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

Anonymous authors
Paper under double-blind review

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 given protein target 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 pocket 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 binding affinity, potency, and synthetic accessibility of the candidates generated for the SARS-CoV-2 Main Protease (M_{pro}).

1 INTRODUCTION

Recent advancements in machine learning (ML) and artificial intelligence (AI) have demonstrated the potential to revolutionize drug design 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). The potential chemical space is composed of over 10^{60} molecules, and 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 for identifying new drug candidates is that the generated molecule must have optimized properties that allow it to effectively bind with the required target and have a minimal off-target effect. However, if we consider molecule generation as a controlled and a dynamic step-by-step process, it is possible to produce end products that possess these optimization properties . This approach allows us to formulate de novo drug design as a reinforcement learning (RL) problem and utilize algorithms that best learn the 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

The current work aims to create a novel approach to address the problem of generating molecules optimized for a specific protein target starting from scratch while optimize these candidates for multiple properties. In recent years, RL based work to generate molecules have been introduced that attempt to tackle the problem in different ways. Some consider the protein target as a sequence while generating candidates, others do not. A method by Popova et al. (2018) utilizes a fragment-replacement-based approach to optimizing 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 variational autoencoders (VAEs), which learn on the latent space representation of

molecules to try and 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. These methods have tested the drug-likeness post-generation against some given target protein, but the target was never directly involved during the RL loop. When designing drug candidates that target specific proteins, learning how the molecule interacts with these proteins is an invaluable information to the generative process that can help in accelerating the automation in 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. 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 used VAEs to generate 2D molecules, but it combines the latent space of the SMILES with the embedding of a specific protein to generate molecules with protein information in consideration.

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 also considers the 3D structures of the generated compound built by placing atom by atom in the 3D space. This is something that hasn't been explored, as most methods simply rely on a 2D representation of the molecule in the form of SMILES. To overcome the sequence-based representation of the target and the drug, we upgraded the actor from a 2D-SMILES generator to a 3D-scaffold-based molecule generator. The 3D-MolGNN$_{RL}$ method uses the previously stated 3D-scaffold-based generator 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 prototype demonstrated in the PaccMann^{RL} model. As mentioned previously, the PaccMann^{RL} model uses a pre-trained compound generator called the SELFIES-VAE that produces SMILES strings of the compounds. Even though the SELFIES-VAE produces valid chemical compounds, the validity of the generated molecules in terms of binding to the actual 3D pocket of the target is still a challenge. Towards this, we enhanced the PaccMann^{RL} method with a structure-based graph neural network (GNN) critic. We retained the same actor model (protein-VAE and SELFIES-VAE) to generate the target embedding and target specific molecules respectively, but modified the model by substituting in our own GNN critic and reward function. In effect, we used a graph-based binding probability predictor (GNN$_P$) that takes the 3D structure of the protein pocket and the interacting ligand, and predicts the probability of their interaction without any prior knowledge of the intermolecular interactions. Therefore, we refer to this method as PM-GNN$_{RL}$. The GNN$_P$ model uses the residue and atomic-level representation of a protein pocket as well as an atomic and bond level representation of the molecule. This is unlike the critic from the PaccMann^{RL} model which simply uses the protein sequence and 2D molecular graph.

The major difference between PM-GNN$_{RL}$ and 3D-MolGNN$_{RL}$ is that PM-GNN$_{RL}$ uses a pre-trained agent and critic and the actor is fine-tuned during RL optimization. The 3D-MolGNN$_{RL}$ model only uses a pre-trained critic while the agent is trained from scratch. Here, we achieve fine-tuning by strictly filtering the training data.

2 Experiments and Dataset

For all the experiments, we used three different datasets. For 3D-MolGNN$_{RL}$, we used a dataset of non-covalent inhibitors from the BindingDB dataset (Gilson et al., 2016) and FDA approved drugs FDA . These compounds are filtered based on certain properties like IC$_{50}$, molecular weight, atom type, and functionality and we have almost 10K unique scaffolds. To represent the definition of scaffold, we used Murcko scaffolds Bemis & Murcko (1996). The similar dataset was used in prior 3D-Scaffold model (Joshi et al., 2021). Experimentation and results suggest that compounds containing piperazine as a functional group have higher affinity towards the SARS-CoV-2 M$_{\text{pro}}$ target

(Clyde et al., 2021; Joshi et al., 2021). 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.

For the PM-GNN_{RL} model, we used a subset of the PDBBind-2018 (Su et al., 2019) and DUDE (Mysinger et al., 2012) datasets. The targets from these datasets were filtered based on sequence similarity of the target proteins.

2.1 DATA REPRESENTATION

The PM-GNN_{RL} model is comprised of two pre-trained components in the agent: a biomolecular protein-VAE to store the target embeddings and a sequential compound generator SELFIES-VAE to generate molecules. These components are directly used from the $PaccMann_{RL}$ (Born et al., 2020) model. Both protein-VAE takes the protein sequence which is turned into embeddings for SELFIES-VAE which generates the compounds as SMILES strings. While the agent functions at the sequence level, the graph-based critic (GNN_P) uses the 3D structures of the target protein and the compound. To bridge the gap between the 2D sequence and 3D structure, we modeled the SMILES from the SELFIES-VAE to random conformer using RDkit (rdk). In addition to this, we use the target protein's representation at both the sequence and structural level since the protein-VAE in the agent generates embeddings from the amino acid sequence of the selected target while the critic predicts the binding affinity using the target pocket. All targets used in the dataset are associated with a static protein pocket cropped at a distance of 8Åfrom the bound ligand. To generate these pockets, we used the 3D crystal structure of the target protein bound against a high affinity inhibitor. The binding site of a target protein typically remains the same for different molecules; therefore, we can use the pocket from the complex with it's best inhibiting candidate as an accurate representation of that pocket for other molecules.

For 3D-$MolGNN_{RL}$ model, the agent and critic both function at the 3D structural level. Since 3D-scaffold learns to predict atoms in 3D space, it is easier to convert the partially produced molecule to a GNN readable format. Moreover, 3D-scaffold predicts the actual 3D-coordinates of the resulting molecule which can prove to be more accurate than a randomly predicted 3D-conformer.

3 METHODS

In this section, we discuss in detail the architecture and implementation of both RL methods proposed. They key differences between the two methods is that 3D-$MolGNN_{RL}$ uses 3D representations of both the protein and ligand while PM-GNN_{RL} uses sequences and 2D representations. Secondly, 3D-$MolGNN_{RL}$ uses the reward for every intermediate step of molecule generation, while PM-GNN_{RL} uses the reward at just the terminal state.

3.1 3D-$MOLGNN_{RL}$ FRAMEWORK

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

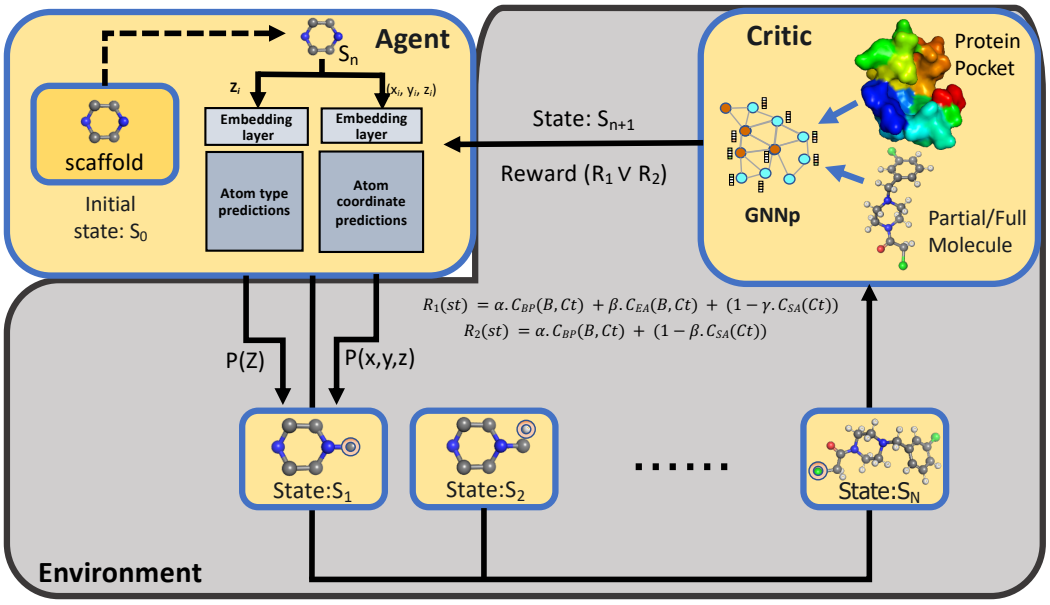


Figure 1: Schematic RL workflow highlighting the interaction of Actor and Critic in our 3D-MolGNN$_{RL}$ model. The agent starts building the molecule from a scaffold (state S_0) and subsequently builds the molecule by choosing an atom based on the reward assigned by the critic for that intermediate state.

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.

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

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

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

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

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

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 here 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 aim is to optimize 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. We used two different reward functions as $R_1(s_t)$ and $R_2(s_t)$ associating with two independent experiments. 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 ans 0.25 for binding probability and synthetic accessibility respectively. We trained the agent for 250 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.

3.2 PM-GNN$_{RL}$ FRAMEWORK

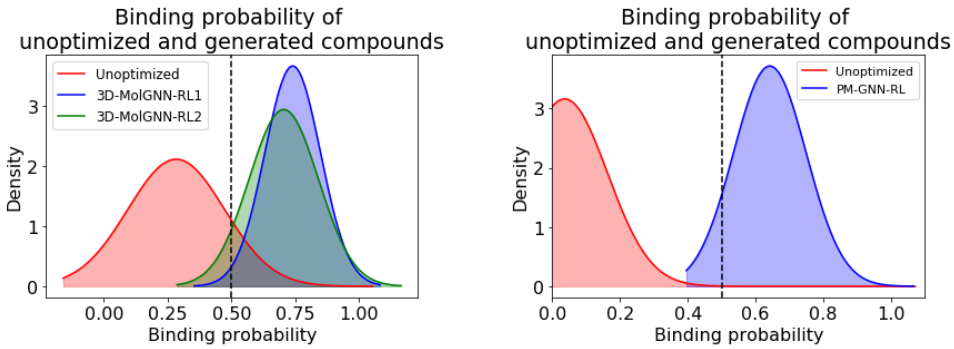
The workflow of the PM-GNN$_{RL}$ is to retrain the agent to produce valid molecules while optimizing them against a specific target with the help of the critic. Firstly the from the protein-VAE are added to the SELFIES-VAE to generate compounds starting from a functional group. Utilizing these encoding, the agent/generator G produces molecular structure which is modeled into a 3D conformer using external molecular generator and is passed into the critic C_{BP} to calculate the reward. The aim is to optimize Θ to produce target compounds C_t specific to the target B. The set of states is the set of all possible SMILES within the max length T in conjunction with the embeddings for target B. The agent G is trained to learn to optimize the policy $\Pi(\Theta)$, by maximizing $\Pi(\Theta) = \sum_{s_t \in S} P_\Theta(s_t) R(s_t)$. The reward R_t is calculated only at the terminal state while for intermediate states, it remains 0.

The reward at terminal state is calculated as a difference between the binding probability of the target with the compound and the Synthetic Accessibility (SA) of the compound. While the original PaccMann^{RL} model considered Toxicity coupled with binding probability/IC_{50} of the target with compound, we chose to use the SA score instead of toxicity. We used two constant weight multipliers α=1 and β=0.5 for binding probability and SA score respectively. The SA scores of the compounds are scaled to be in between 0 and 1 to match the scale of the binding probability.

4 RESULTS

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.

We first demonstrated that the RL portion of the methods were essential to producing molecules with a high binding probability. To accomplish this, we generated molecules for the protein target, M_{pro}, 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



(a) 3D-MolGNN$_{RL}$ models (b) PM-GNN$_{RL}$ models

Figure 2: Comparison of generated compounds before and after optimization by 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$. The unoptimized molecules were generated using the respective agents for each method over the same data but without involvement of the critic.

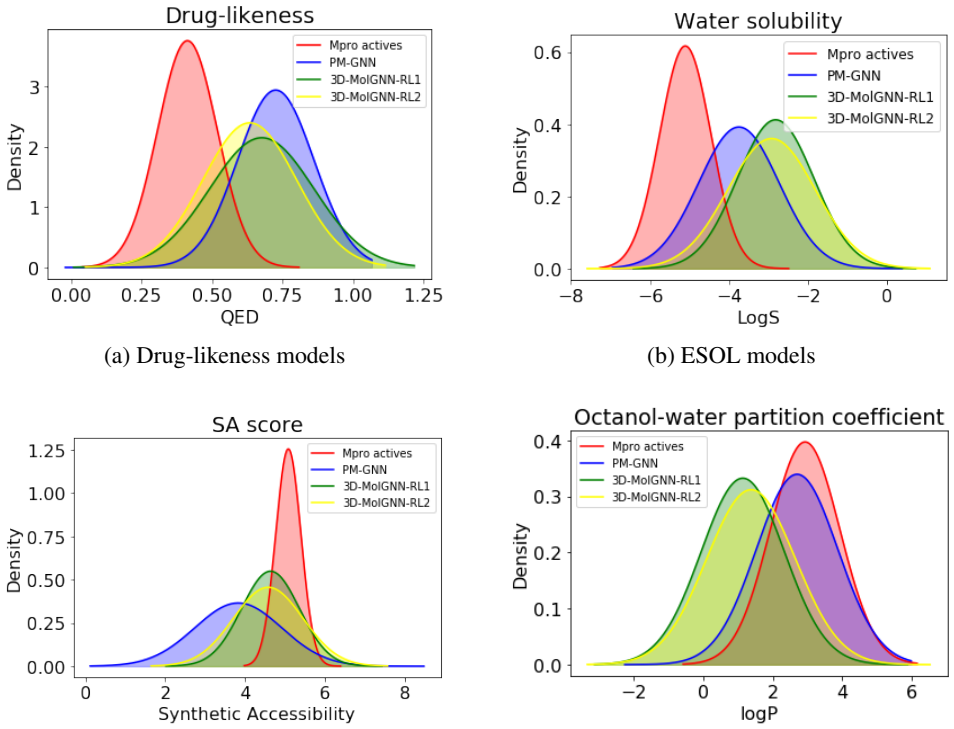
comparison of binding probabilities of the generated compounds towards the M$_{\text{pro}}$ target from the 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ models respectively are shown in figure 2. From the figure, it is evident that the RL mechanism has improved the agent's performance in producing candidates with a greater binding probability. The unoptimized molecules generated by the agents for both in the 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ show a very low binding probability towards M$_{\text{pro}}$ as the predictions are centered around the lower probabilities. On the other hand, the biased compounds from RL optimization show a great improvement in terms of their binding ability as most of the predictions are concentrated towards the high probability values. The PM-GNN$_{RL}$ has a significant increase of 1672.19% from unoptimized to RL optimized mean binding probability, while the 3D-MolGNN$_{RL}$ with the $R_1(s,t)$ reward function produces a 161.05% increase from unoptimized and the model with the $R_2(s,t)$ reward function has a 148.69% increase of the mean.

Next, we looked at the four different drug-likeness properties listed above to serve as comparison metrics. Figure 3 compares these metrics for both 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ with known active molecules for M$_{\text{pro}}$. A full description of each metric is given in the appendix (section A). The first metric, QED, represents a quantification of the desirability of the drug (Bickerton et al., 2012). We can see that in Figure 3a, the RL methods all produce higher scoring molecules than the current experimentally determined M$_{\text{pro}}$ actives. Comparing the mean of the M$_{\text{pro}}$ distribution to the other three, we see that we get a 75.92% improvement with the PM-GNN$_{RL}$ model, a 63.93% improvement with the $R_1(s,t)$ model of the 3D-MolGNN$_{RL}$ method, and a 53.54% improvement using the $R_2(s,t)$ model.

We next look at the distributions of water solubility calculated using the ESOL method (Delaney, 2004). Figure 3b 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 PM-GNN$_{RL}$ improves 26.53% relative to the mean of the M$_{\text{pro}}$ actives, 3D-MolGNN$_{RL}$'s $R_1(s,t)$ model has a 44.74% improvement and the $R_2(s,t)$ model has a 42.72% improvement.

The next metric, synthetic accessibility (SA), is a term included in each of the reward functions of the RL methods tested. Figure 3c illustrates the distributions obtained by each method and highlights some interesting results. The PM-GNN$_{RL}$ model produces the most synthesizable molecules, but also produces the widest range, not always guaranteeing a low SA. This method improves the mean by 24.7% relative to the M$_{\text{pro}}$ actives. The two 3D-MolGNN$_{RL}$ models on the other hand, have a much more consistent range of SA scores that center around a 5, which indicates consistently below average scores. $R_1(s,t)$ shows an 8.78% improvement and $R_2(s,t)$ shows a 9.92% improvement. 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 3d, we can see how each method produces molecules spanning the range of -2 to 6. The PM-GNN model provides a 7.87% improvement to the M$_{\text{pro}}$ actives in logP value, whereas the



(a) Drug-likeness models

(b) ESOL models

(c) Synthetic accessibility score models

(d) Partition coefficient models

Figure 3: Comparison of the properties of the molecules produced from 3D-MolGNN$_{RL}$ and PM-GNN$_{RL}$ models against the experimentally identified active compounds for M$_{\text{pro}}$ target.

3D-MolGNN$_{RL}$ method obtains a 61.21% improvement using the $R_1(s,t)$ reward and a 53.08% improvement using the $R_2(s,t)$ reward.

Table 1: Details about the drug-likeness metrics proposed in this work. We look at all three reward functions outlined in the text: PM-GNN's reward function, and 3D-MolGNN$_{RL}$'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 the mean for QED. 2D and 3D snapshots of the top 3 candidates from each method are available in appendix section B.

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 RDkit. The methods of calculating these properties are described in further detail in appendix subsection A.5. From table 2, we found that in terms of validity, PM-GNN$_RL$ produced molecules with 100% validity, followed by 3D-MolGNN$_{RL}$ R1 and finally by 3D-MolGNN$_{RL}$ R2. In terms of uniqueness and novelty however, both the 3D-MolGNN$_{RL}$ methods performed better

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%

than the PM-GNN$_{RL}$ methods. These methods yielded nearly 100% unique and novel compounds from their valid set. 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 highest Validity, Uniqueness and Novelty.

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 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.

REPRODUCIBILITY STATEMENT

All work done for this paper is reproducible by obtaining the correct code and data. The anonymous code for the 3D-MolGNN$_{RL}$ method can be found in the Supplementary Information, while the data used to train and test the methods can be found in section 2. Each performance metric can be calculated from the code, but is also available in appendix section A.

REFERENCES

Zinc database, (http://zinc.docking.org/substances/subsets/fda/?page=1(30 August 2020, date last accessed).

Landrum G. RDKit: Open-Source Cheminformatics Software. 2016; (30 September 2020, date last accessed) http://rdkit.org/.

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, Leighton Coates, Ian Foster, Darin Hauner, Vilmos Kertesz, Neeraj Kumar, Hyungro Lee, Zhuozhao Li, Andre Merzky, Jurgen G. Schmidt, Li Tan, Mikhail Titov, Anda Trifan, Matteo Turilli, Hubertus Van Dam, Srinivas C. Chennubhotla, Shantenu Jha, Andrey Kovalevsky, Arvind Ramanathan, Martha S. Head, and Rick Stevens. High throughput virtual screening and validation of a sars-cov-2 main protease non-covalent inhibitor. *bioRxiv*, 2021. doi: 10.1101/2021.03.27.437323. URL https://www.biorxiv.org/content/early/2021/04/02/2021.03.27.437323.

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.

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. URL http://papers.nips.cc/paper/8005-constrained-graph-variational-autoencoders-for-molecule-design.

Michael M. Mysinger, Michael Carchia, John. J. Irwin, and Brian K. Shoichet. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, jul 2012. ISSN 0022-2623. doi: 10.1021/jm300687e. URL https://pubs.acs.org/doi/10.1021/jm300687e.

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.

Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling*, 59(2):895–913, feb 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00545. URL https://pubs.acs.org/doi/10.1021/acs.jcim.8b00545.

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:

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

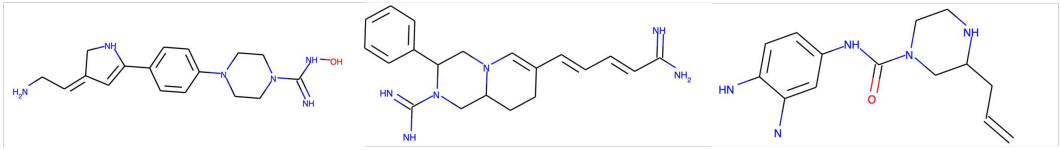
$$\text{Unique} = \frac{\text{Number of unique molecules}}{\text{Number of valid molecules}},$$

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

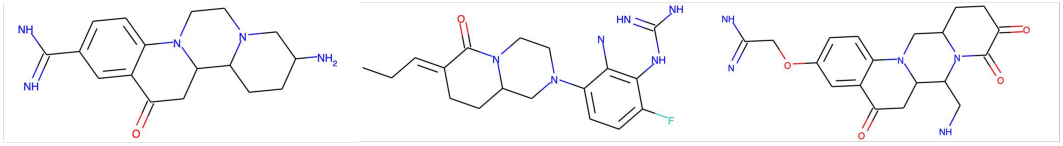
B TOP CANDIDATES

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

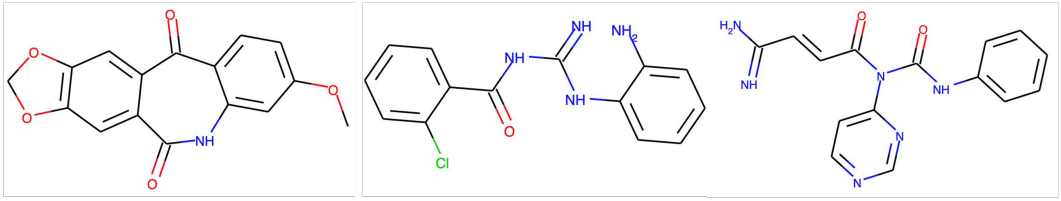
B.1 2D REPRESENTATIONS



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



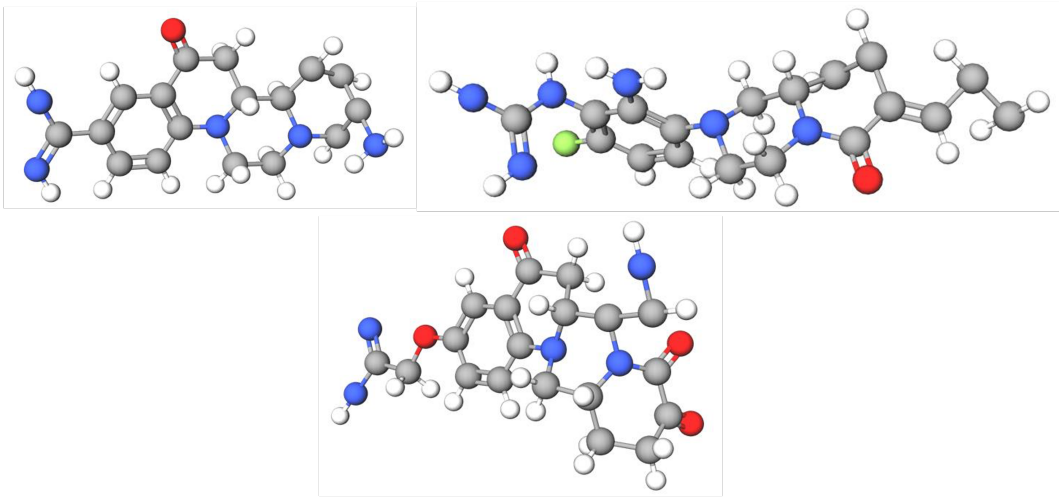
(b) 3D-MolGNN$_{RL}$ - R1 Top 3 candidates: 91.31%, 91.82%, 88.07%



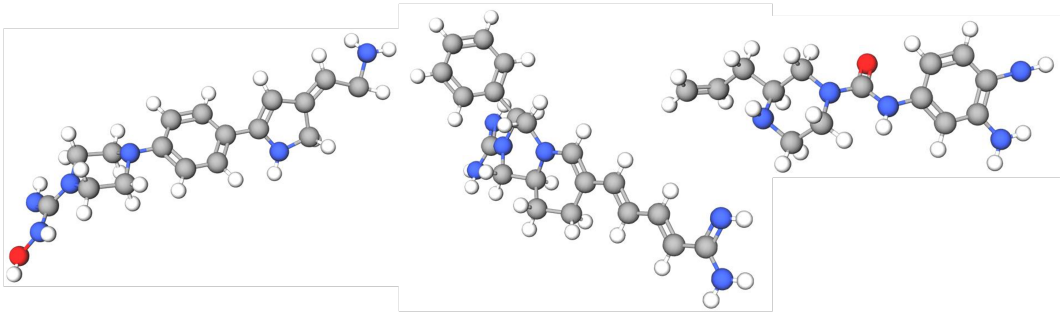
(c) PM-GNN$_{RL}$ Top 3 candidates: 90.93%, 87.42%, 85.45%

Figure 4: The top 3 candidates for each method and 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.

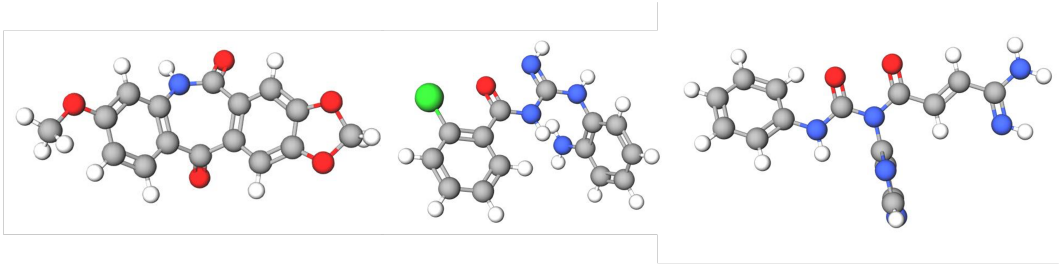
B.2 3D REPRESENTATIONS



(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%



(c) PM-GNN$_{RL}$ Top 3 candidates: 90.93%, 87.42%, 85.45%

Figure 5: The top 3 candidates for each method and 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 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

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

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

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

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

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

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