JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

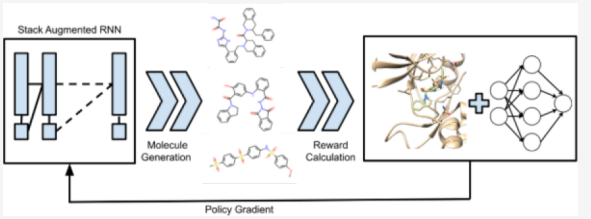
MoleGuLAR: Molecule Generation Using Reinforcement Learning with Alternating Rewards

Manan Goel, Shampa Raghunathan, Siddhartha Laghuvarapu, and U. Deva Priyakumar*

Cite This: https://doi.org/10.1021/acs.jcim.1c01341 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The design of new inhibitors for novel targets is a very important problem especially in the current scenario with the world being plagued by COVID-19. Conventional approaches such as high-throughput virtual screening require extensive combing through existing data sets in the hope of finding possible matches. In this study, we propose a computational strategy for de novo generation of molecules with high binding affinities to the specified target and other desirable properties for druglike molecules using reinforcement learning. A deep generative model built using a stack-augmented recurrent neural network initially trained to generate druglike molecules is optimized using reinforcement learning to start generating molecules with desirable properties like LogP, quantitative estimate of drug likeliness, topological polar surface area, and hydration free energy along with the binding affinity. For multiobjective optimization, we have devised a novel strategy in which the property being used to calculate the reward is changed periodically. In comparison to the conventional approach of taking a weighted sum of all rewards, this strategy shows an enhanced ability to generate a significantly higher number of molecules with desirable properties.



1. INTRODUCTION

The advent of data-driven techniques across multiple domains of computer science, such as robotics, natural language processing, and computer vision, has found immense success, and this has led to their application in natural sciences.[1,2] The curation of large data sets[3−5] has increased the relevance of machine-learning-based approaches in problems like molecular property prediction, conceiving retrosynthetic pathways, protein structure prediction, and drug discovery.[6−9]

Drug discovery is a long, expensive, and arduous process which combines a wide range of disciplines including chemistry, biology, and pharmacology. For a novel target, the conventional approach is to perform high-throughput screening on chemical libraries to identify small molecules that bind well to the target. The identified hits are then optimized to get higher binding affinity, reduce toxicity, and improve oral bioavailibity.[10,11] The time and expense involved in this process give rise to alternate in silico approaches like virtual screening wherein small molecules from existing drug libraries are computationally evaluated by generating protein ligand complexes using docking calculations and ranking them using a scoring function.[12,13] However, these also come with the caveat that finding the most stable conformation of the complex is a nonconvex optimization problem, and it can take a very large amount of time (≈10 min per molecule) to find the most optimal conformation. These can be made faster using machine-learning-based approaches like the works of Aggarwal et al.[14] for detecting the ligand-binding site, Chelur and Priyakumar[15] for binding residue detection, and Mehta et al.[16] for enhanced molecular sampling.[14−16] However, even the most exhaustive studies[17] have been able to find binding affinities of $\approx 10^8$ molecules on a single target which is minuscule in comparison to the vast magnitude of the chemical space with about 10^{60} synthesizable molecules.[18] This posits the argument for the de novo generation of molecules with high binding affinities to the required target instead of searching in existing libraries.

Machine-learning-based approaches like recurrent neural networks, generative adversarial networks (GANs), and variational autoencoders have recently been adopted for molecule generation. Gupta et al.[19] used long short-term memory recurrent neural networks, generally used for natural language processing tasks, to generate molecules in the form of SMILES (simplified molecular-input line-entry system), which is a string representation of molecules and has it is own grammar and semantics.[20] GANs are generative models that learn the probability distribution of the training data, and sampling from the distribution can then be used to generate synthetic data points. This model has also been applied to the generation of molecules with desirable properties in works by De Cao and Kipf,[21] Prykhodko et al.,[22] and Maziarka et al.[23] Jin et al.[24] used the graph representation of molecules to train a variational autoencoder that could then generate graphs of new

Received: November 5, 2021

molecules.[24] Kusneret al.,[25] Griffiths and Hernández-Lobato,[26] and Lim et al.[27] used SMILES representations for generating molecules through the variational autoencoder architecture.[25−27] In fact, applications of deep learning models for molecule generation have proven to be very successful in recent years.[19,28−30]

The next challenge is to generate molecules with desirable properties for which the two major approaches being adopted are reinforcement learning and latent space optimization. Variational autoencoders are capable of learning a continuous space representation of molecules[31−34] which can then be optimized to generate molecules with target properties through techniques like Bayesian optimization and swarm optimization.[35,36] Gao et al.[37] used an autoencoder architecture and a generator model in combination to incrementally update the latent space representation of the given molecule to reach a molecule with the desired properties.[37] Reinforcement learning can be used to generate desirable molecules by decomposing the process as a sequence of states and actions to maximize a reward which in this case is the desirable property. Popova and coworkers used stack-augmented gated recurrent units (GRUs) to generate molecules followed by reinforcement learning guided optimization on properties like LogP, quantitative estimate of drug likeliness (QED), and synthetic accessibility.[38] Guimaraes et al.[39] combined the GAN and reinforcement learning frameworks for the task while You et al.[40] and Khemchandani et al.[41] used a graph-based policy network to generate molecular graphs.[39−41] Boitreaud et al.[42] and Born et al.[43] combined a variational autoencoder model with reinforcement learning to generate molecules with high binding affinities to the specified target and antiviral candidates, respectively.[42−44] Bung et al.[45] used reinforcement learning with a SMILES generator to generate molecules with high binding affinities to JAK2 and SARS-CoV-2 3CL proteins, respectively.[45,46]

In this work we propose a molecule generation pipeline, MoleGuLAR (molecule generation using reinforcement learning with alternating rewards, Figure 1), which uses a stack augmented recurrent neural network (RNN) initially trained to generate valid druglike molecules which is then optimized to generate molecules with a high binding affinity to the specified target. For the binding affinity calculation, we tried two methodologies: (1) performing docking calculation to find the most stable complex and the corresponding binding affinity and (2) using a machine-learning model trained to predict binding affinities. In the case of multiobjective optimization, we found that using a weighted sum of the rewards from different properties may not be effective in some cases because it is possible that one or more properties dominate others leading to poor results. Hence, we also propose a novel optimization strategy in which the reward is alternated so that the model changes the region from which it samples in the chemical space. When the reward is changed, the generator starts from a better position with respect to one property when optimization for another property is started. We also showcase its application on two proteins: M_{pro} of SARS-CoV-2 and TTBK1 with a wide set of target properties. The robustness of this strategy is further showcased by using it to optimize the model for conflicting properties along with the binding affinity.

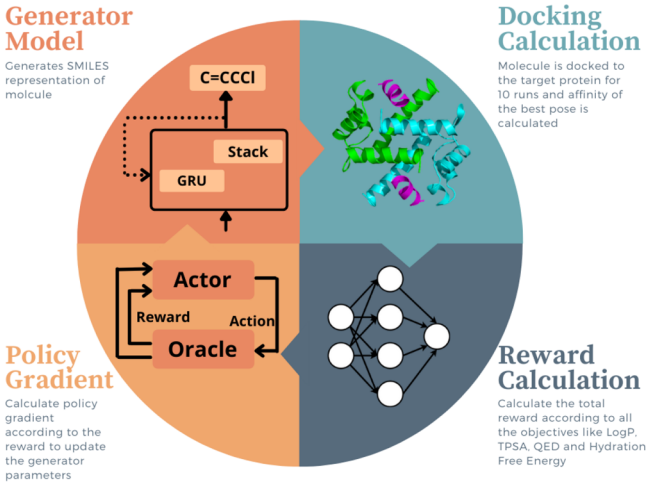


Figure 1. Pipeline used by MoleGuLAR for generating molecules with high binding affinity to a specified drug target.

2. THEORY AND METHODS

This section describes the various components of the proposed framework (Figure 1). The formulation of the stack-augmented RNN as the generator model is detailed in Section 2.1 followed by methods for binding affinity calculation and hydration free energy calculation in Sections 2.2 and 2.3, respectively. The formulation of the molecule generation as a Markov decision process, use of reinforcement learning to maximize a given reward function using policy gradient, and the two optimization strategies used in this study are described in Section 2.4.

2.1. Generator. The generator module makes use of a stack-augmented GRU which outputs molecules as SMILES strings as presented by Popova et al.[38,47] A valid SMILES string must have correct valency for all atoms, and all ring openings and closures must be counted; hence, conventional RNNs do not work well on this task because of their inability to count. Therefore, the addition of a memory unit along with the RNN forms an appropriate model which is explained further in Section S1.1 of the Supporting Information.

The stack RNN is initially trained on ≈1.5 million druglike molecules from the ChEMBL21 database[5] to learn the rules and grammar of SMILES strings.

2.2. Binding Affinity Calculation. *2.2.1. Docking Calculations.* The generator model once trained is then used to produce ligands which are then docked to the specified target to find the most stable conformation of the complex and to find the corresponding binding affinity further referred to as BA in the article. The 3D structure of the molecule from the SMILES string is obtained using the RDKit toolkit.[48] The target proteins TTBK1 (PDB ID: 4BTK) and SARS-CoV-2 M_{pro} complexed with N3 inhibitor (PDB ID: 6LU7) are obtained from the Research Collaboratory for Structural Bioinformatics PDB.[49] The ligand and protein structure is then converted to a format suitable for the input to the docking software using AutoDockTools4. The molecule docking grid is generated in the next step using the AutoGrid4 utility, and finally the docking calculation is done using AutoDock-GPU while keeping the protein active site rigid.[50,51] This tool is referred as AutoDock in the rest of the article. Detailed information about the docking methodology is provided in the Supporting Information.

2.2.2. Machine-Learning Models. We also tested the use of machine-learning models as a placeholder for AutoDock to predict the binding affinities of the generated ligands instead of performing docking calculations to reduce the computation time. In order to do this, we obtained a data set of ≈2 million

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

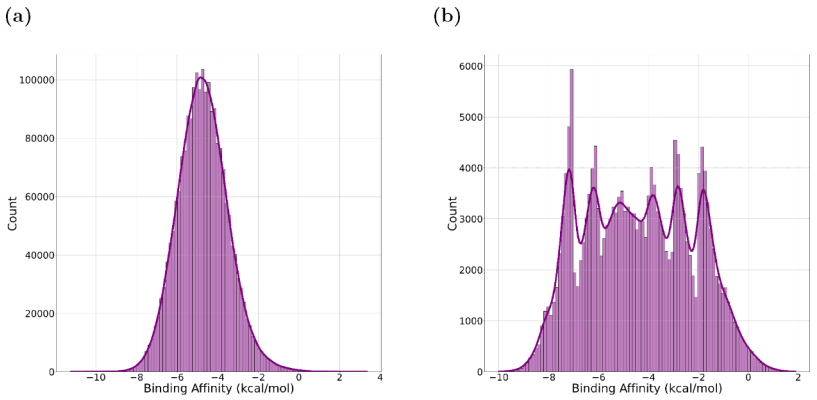


Figure 2. Distribution of binding affinities. (a) Two million molecules with TTBK1 and (b) selected molecules from buckets.

molecules obtained from the HTS collection by Enamine[52] docked with the TTBK1 protein.

Figure 2a shows that a significant number of molecules lie in a small range of binding affinities, and hence, using that for the predictor model tends to overfit (Figure S2 of the Supporting Information). In order to tackle this issue, we split the entire data set into smaller bins of 1 kcal/mol and sampled 25K molecules from each bin and all the molecules if the number of molecules is less than 25K. Figure 2b shows the distribution of the obtained subset consisting of ≈200 K molecules. This is then split into training, testing, and validation sets in the ratio 80:10:10.

We further tested various machine-learning models for this regression task. Jaeger et al.[53] proposed the Mol2vec[53] model for learning vector representations of SMILES strings that can then be used as input for further downstream tasks like binding affinity prediction as done by Mehta et al.[16] along with predicting other properties. Using these embeddings, a random forest model with 250 decision trees was trained for predicting the BA. The aforementioned model with input features from the embeddings obtained from graph isomorphism networks (GIN) proposed by Xu et al.[54] and Hu et al.[55] were also used for the task. The drawback of these approaches is that the embeddings being used remain constant during training and that leads to poor performance (Figure S3 of the Supporting Information). Fine tuning these representations during the training process helps to improve the accuracy for which three linear layers were added with the GIN embeddings, and the model is then trained end-to-end.

2.3. Hydration Free Energy Prediction. The hydration free energy (ΔG_{Hyd}) of a molecule measures its interaction with water and forms an important part of the drug delivery system. The state of the art methods for predicting it include message-passing neural networks (MPNN)[9] as shown by Wu et al.[3] in MoleculeNet[3] and chemically interpretable graph interaction networks (CIGIN)[56] by Pathak et al.[56] To predict ΔG_{Hyd}, presently we took out 10% molecules out of 643 molecules as a hold out test set and performed fivefold cross-validation on the remaining data set with 10% going in the validation set and 80% in the training set.

2.4. Reinforcement Learning. A reinforcement-learning pipeline generally consists of two modules: the actor and the critic. The actor takes the current state (s_t) of the system and performs an action (a_t) that should maximize the reward. The critic sees a_t, s_t, and the state obtained by performing the action (s_{t+1}) and penalizes or rewards the actor.

Generation of a SMILES string can be modeled as a Markov decision process where s_t denotes the SMILES string constructed so far, and a_t denotes the addition of a token to s_t. We also define a terminal state s_T which signifies the end of the molecule and initial state s_0. The whole generation process is depicted in Figure 3.

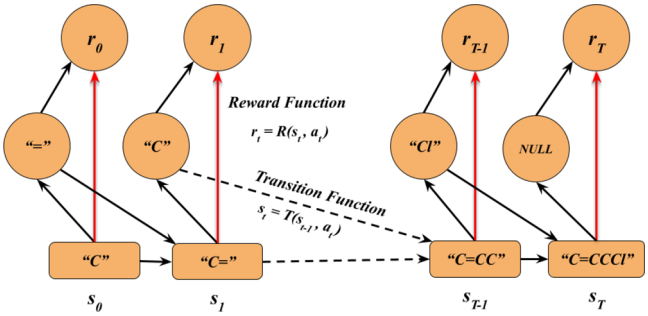


Figure 3. Illustration of SMILES string generation as a Markov decision process. At each state s_t, the agent performs an action a_t to give the updated state s_{t+1} and provide a reward according to the state.

The generator model parametrized by Θ estimates the probability, $p(a_t|s_t, \Theta)$, samples a_t from the probability distribution, and updates the state until s_T is reached. Rewards of all intermediate states s_t with $t < T$ are 0 since it is possible that the intermediate SMILES strings may not represent a valid molecule. s_T is then sent to the critic which returns the reward $r(s_T)$. Hence, the task here is to find Θ such that the expected reward given by eq 1 is maximized. This is done using the REINFORCE algorithm[57] which is detailed further in the Supporting Information.

$$R(\Theta) = \mathbb{E}[r(s_T)|s_0, \Theta] = \sum_{s_T \in S} p_\Theta(s_T) r(s_T) \quad (1)$$

For the multiobjective setup, the reward function $r(s_T)$ is composed of multiple components from the different properties for which the model is being optimized. The two reward strategies that we propose are

- Weighted Sum Rewards: The total reward $r(s_T)$ is expressed as a weighted sum of all other components:

$$r(s_T) = w_1 D(s_T) + w_2 L(s_T) + w_3 Q(s_T) + w_4 T(s_T) + w_5 H(s_T)$$

where D fetches the reward for BA, L for LogP, Q for QED, T for topological polar surface area, and H for ΔG_{Hyd}. Weights are kept as hyperparameters and remain constant throughout the

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

optimization process. The functional forms of the reward for each property are given in Table S2 of the Supporting Information.

•Alternating Rewards: The aforementioned approach does not work especially in cases where properties are conflicting like high TPSA, and more negative hydration free energy would be contradictory in nature. In such cases we have devised a strategy wherein all the weights are changed to 0 except one. This takes the generator model into the space where one property is optimal providing a better starting point when optimization for the other property is started. The current strategy works extremely well across most of the tasks and removes the requirement for acute hyperparameter tuning to find the most optimal weights for each reward function. Further details are given in the Results and Discussion section.

3. RESULTS AND DISCUSSION

This section describes the performance of machine-learning models for predicting binding affinity and ΔG_{Hyd} as well as application of the proposed pipeline on the targets:

Table 1. Performance of Predictor Models for BA in Terms of Performance Metrics MAE (kcal/mol) and Coefficient of Determination (R^2)

model	MAE (kcal/mol)	R^2
graph embeddings + random forest	0.87	0.55
Mol2vec + random forest	0.47	0.91
graph isomorphism network (GIN)	0.45	0.93

- SARS-CoV-2 M_{pro} (PDB ID: 6LU7): With the world in the midst of a global pandemic caused by COVID-19, the main protease (M_{pro}) has been identified as an important target due its vital role in viral transcription and replication.[58]
- TTBK1 (PDB ID: 4BTK): Neurodegenerative diseases have become extremely common over the past few years, and the tau-tubulin kinase 1 has proved to be an attractive target to combat a wide variety of neurodegenerative diseases.[59]

All of the presented experiments were performed using an Intel Xeon E5-2640 v4 processor and a Nvidia GeForce RTX 2080Ti GPU. The implementation details of all the machine-learning models are described in Section S1.3 of the Supporting Information.

3.1. Machine-Learning Predictor Models.

3.1.1. Binding Affinity. The performance of the random forest model with different embeddings and the performance of GIN discussed in Section 2.2.2 for BA prediction on the test set are reported in Table 1.

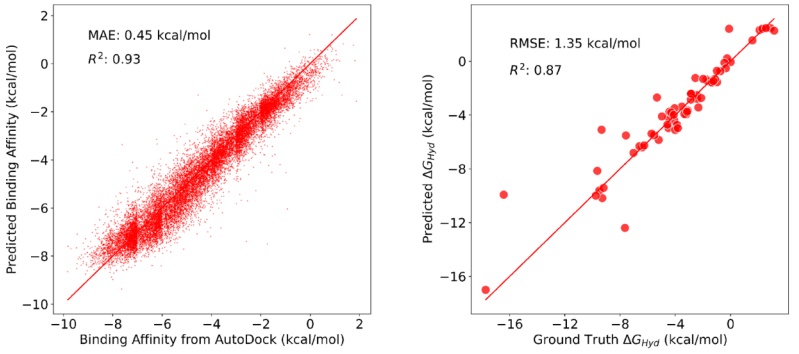
The use of constant embeddings for the random forest models leads to a higher mean absolute error (MAE) in comparison to the GIN model because in the latter, the model learns to automatically extract more relevant information from the molecular graph. The former also showed comparatively poorer performance in the desirable region, i.e., where the BA is high due to the lesser number of samples in that range in the data set. The correlation between the predicted values and ground truth values is shown in Figure 4a.

3.1.2. Hydration Free Energy. Figure 4b shows the correlation of predicted and ground truth ΔG_{Hyd} in the test set obtained from the FreeSolv data set. The MPNN model succeeds in achieving a high degree of accuracy with a root mean squared error (RMSE) of 1.35 kcal/mol and close correlation characterized by the R^2 score of 0.87. However, the use of machine-learning models for predicting properties of a molecule comes with the caveat that in the case in which any substructures in the molecule are not present in the training set, the prediction may be inaccurate. This is especially true in regions where the existing data is very sparse like cases when the binding affinity is extremely high. As shown in the subsequent sections, performing docking calculations on the generated molecules addresses this issue.

3.2. Single Objective Optimization.

The initial tests were performed to analyze the ability of the generator model based only on SMILES to learn to generate molecules with structure complementary to the binding pocket. In sections 3.2.1 and 3.2.2, we evaluate two different approaches to obtaining the binding affinity, namely docking calculations and using a GIN predictor model, respectively. To further validate the performance, the optimization was done from scratch across three runs with different random seeds, and the distribution of the properties of the generated molecules is given in the Supporting Information. Refer to Section S4 of the Supporting Information for the statistics and generated molecules from each of the trials in Section 3.2.

3.2.1. Docking Calculations. For both TTBK1 and SARS-CoV-2 M_{pro}, the generator model was optimized for 175 iterations with 15 policy gradient steps in each and a batch of 10 molecules. At the end of every iteration, 100 molecules were



(a) BA predicted by GIN vs. ground truth values from AutoDock

(b) ΔG_{Hyd} predicted by MPNN vs. ground truth values from FreeSolv

Figure 4. Correlation plots between predicted values from the machine-learning models and ground truth values from the respective data sets.

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

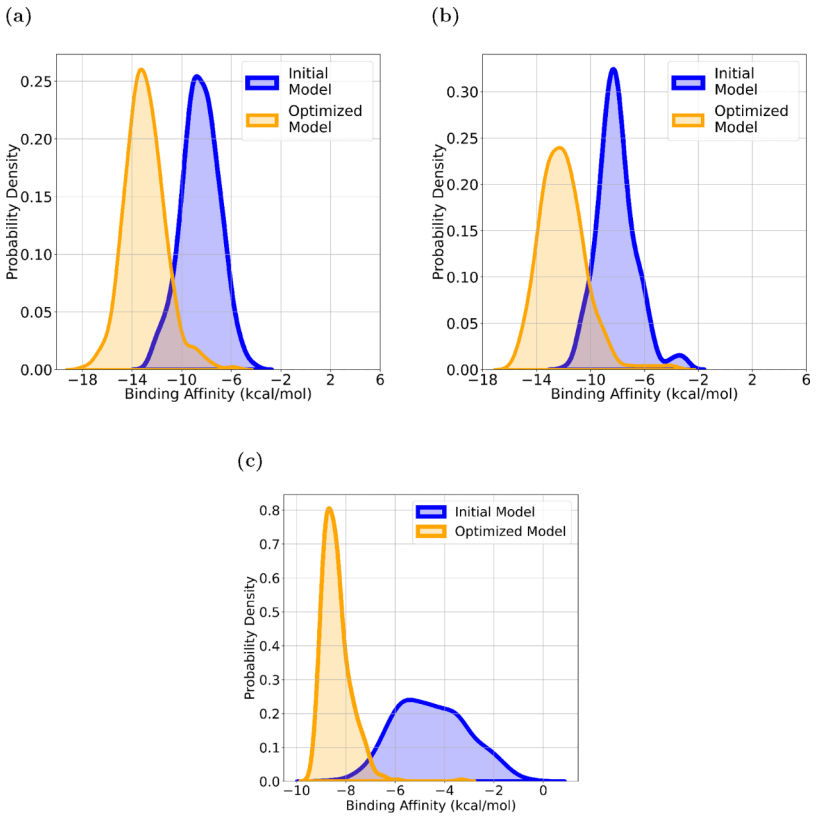


Figure 5. Distribution of BA of generated molecules. BA of generated molecules before and after optimizing the generator for reward from BA with (a) SARS-CoV-2 M_{pro}, (b) TTBK1, and (c) TTBK1 calculated using GIN.

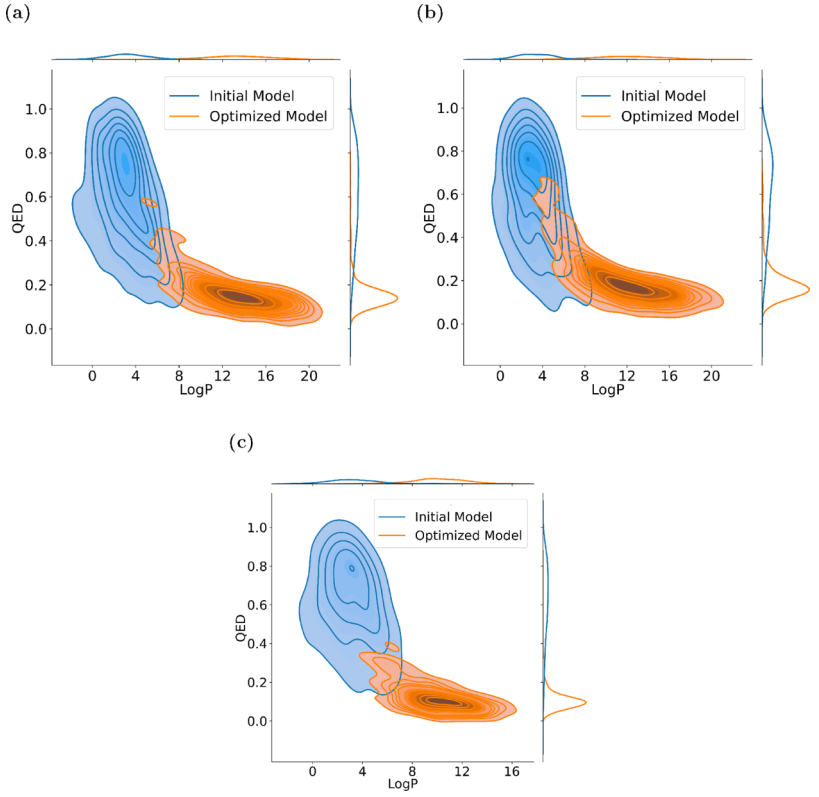


Figure 6. Joint distribution of LogP and QED of generated molecules before and after optimizing the model for reward from BA with (a) SARS-CoV-2 M_{pro}, (b) TTBK1, and (c) TTBK1 calculated using GIN.

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

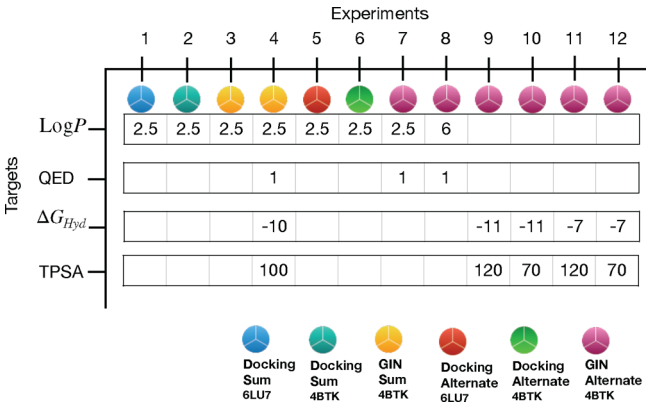


Figure 7. Tests performed for multiobjective optimization: protein PDB ID, tools used for BA calculation, different target values of respective properties, and optimization strategies. ΔG_{Hyd} and TPSA are in kcal/mol and Å^2, respectively.

generated, and their docking scores were calculated. After the completion of 175 iterations, 500 molecules were generated from the initial model and the optimized model. The distribution of the binding affinities given in Figures 5a and 5b shows a significant shift toward more desirable regions.

The above approach shows great performance in optimization for BA, but typical druglike molecules have constraints on other properties as well. Ideally, LogP should be between 0 and 5 for oral drugs, QED[60] close to 1, and TPSA < 90 Å^2, which are violated during single objective optimization (Figures 6a and 6b). Hence, there is a need for multiobjective optimization in order to keep the other properties in check as well.

3.2.2. Using GIN. The use of a GIN for BA prediction leads to an approximately 10-fold speed-up in optimization taking only about 5 h while still keeping the high performance. To further validate the generator, 500 molecules were generated and docked to TTBK1. The shifts in distribution of BA, LogP, and QED are shown in Figures 5c and 6c. However, the undesirable LogP and QED persist, and hence multiobjective optimization is used.

3.3. Multiobjective Optimization. In order to address the shortcomings of single objective optimization, the rewards from different properties were also integrated into the policy gradient calculation, and the performance was tested for the two proteins using different target values and both optimization strategies. These are listed in Figure 7, and the results have been discussed in the subsequent sections. Section 3.3.1 describes the conventional method for multiobjective optimization by taking a weighted sum of rewards from different properties, and Section 3.3.2 shows the performance of MoleGuLAR in generating molecules with specified properties. To further validate the performance, the optimization was done from scratch across three runs with different random seeds, and the distribution of the properties of the generated molecules is discussed in the Supporting Information. Refer to Section S5 of the Supporting Information for the statistics and generated molecules from each of the tests in Section 3.3.

3.3.1. Weighted Sum Reward. Tests were done keeping the weights for each reward function equal to 1 in order to optimize the BA calculated using AutoDock along with target LogP = 2.5.

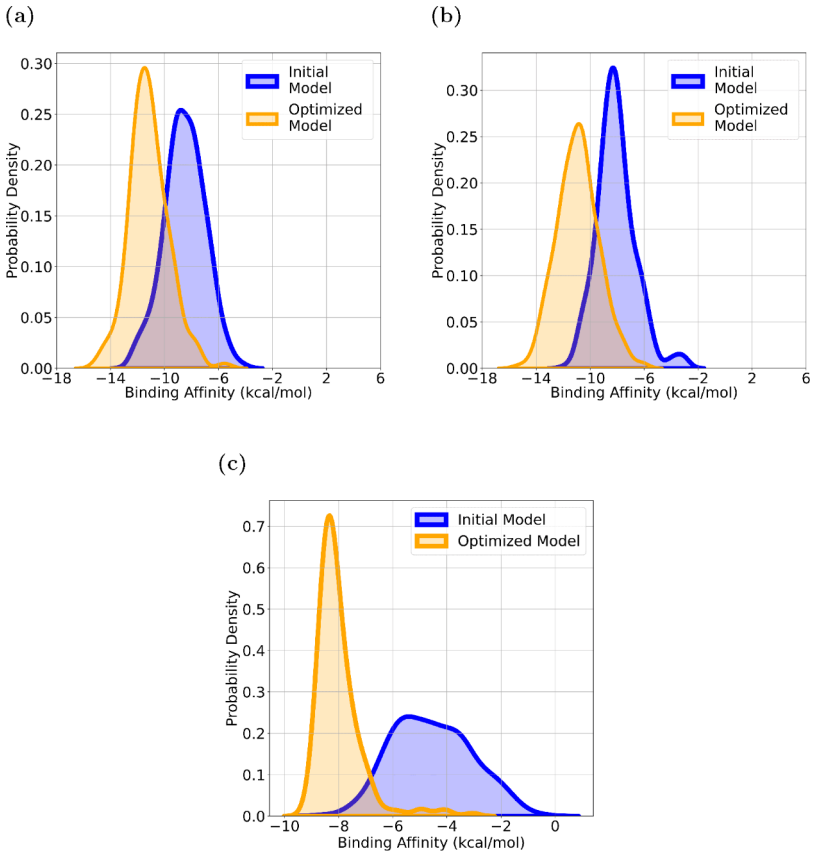


Figure 8. Distribution of BA of generated molecules before and after optimizing the generator for sum of rewards from target LogP = 2.5 and high BA calculated with (a) SARS-CoV-2 M_{pro}, (b) TTBK1, and (c) TTBK1 calculated using GIN.

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

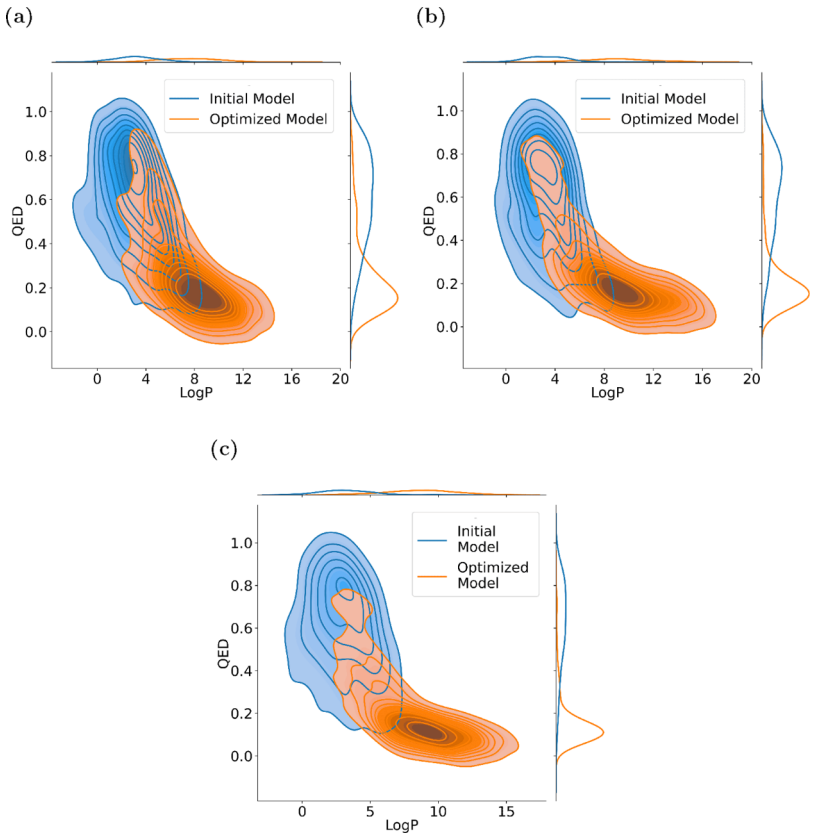


Figure 9. Joint distribution of LogP and QED of generated molecules before and after optimizing the generator for sum of rewards from target LogP = 2.5 and high BA with (a) SARS-CoV-2 M_{pro}, (b) TTBK1, and (c) TTBK1 calculated using GIN.

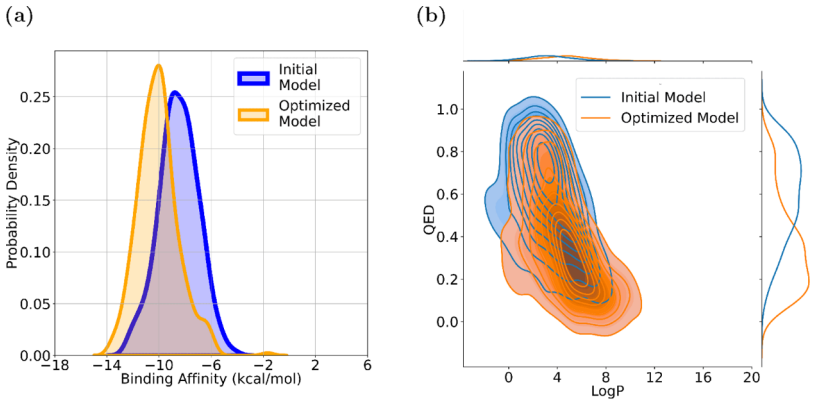


Figure 10. (a) Distribution of BA and (b) joint distribution of LogP and QED before and after optimizing the model for LogP = 2.5 and high BA with SARS-CoV-2 M_{pro} calculated using AutoDock by alternating rewards.

While there was improvement in the distribution of LogP in comparison to single objective optimization and BA in comparison to the initial model (Figure 8), the target was not achieved yet (Figure 9). A similar observation was seen when GIN was used for BA calculation instead of AutoDock in the same setting (Figures 8c and 9c). Further testing was done using the GIN BA predictor (Figure S16 of the Supporting Information), in which the generator was optimized to generate molecules with various simultaneous targets, i.e., LogP = 2.5, maximum QED, TPSA = 100 Å^2, and ΔG_{Hyd} = −10 kcal/mol, and the weighted sum of all rewards was taken. This worked well for BA, TPSA, and ΔG_{Hyd} but failed to achieve the target LogP and showed a very low QED. This is in agreement with the OptiMol pipeline by Boitreaud et al.[42] who showed that optimizing for BA led to a reduction in QED.[42] In order to tackle this, we propose the following alternating rewards strategy for optimization.

3.3.2. Alternating Rewards. The pipeline's exceptional performance on single objective tasks helped formulate the strategy that only one objective be optimized at a time and the objective be changed at regular intervals. Taking the example of LogP and BA, initially the model moves to generating molecules with better BA, but after a few iterations, the reward is switched to optimize for LogP. Figure S19 of the Supporting Information shows the variation of BA with SARS-CoV-2 M_{pro} and LogP with the iterations. The generator is rewarded for BA during the iterations marked red and for LogP during the iterations marked blue, and it can be seen that when the reward switches the model

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

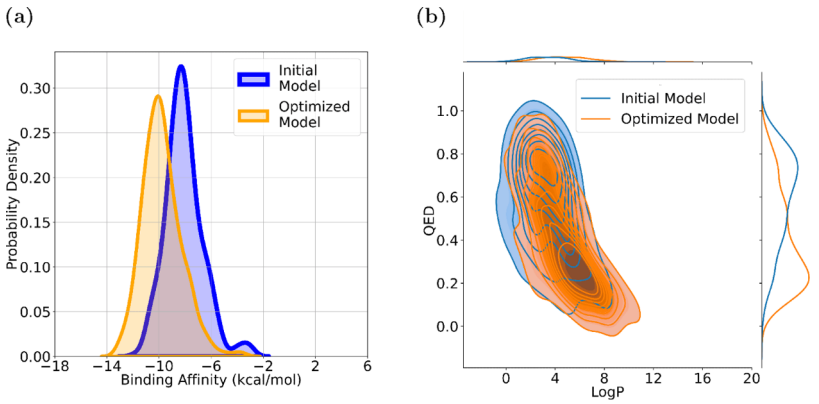


Figure 11. (a) Distribution of BA and (b) joint distribution of LogP and QED before and after optimizing the model for LogP = 2.5 and high BA with TTBK1 calculated using AutoDock by alternating rewards.

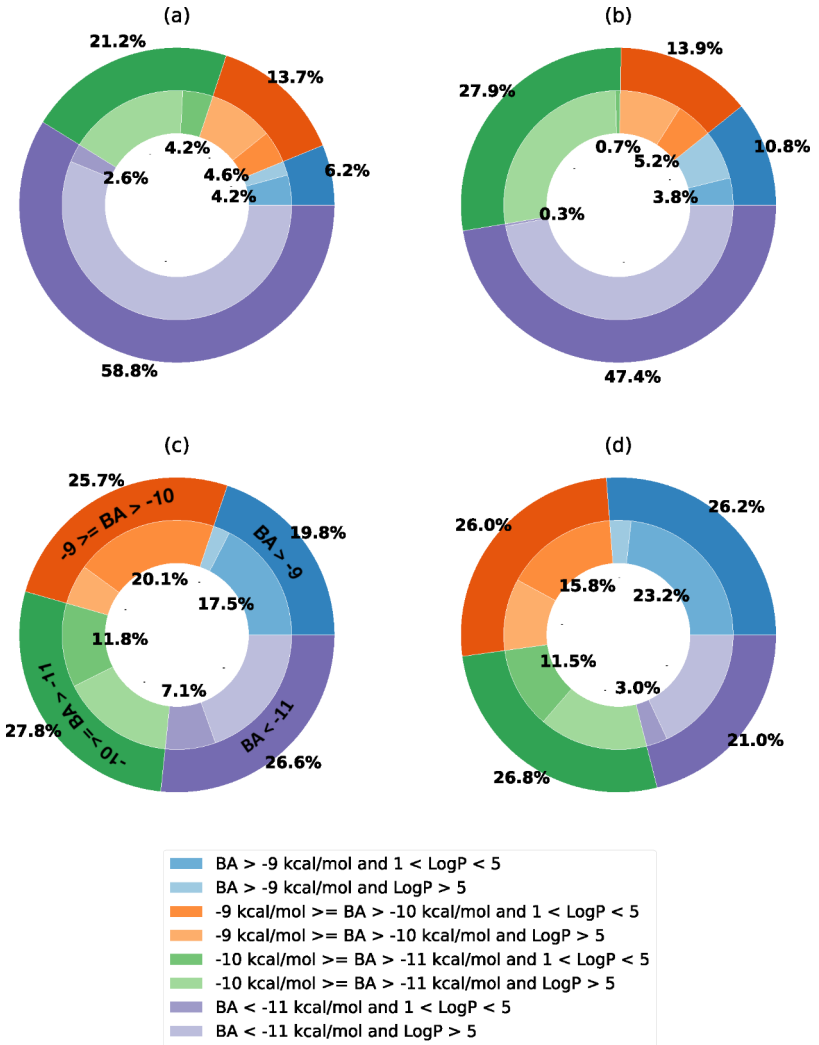


Figure 12. Fraction of molecules with BA and LogP in and out of desirable regions from those generated by the model optimized for (a) sum of rewards from BA with SARS-CoV-2 M_{pro} and LogP, (b) sum of rewards from BA with TTBK1 and LogP, (c) alternating rewards from BA with SARS-CoV-2 M_{pro} and LogP, and (d) alternating rewards from BA with TTBK1 and LogP. For legends of the outermost part of the pie chart, see (c), as they repeat in the rest of the pie charts.

is already sampling from the space with high BA and moves toward the region close to the target LogP and vice versa. Figures 10 and 11 show the application of this strategy to SARS-CoV-2 M_{pro} and TTBK1, respectively, using AutoDock. We can see a better distribution for BA as well as a significant overlap in the most desirable and optimized regions of LogP and QED. Furthermore, Figure 12 shows the ability of the current strategy to consistently generate a higher percentage of hit molecules in comparison to the weighted sum approach.

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

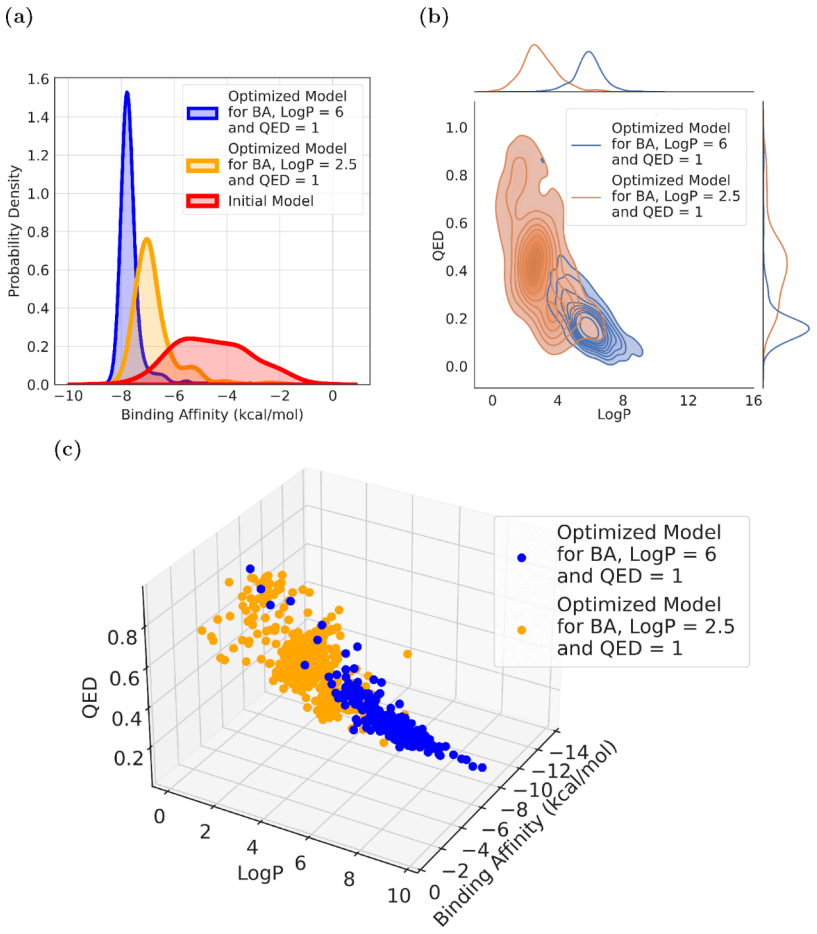


Figure 13. (a) Distribution of BA, (b) joint distribution of LogP and QED, and (c) 3D representation of properties of generated molecules before and after optimizing the generator for high BA with TTBK1 calculated using GIN.

Table 2. LogP and QED Targets along with Obtained Mean Values of BA, LogP, and QED of the Corresponding Generated Data as Well as the Best BA

target LogP	target QED	mean BA (kcal/mol)	best BA (kcal/mol)	mean LogP	mean QED
2.5	1	−6.76	−8.18	2.9	0.42
6	1	−7.64	−8.41	5.87	0.19

Table 3. TPSA and ΔG_{Hyd} Targets along with Mean Values of BA, TPSA, and ΔG_{Hyd} of the Corresponding Generated Data as Well as the Best BA

target TPSA (Å^2)	target ΔG_{Hyd} (kcal/mol)	mean TPSA (Å^2)	mean ΔG_{Hyd} (kcal/mol)	mean BA (kcal/mol)	best BA (kcal/mol)
70	−11	88.77	−10.13	−6.11	−8.36
120	−11	117.25	−10.75	−6.65	−8.32
70	−7	71.64	−7.42	−7.4	−8.90
120	−7	99.16	−8.42	−6.85	−8.64

The GIN model was used for further testing. Different targets were kept for different properties to evaluate the model's capability of achieving all targets simultaneously. In Figure 13 and Table 2, it can be seen that the model is capable of generating molecules with high BA along with maximizing QED subject to the LogP being constrained to 2.5 and 6. Furthermore, in Figure 13c, there is a clear separation of the distributions in three dimensions showing the model's ability to navigate different regions of the chemical space where molecules possess the desired properties.

A similar test was repeated with TPSA and ΔG_{Hyd} along with BA to see how the optimization strategy handles conflicting targets since higher the TPSA, the more negative the ΔG_{Hyd}. The target pairs are shown in Table 3, and the obtained results are discussed in Section S5.2.5 of the Supporting Information.

The best hits from 500 molecules generated from each of the aforementioned tests using alternating rewards are shown in Figure 14.

4. CONCLUSION

In this study, we propose MoleGuLAR, a pipeline for de novo generation of druglike molecules with high BA to novel targets along with other desirable properties using alternating rewards. Reinforcement learning is used to optimize the generator model weights to maximize the rewards obtained from calculated properties. A novel optimization strategy is also proposed for the multiobjective setup in which the reward function is switched to optimize for a different property at regular intervals instead of the conventional approach in which the sum of rewards from all properties is taken. We also show the performance of two ways of calculating BA, i.e., using AutoDock and using a predictor model, while also weighing the merits and demerits of both approaches as a part of the pipeline. Further work can include training the BA predictor models on the fly using techniques like active learning to make the pipeline more robust and efficient. The use of this architecture significantly reduces the number of

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

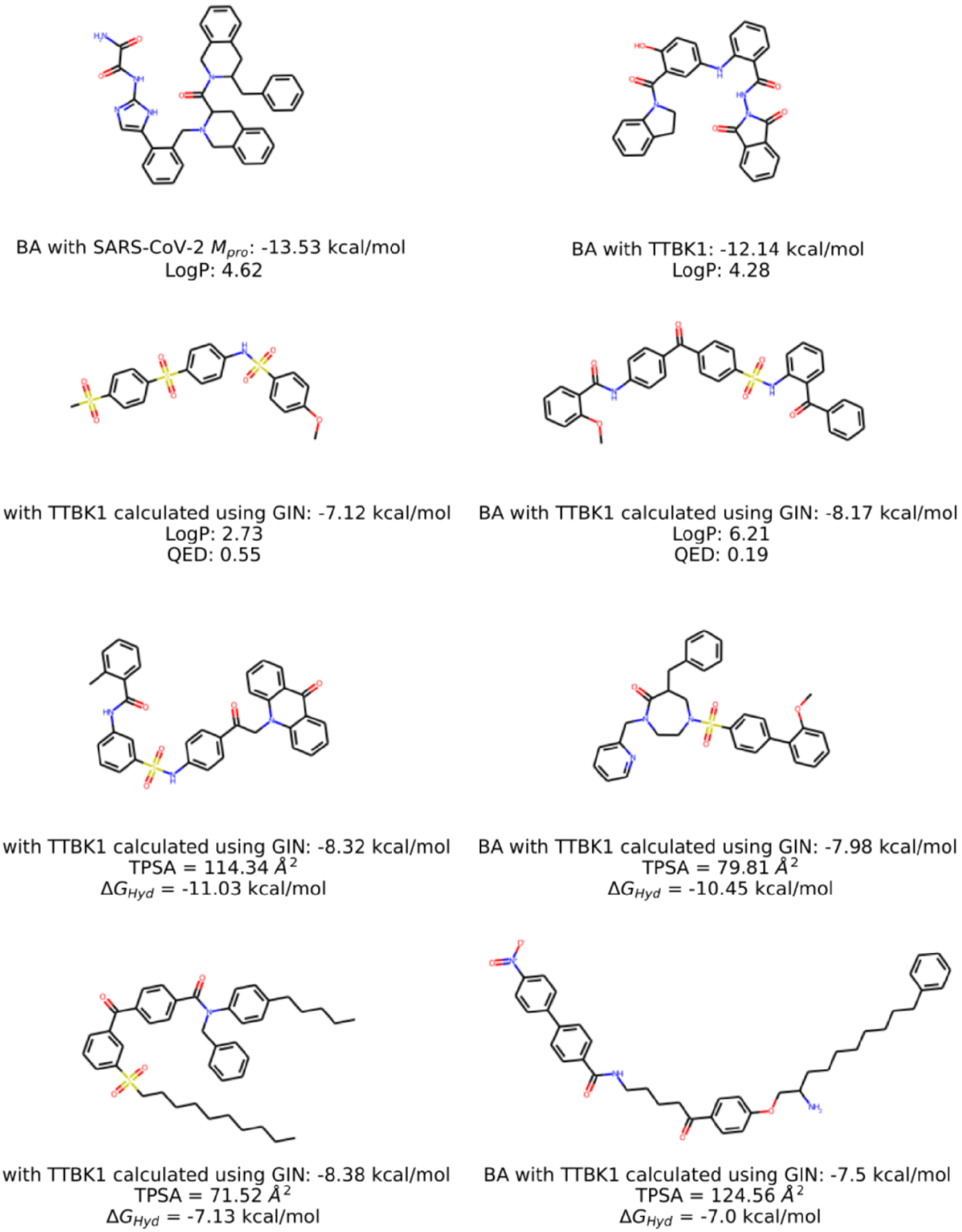


Figure 14. Top hits from 500 molecules generated after each test done using alternating rewards.

docking calculations required to identify potential drugs for a novel target removing a major bottleneck in the drug discovery process and can potentially be used to generate targeted drug libraries. We show that the alternating reward strategy is extremely robust in finding potential hits for the target across a wide set of target properties.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01341.

Performance of different machine-learning models for predicting BA, reward functions for each property, tables of mean values of respective properties of molecules

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

generated for each test along with structures of the top hits, protein−ligand interactions, and supplementary discussions and methods (PDF)

AUTHOR INFORMATION

Corresponding Author

U. Deva Priyakumar − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India;* orcid.org/0000-0001-7114-3955; Email: deva@iiit.ac.in

Authors

Manan Goel − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

Shampa Raghunathan − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India; École Centrale School of Engineering, Mahindra University, Hyderabad 500 043, India*

Siddhartha Laghuvarapu − *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c01341

Notes

The authors declare no competing financial interest.

The source code, an extension of the work by Popova and co-workers (https://github.com/isayev/ReLeaSE), along with documentation and models optimized for each test are available at https://github.com/devalab/MoleGuLAR.

ACKNOWLEDGMENTS

We acknowledge the financial support through the DST-SERB grant (no. CVD/2020/000 343), DST/WOS-A grant (no. SR/WOS-A/CS-19/2 018 (G)), and IHub-Data, IIIT Hyderabad. This work was partially funded by Intel Corp. as part of its Pandemic Response Technology Initiative (PRTI).

REFERENCES

(1) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *Nature* **2018**, *559*, 547−555.

(2) Sadowski, P.; Baldi, P. *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*; Springer: New York, 2018; pp 269−297.

(3) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.

(4) Irwin, J. J.; Shoichet, B. K. ZINC−A free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177−182.

(5) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Felix, E.; Magariños, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47*, D930−D940.

(6) Senior, A. W.; Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Žídek, A.; Nelson, A. W. R.; Bridgland, A.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706−710.

(7) Segler, M. H.; Waller, M. P. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chem. - Eur. J.* **2017**, *23*, 5966−5971.

(8) Schreck, J. S.; Coley, C. W.; Bishop, K. J. Learning retrosynthetic planning through simulated experience. *ACS Cent. Sci.* **2019**, *5*, 970−981.

(9) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. *Neural message passing for quantum chemistry*; International Conference on Machine Learning, 2017; pp 12631272

(10) Szymański, P.; Markowicz, M.; Mikiciuk-Olasik, E. Adaptation of high-throughput screening in drug discovery—toxicological screening tests. *Int. J. Mol. Sci.* **2012**, *13*, 427−452.

(11) Broach, J. R.; Thorner, J. High-throughput screening for drug discovery. *Nature* **1996**, *384*, 14−16.

(12) Maia, E. H. B.; Assis, L. C.; de Oliveira, T. A.; da Silva, A. M.; Taranto, A. G. Structure-based virtual screening: From classical to artificial intelligence. *Front. Chem.* **2020**, *8*..

(13) Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E. W. Computational methods in drug discovery. *Pharmacol. Rev.* **2014**, *66*, 334−395.

(14) Aggarwal, R.; Gupta, A.; Chelur, V.; Jawahar, C. V.; Priyakumar, U. D. DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks. *J. Chem. Inf. Model.* **2021**..

(15) Chelur, V.; Priyakumar, U. D. *BiRDS-Binding Residue Detection from Protein Sequences using Deep ResNets*; **2021**.

(16) Mehta, S.; Laghuvarapu, S.; Pathak, Y.; Sethi, A.; Alvala, M.; Priyakumar, U. D. MEMES: Machine learning framework for Enhanced MolEcular Screening. *Chem. Sci.* **2021**, *12*, 11710.

(17) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224−229.

(18) Reymond, J.-L. The chemical space project. *Acc. Chem. Res.* **2015**, *48*, 722−730.

(19) Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative recurrent networks for de novo drug design. *Mol. Inf.* **2018**, *37*, 1700111.

(20) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.

(21) De Cao, N.; Kipf, T. *MolGAN: An implicit generative model for small molecular graphs*; arXiv preprint; arXiv:1805.11973, **2018**.

(22) Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminf.* **2019**, *11*, 1−13.

(23) Maziarka, Ł.; Pocha, A.; Kaczmarczyk, J.; Rataj, K.; Danel, T.; Warchoł, M. Mol-CycleGAN: a generative model for molecular optimization. *J. Cheminf.* **2020**, *12*, 1−18.

(24) Jin, W.; Barzilay, R.; Jaakkola, T. *Junction tree variational autoencoder for molecular graph generation*; International Conference on Machine Learning, 2018; pp 2323−2332.

(25) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. *Grammar variational autoencoder*; International Conference on Machine Learning, 2017; pp 1945−1954.

(26) Griffiths, R.-R.; Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **2020**, *11*, 577−586.

(27) Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Cheminf.* **2018**, *10*, 1−9.

(28) Kell, D. B.; Samanta, S.; Swainston, N. Deep learning and generative methods in cheminformatics and chemical biology: navigating small molecule space intelligently. *Biochem. J.* **2020**, *477*, 4559−4580.

(29) Jiménez-Luna, J.; Grisoni, F.; Weskamp, N.; Schneider, G. Artificial intelligence in drug discovery: Recent advances and future perspectives. *Expert Opin. Drug Discovery* **2021**, *16*, 949−959.

(30) Bagal, V.; Aggarwal, R.; Vinod, P.; Priyakumar, U. D. *LigGPT: Molecular Generation using a Transformer-Decoder Model*; chemrxiv preprint; **2021**. DOI: 10.26434/chemrxiv.14561901.v1.

(31) Winter, R.; Montanari, F.; Noé, F.; Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating

equivalent chemical representations. *Chemical science* **2019**, *10*, 1692−1701.
(32) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.
(33) Koge, D.; Ono, N.; Huang, M.; Altaf-Ul-Amin, M.; Kanaya, S. Embedding of Molecular Structure Using Molecular Hypergraph Variational Autoencoder with Metric Learning. *Mol. Inf.* **2021**, *40*, 2000203.
(34) Tavakoli, M.; Baldi, P. Continuous Representation of Molecules using Graph Variational Autoencoder. *Proceedings of the AAAI 2020 Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physical Sciences*, Stanford, CA, March 23−25, 2020.
(35) Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Noé, F.; Clevert, D.-A. Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci.* **2019**, *10*, 8016−8024.
(36) Korovina, K.; Xu, S.; Kandasamy, K.; Neiswanger, W.; Poczos, B.; Schneider, J.; Xing, E. *Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations*; International Conference on Artificial Intelligence and Statistics, 2020; pp 3393−3403.
(37) Gao, K.; Nguyen, D. D.; Tu, M.; Wei, G.-W. Generative network complex for the automated generation of drug-like molecules. *J. Chem. Inf. Model.* **2020**, *60*, 5682−5698.
(38) Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, No. eaap7885.
(39) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. *Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models*; arXiv preprint; arXiv:1705.10843, **2017**.
(40) You, J.; Liu, B.; Ying, R.; Pande, V.; Leskovec, J. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Red Hook, NY, 2018; pp 6412−6422.
(41) Khemchandani, Y.; O'Hagan, S.; Samanta, S.; Swainston, N.; Roberts, T. J.; Bollegala, D.; Kell, D. B. DeepGraphMolGen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *J. Cheminf.* **2020**, *12*, 1−17.
(42) Boitreaud, J.; Mallet, V.; Oliver, C.; Waldispuhl, J. Optimol: Optimization of binding affinities in chemical space for drug discovery. *J. Chem. Inf. Model.* **2020**, *60*, 5658−5666.
(43) Born, J.; Manica, M.; Oskooei, A.; Cadow, J.; Markert, G.; Martínez, M. R. PaccMannRL: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. *Iscience* **2021**, *24*, 102269−269.
(44) Born, J.; Manica, M.; Cadow, J.; Markert, G.; Mill, N. A.; Filipavicius, M.; Janakarajan, N.; Cardinale, A.; Laino, T.; Martínez, M. R. Data-driven molecular design for discovery and synthesis of novel ligands: a case study on SARS-CoV-2. *Mach. Learn.: Sci. Technol.* **2021**, *2*, 025−024.
(45) Bung, N.; Krishnan, S. R.; Bulusu, G.; Roy, A. De novo design of new chemical entities for SARS-CoV-2 using artificial intelligence. *Future Med. Chem.* **2021**, *13*, 575−585.
(46) Krishnan, S. R.; Bung, N.; Bulusu, G.; Roy, A. Accelerating de novo drug design against novel proteins using deep learning. *J. Chem. Inf. Model.* **2021**, *61*, 621−630.
(47) Joulin, A.; Mikolov, T. Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Cambridge, MA, 2015; Vol. *1*, pp 190−198.
(48) Landrum, G. *RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling*; 2013.
(49) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The protein data bank. *Nucleic acids research* **2000**, *28*, 235−242.
(50) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785−2791.
(51) Santos-Martins, D.; Solis-Vasquez, L.; Tillack, A. F.; Sanner, M. F.; Koch, A.; Forli, S. Accelerating AutoDock4 with GPUs and gradient-based local search. *J. Chem. Theory Comput.* **2021**, *17*, 1060−1073.
(52) HTS Collection. https://enamine.net/compound-collections/screening-collection/hts-collection (accessed on 11/23/2021).
(53) Jaeger, S.; Fulle, S.; Turk, S. Mol2vec: unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **2018**, *58*, 27−35.
(54) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. *How Powerful are Graph Neural Networks?* 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, May 6−9, 2019.
(55) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V. S.; Leskovec, J. *Strategies for Pre-training Graph Neural Networks*; 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26−30, 2020.
(56) Pathak, Y.; Mehta, S.; Priyakumar, U. D. Learning Atomic Interactions through Solvation Free Energy Prediction Using Graph Neural Networks. *J. Chem. Inf. Model.* **2021**, *61*, 689−698.
(57) Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* **1992**, *8*, 229−256.
(58) Zhang, L.; Lin, D.; Sun, X.; Curth, U.; Drosten, C.; Sauerhering, L.; Becker, S.; Rox, K.; Hilgenfeld, R. Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science* **2020**, *368*, 409−412.
(59) Sato, S.; Cerny, R. L.; Buescher, J. L.; Ikezu, T. Tau-tubulin kinase 1 (TTBK1), a neuron-specific tau kinase candidate, is involved in tau phosphorylation and aggregation. *J. Neurochem.* **2006**, *98*, 1573−1584.
(60) Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90−98.

https://doi.org/10.1021/acs.jcim.1c01341
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX