)} \cdot \left[1 - \frac{1}{1+\exp\left(-\frac{x-c_i+\frac{d_i}{2}}{e_i}\right)}\right],$$

where $a_i, ..., f_i$ can be found in the supplementary table of the original publication (Bickerton et al., 2012).

A.2 ESTIMATING AQUEOUS SOLUBILITY

The metric, water solubility, calculates the log solubility (log$S$) of the molecule. In this work, the solubility is determined by ESOL (Delaney, 2004). The majority of drugs posses a log$S$ between -8 and -2. The more positive the value, the more water soluble the molecule.

ESOL as defined in Delaney (2004) can be calculate as the multiple linear regression of:

1. clogP
2. Molecular weight (MWT)
3. Rotatable bonds (RB)
4. Aromatic proportion (AP)

given as:

$$Log(S_w) = 0.16 - 0.63clogP - 0.0062MWT + 0.066RB - 0.74AP$$

A.3 SYNTHETIC ACCESSIBILITY

SA is one of the most critical metrics to use in determining the simplicity in experimentally synthesizing a molecule. It is not a score that dictates how effective the molecule is, but rather a practical measure of it's complexity. The SA score is between 1 to 10, where 1 indicates an easily synthesizable molecule and 10 indicates a complex one.

The algorithm for calculating the SA score of a molecule (as represented in Ertl & Schuffenhauer (2009)) is given as:

$$SAscore = fragmentScore - complexityPenalty,$$

where the fragment score is calculated as a sum of contributions of all fragments in the molecule divided by the number of fragments in this molecule. The contribution of a fragment is obtained from a database of fragment contributions that were generated by statistical analysis of substructures in the PubChem collection.

The complexity penalty is a score given to characterize the presence of complex structural features in the molecules. It is defined as a combination of the following:

$$ringComplexityScore = log(nRingBridgeAtoms + 1) + log(nSpiroAtoms + 1)$$
$$stereoComplexityScore = log(nStereoCenters + 1)$$
$$macrocyclePenalty = log(nMacrocycles + 1)$$
$$sizePenalty = nAtoms^{1.005} - natoms$$

A.4 Hydrophilicity vs. Lipophilicity

This metric, known as log$P$, is the calculated octanol-water partition coefficient of a given molecule. The values represents if a drug is either very hydrophilic (-3) or very lipophilic (+10). This specific metric is present in Lipinski's rule as value that needs to be less than 5 to be considered a drug candidate.

To calculate the partition function for octanol-water partition coefficient that dictates whether a molecule is more hydrophilic or lipophilic we utilize the RDKit package implementation of the Crippen approach (Wildman & Crippen, 1999). It simply calculates the sum of the contributions of each of the atoms in the molecule. Intramolecular interactions are accounted for my classifying atoms into different types based on their attached a_i and neighboring atoms n_i :

$$P_{calc} = \sum_i n_i a_i,$$

where P can be further calculated into $logP$. A full list of the atomic descriptors and contributions can be found in the main text of Wildman & Crippen (1999).

A.5 Validity, Uniqueness, and Novelty

To analyze the novelty of our compounds, we need to look at how we calculate the validity and uniqueness of our compounds.

These are given as follows:

$$\text{Validity} = \frac{\text{Number of valid molecules}}{\text{Number of generated molecules}},$$

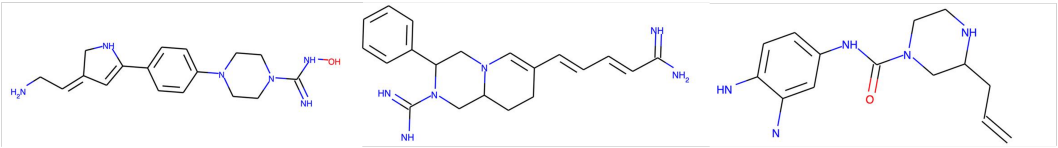
$$\text{Unique} = \frac{\text{Number of unique molecules}}{\text{Number of valid molecules}},$$

$$\text{Novelty} = \frac{\text{Number of generated molecules not in training set}}{\text{Number of unique and valid generated molecules}}$$

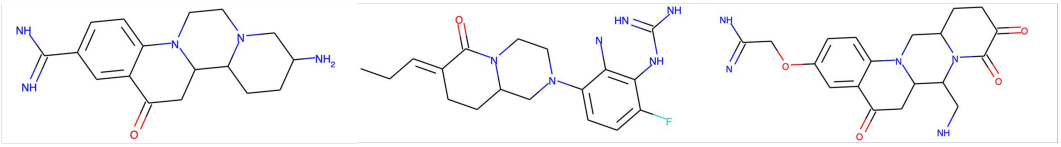
B TOP CANDIDATES

Shown below are snapshots of the top 3 candidates from each of the reward functions presented in this work.

B.1 2D REPRESENTATIONS



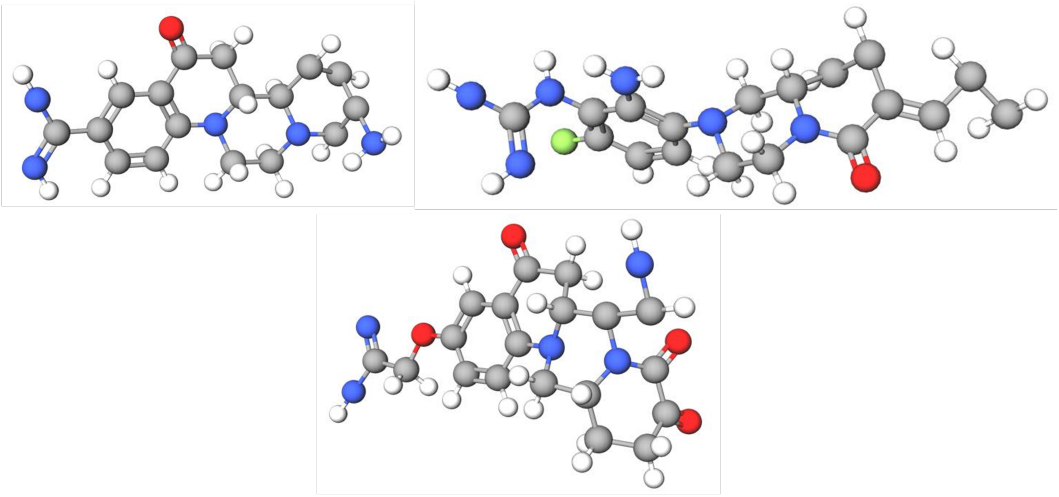
(a) 3D-MolGNN$_{RL}$ - R2 Top 3 candidates: 94.49%, 91.02%, 84.88%



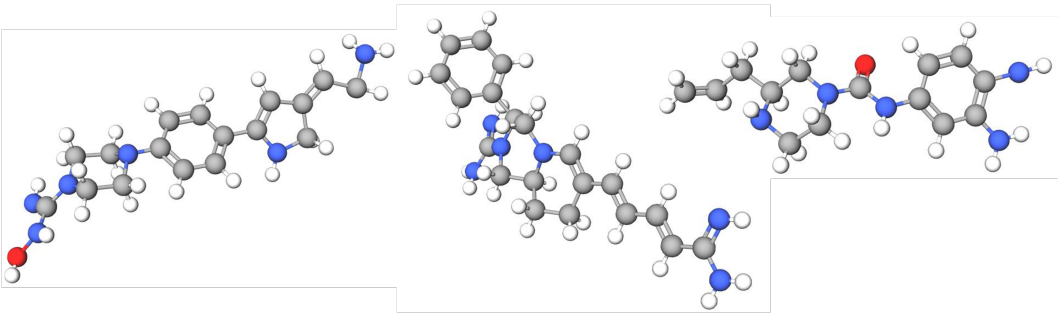
(b) 3D-MolGNN$_{RL}$ - R1 Top 3 candidates: 91.31%, 91.82%, 88.07%

Figure 5: The top 3 candidates for each reward function in this work. The candidates are shown in descending order from left to right. Their associated binding probability is listed in the subcaption also in descending order from left to right.

B.2 3D REPRESENTATIONS



(a) 3D-MolGNN$_{RL}$ - R1 Top 3 candidates: 91.31%, 91.82%, 88.07%



(b) 3D-MolGNN$_{RL}$ - R2 Top 3 candidates: 94.49%, 91.02%, 84.88%

Figure 6: The top 3 candidates for each reward function in this work. The candidates are shown in descending score order from left to right and down. Their associated binding probability is listed in the subcaption also in descending order from left to right.

C Ablation Study on Reward Function

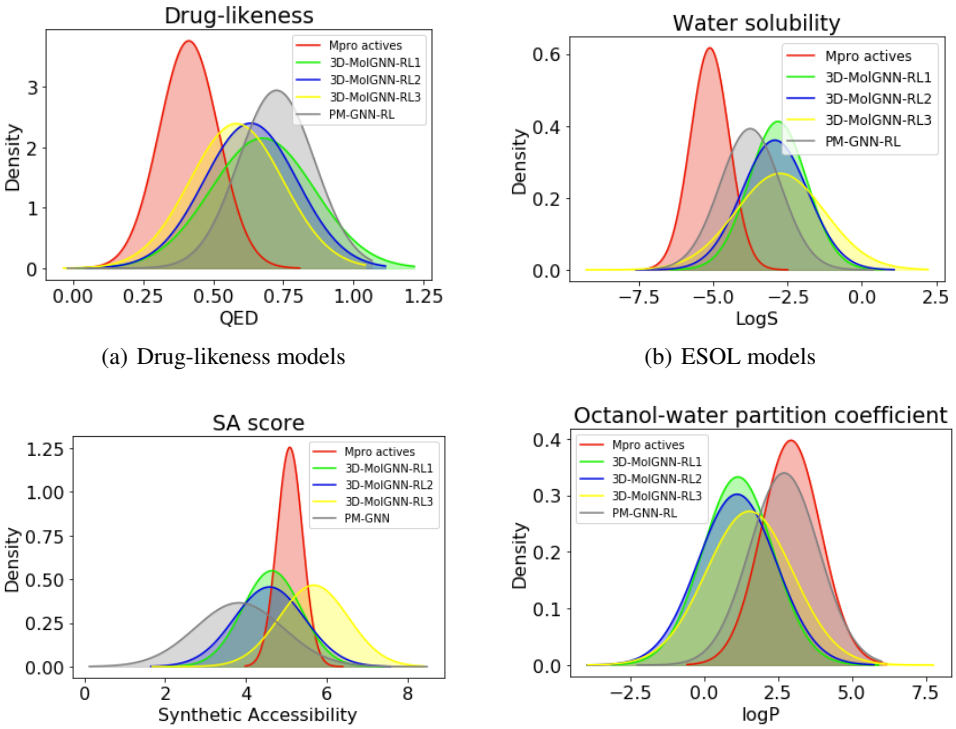
In order to see the importance of optimizing for multiple properties in the reward function, we tested different training weight, and reward function combinations for the model. The most important 2 were listed in the main text of the paper, but we tested 2 more reward functions to test performance.

We used the following combinations and compared how their output scored in the drug-like metrics. We see a steady decrease in performance from Reward 1 to Reward 3 showing that the inclusion of Binding Probability is important for generating molecules with a drug-like features. We also compare our 3D method with the 2D protein inclusion using the original PaccMann Method but with our GNN instead of their PaccMann critic to see if the 2D features have an effect.

$$R_1(s_t) = \alpha.C_{BP}(B, C_t) + \beta.C_{EA}(B, C_t) + (1 - \gamma.C_{SA}(C_t))$$

$$R_2(s_t) = \alpha.C_{BP}(B, C_t) + (1 - \beta.C_{SA}(C_t))$$

$$R_3(s_t) = \alpha.C_{EA}(B, C_t) + (1 - \beta.C_{SA}(C_t))$$



(a) Drug-likeness models

(b) ESOL models

(c) Synthetic accessibility score models

(d) Partition coefficient models

Figure 7: Comparison of the properties of the molecules produced from 3D-MolGNN$_{RL}$ against the experimentally identified active compounds for M_{pro} target.

Overall, figure 7 shows that the complete 3 term reward function for the 3D-MolGNN$_{RL}$ method produces the best scoring molecules overall for the given drug-like metrics.

D 3D-Scaffold Hyperparameters

For the 3D Scaffold model we use the following hyperparameters: batch size of 2, split of 1000-500 training, validation set, 150 max epochs, learning rate of 0.0001, learning rate patience of 10, learning rate minimum of 1e-6, learning rate decay of 0.5, atom-wise representation layer size of 32, cutoff radius of local environment of 10, interaction layer size of 6, 25 gaussians to expand distances with a max distance of 15 and 300 distance bins.

E GNN HYPERPARAMETERS

Hyperparameters such as network depth, layer dimension, and learning rate can have a large effect on model training and the weights in the final realized model. Therefore, we performed a number of trainings to examine combinations of learning rate, number of attention heads, and layer dimension. We used the parameters from our best model to create a baseline: a learning rate of 0.0001, four attention heads, and a layer dimension of 70 produced an average test AUROC of 0.854. The test parameters and top scoring configuration are outlined and highlighted in Table 3. The hyperparameters that performed best were a learning rate of 0.0001, two attention heads, and a dimension of 70. These parameters resulted in an average test AUROC of 0.864, still scoring lower than our implemented baseline configuration.

Table 3: Values of examined hyperparameters: learning rate (lr), number (N) of attention heads, and dimension (D) of the GAT layer in each attention head. A grid search was performed on each combination of parameters. The optimal combination is shown in bold.

lr	N	D
0.001	2	**70**
0.0001	3	140
0.00001	**4**	210
		280

Trainings were carried out over 200 epochs on a quarter of the data taken from our dataset consisting of 2,000 samples per target with 79 targets, using the same train-test split for all trainings. The decreased performance compared to the models is expected due to the reduced set of data used to train the model. Thirty of the 36 combinations trained without error and are shown in Table 4.

Table 4: Values of examined hyperparameter set with average train and Test ROC .

HYPERPARAMETER SET	TRAIN ROC AVG	TEST ROC AVG
LR_0.001_N5_D70	0.754	0.771
LR_0.001_N4_D210	0.606	0.647
LR_0.001_N4_D140	0.605	0.661
LR_0.001_N3_D70	0.834	0.835
LR_0.001_N3_D280	0.610	0.630
LR_0.001_N3_D210	0.672	0.694
LR_0.001_N3_D140	0.714	0.742
LR_0.001_N2_D70	0.937	0.861
LR_0.001_N2_D140	0.788	0.781
LR_0.001_N2_D280	0.762	0.755
LR_0.001_N2_D210	0.767	0.773
LR_0.0001_N4_D128	0.898	0.854
LR_0.0001_N3_D70	0.894	0.855
LR_0.0001_N3_D210	0.930	0.860
LR_0.0001_N3_D140	0.917	0.862
LR_0.0001_N2_D70	0.912	0.864
LR_0.0001_N2_D210	0.950	0.861
LR_0.0001_N2_D140	0.940	0.853
LR_0.00001_N4_D70	0.707	0.726
LR_0.00001_N4_D210	0.776	0.783
LR_0.00001_N4_D140	0.748	0.769
LR_0.00001_N3_D70	0.735	0.758
LR_0.00001_N3_D210	0.778	0.772
LR_0.00001_N3_D140	0.763	0.755
LR_0.00001_N2_D70	0.713	0.749
LR_0.00001_N2_D210	0.776	0.783
LR_0.00001_N2_D140	0.774	0.782

F MATH NOTATIONS

S_t - state of the molecule at for action t

T - Terminal state/Max length of molecule

S - all possible states

R_t - reward for the partial molecule at step t

M_t - candidate molecule

B - target protein

X_c - Target embeddings from protein-VAE for SELFIES-VAE G - generator

C_{BP} - Critic for Binding Probability prediction

C_{SA} - Critic for SA score prediction

C_{EA} - Critic for Experimental Affinity prediction

$\Pi(\Theta)$ - optimization policy

N - steps per episode

P_Θ - probability associated with the action t

Z_{next} - ground truth type of the next atom

$\hat{p}_{\text{type}}^{Z_{\text{next}}}$ - probability that the model assigns to that type at the current step.

$\hat{p}_j^b$ - probability that the model assigns for the distance between $\mathbf{r}_j$ and $\mathbf{r}_{\text{next}}$ to fall into distance bin $b \in B$ at the current step

PM-GNN$_{RL}$ model

$\Pi(\Theta) = \sum_{s_t \in S} P_\Theta(s_t) R(s_t)$

$$R(s_t) = \begin{cases} \alpha.C_{BP}(B, C_t) + \beta.C_{SA}(C_t), & t = T \\ 0, & t < T \end{cases}$$

3D-MolGNN-$_{RL}$ model

$\Pi(\Theta) = \sum_{s_t \in S} (P_\Theta(s_t) - R(s_t))$

$P_\Theta(s_t) = P_\Theta(s_t)^{type} + P_\Theta(s_t)^{dist}$

$P_\Theta(s_t)^{type} = -\log\left(\hat{p}_{\text{type}}^{Z_{\text{next}}}\right)$

$P_\Theta(s_t)^{dist} = \sum_{j=1}^{N} \sum_{b \in B} q_j^b \log\left(\hat{p}_j^b\right)$

$R_1(s_t) = \alpha.C_{BP}(B, C_t) + \beta.C_{EA}(B, C_t) + (1 - \gamma.C_{SA}(C_t))$

$R_2(s_t) = \alpha.C_{BP}(B, C_t) + (1 - \beta.C_{SA}(C_t))$

$R_3(s_t) = \alpha.C_{EA}(B, C_t) + (1 - \beta.C_{SA}(C_t))$