Title

Spatial Graph Attention and Curiosity-driven Policy for Antiviral Drug Discovery

Permalink

https://escholarship.org/uc/item/47n2m7z4

Authors

Wu, Yulun
Cashman, Mikaela
Choma, Nicholas
et al.

Publication Date

2021-06-03

Peer reviewed

Spatial Graph Attention and Curiosity-driven Policy for Antiviral Drug Discovery

Yulun Wu [† §]
yulun_wu@berkeley.edu

Nicholas Choma [‡ §]
njchoma@lbl.gov

Andrew Chen [‡ §]
adchen@lbl.gov

Mikaela Cashman [∥ §]
cashmanmm@ornl.gov

Érica T. Prates [∥ §]
teixeiraprae@ornl.gov

Manesh Shah [¶]
shahmb@ornl.gov

Verónica G. Melesse Vergara [∥ ¶]
vergaravg@ornl.gov

Austin Clyde [††]
aclyde@uchicago.edu

Thomas S. Brettin [‡‡]
brettin@anl.gov

Wibe A. de Jong [‡ §]
wadejong@lbl.gov

Neeraj Kumar [§§ §]
neeraj.kumar@pnnl.gov

Martha S. Head [∥ §]
headms@ornl.gov

Rick L. Stevens [†† ‡‡]
stevens@anl.gov

Peter Nugent [‡ §]
penugent@lbl.gov

Daniel A. Jacobson [∥ ¶ §]
jacobsonda@ornl.gov

James B. Brown [† ‡ §]
jbbrown@lbl.gov

Abstract

We developed Distilled Graph Attention Policy Networks (DGAPNs), a curiosity-driven reinforcement learning model to generate novel graph-structured chemical representations that optimize user-defined objectives by efficiently navigating a physically constrained domain. The framework is examined on the task of generating molecules that are designed to bind, noncovalently, to functional sites of SARS-CoV-2 proteins. We present a spatial Graph Attention Network (sGAT) that leverages self-attention over both node and edge attributes as well as encoding spatial structure — this capability is of considerable interest in areas such as molecular and synthetic biology and drug discovery. An attentional policy network is then introduced to learn decision rules for a dynamic, fragment-based chemical environment, and state-of-the-art policy gradient techniques are employed to train the network with enhanced stability. Exploration is efficiently encouraged by incorporating innovation reward bonuses learned and proposed by random network distillation. In experiments, our framework achieved outstanding results compared to state-of-the-art algorithms, while increasing the diversity of proposed molecules and reducing the complexity of paths to chemical synthesis.

† University of California, Berkeley,   ‡ Lawrence Berkeley National Laboratory,   § National Virtual Biotechnology Laboratory, US Department of Energy,   ∥ Oak Ridge National Laboratory,   ¶ University of Tennessee, Knoxville,   †† University of Chicago,   ‡‡ Argonne National Laboratory,   §§ Pacific Northwest National Laboratory

1 Introduction

This work aims to address the challenge of establishing an automated process for the design of objects with connected components, such as molecules, that optimize specific properties. Achieving this goal is particularly desirable in drug development and materials science, where manual discovery remains a time-consuming and expensive process [19, 46]. However, there are two major difficulties that have long impeded rapid progress. Firstly, the chemical space is discrete and massive [41], presenting a complicated environment for an Artificial Intelligence (AI) approach to efficiently and effectively explore. Secondly, it is not trivial to compress such connected objects into feature representations that preserve most of their information, while also being highly computable for Deep Learning (DL) methods to exploit.

We introduce Distilled Graph Attention Policy Networks (DGAPN), a framework that advances prior work in addressing both of these challenges. We present a Reinforcement Learning (RL) architecture that is efficiently encouraged to take innovative actions and an environment that is able to construct chemically valid fragment-based action space. We also propose a hybrid Graph Neural Network (GNN) that encodes graph-structured objects' spatial structure and edge attributes in addition to adjacency structure and node attributes. The following paragraphs discuss how we addressed limitations of prior work and its relevance to the discovery of antiviral drugs. For more descriptions of key prior methodologies that we used as benchmarks in this paper, see Section 5.

Graph Representation Learning   String representation of molecules acquired by the simplified molecular-input line-entry system (SMILES) [55] have been widely used in attempts to generate molecules of interest [15, 37, 42, 56, 28, 13, 7, 49, 23]. Despite being spatially efficient, string representations suffer from significant information loss and poor robustness [31]. Graph representations have become predominant and preferable for their ability to efficiently encode an object's scaffold structure and attributes. Graph representations are particularly ideal for RL since intermediate representations can be decoded and evaluated for reward assignments. While recent GNNs such as Graph Convolutional Networks (GCNs, [26]) and Graph Attention Networks (GATs, [53]) have demonstrated impressive performance on many DL tasks, further exploitation into richer information contained in graph-structured data is needed to faithfully represent the complexity of chemical space [34, 54, 4, 1]. Two key limitations of prior work are (1) information embedding is heavily node-dominant and (2) graph's spatial information, which may be informative, or essential, in molecular biology, is largely discarded. Distinct from [8] and [5], we extended GATs to be edge-featured, while still node-centric for efficiency, and perform weighted convolution with a sparsified distance matrix to capture the spatial structure.

Reinforcement Learning   A variety of graph generative models have been used in prior work, predominantly Variational Autoencoders (VAEs) [50, 44, 32, 33] and Generative Adversarial Networks (GANs) [9]. While some of these have a recurrent structure [30, 59], RL algorithms excel in sequential generation due to their ability to interact dynamically with the environment and resist overfitting on training data. Both policy learning [58] and value function learning [61] have been adopted for molecule generation: however, the majority generate molecules node-by-node and edge-by-edge. In comparison, an action space consisting of molecular fragments, i.e., a collection of chemically valid components and realizable synthesis paths, is favorable since different atom types and bonds are defined by the local molecular environment. Fragment-by-fragment sequential generation has been used in VAE [21], but has not been utilized in a graph RL framework. In this work, we designed our environment with the Chemically Reasonable Mutations (CReM, [40]) library to realize a valid fragment-based action space. Furthermore, we addressed mode collapse which commonly occurs in both policy gradient and Q-learning algorithms by employing an efficient exploration technique, adapting Random Network Distillation (RND, [3]) to GNNs and thus encouraging curiosity-driven policies through a surrogate innovation reward for each intermediate state.

Antiviral Drug Discovery — A Timely Challenge   The severity of the COVID-19 pandemic highlighted the major role of computational workflows to characterize the viral machinery and identify druggable targets for the rapid development of novel antivirals. Particularly, the synergistic use of DL methods and structural knowledge is at the cutting edge of molecular biology — consolidating such integrative protocols to accelerate drug discovery is of paramount importance [57, 20]. Here we experimentally examined our architecture on the task of discovering novel inhibitors targeting the

SARS-CoV-2 non-structural protein endoribonuclease (NSP15), which is critical for viral evasion of host defense systems [39]. Our framework combines AI with structure-based methods to enable a broad and efficient search for drug-like inhibitors on the chemical space, without being restricted to existing databases of compounds. Structural information about the putative protein-ligand complexes was integrated into this framework with AutoDock-GPU [45], which leverages the GPU resources from leadership-class computing facilities, including the Summit supercomputer, for high-throughput molecular docking [29]. We show that our results outperformed state-of-the-art generation models in finding a chemically diverse set of molecules with high affinity to the target and the best synthetic accessibility score among state of the art models.

2   Problem Formulations

Our goal is to establish a set of decision rules to generate graph-structured data that maximizes a compound objective under certain constraints. In the case of molecular graphs, we compose the objective by weighing over multiple factors, including chemical properties, innovation, and entropy. The constraints are the set of rules under which the chemical validity of generated molecules is ensured. The process of generating each individual molecule is sequential and concludes after certain user-defined criteria are met.

Similar to prior formulations, the generating process is defined as a time homogeneous Markov Decision Process (MDP). We give a formal definition of this process in the Appendix. Under this setting, the action policies and state transition dynamics at step t can be factorized according to the Markov property:

$$P(a_t|s_0, a_0, s_1, a_1, \ldots, s_t) = P(a_t|s_t) := \pi(a_t|s_t) \tag{1}$$

$$P(s_{t+1}|s_0, a_0, s_1, a_1, \ldots, s_t, a_t) = P(s_{t+1}|s_t, a_t) := \rho(s_{t+1}|s_t, a_t) \tag{2}$$

where $\{s_t, a_t\}_t$ are state-action sequences. A reward function $r(s, a)$ is used to assess an action a taken at a given state s. The process terminates at an optional stopping time T and s_T is then proposed as the final product of the current generating cycle. We aim to estimate the optimal policy π in terms of a specified objective to be defined in Section 2.2.

2.1   Environment Settings

Single-atom or single-bond additions are often not realizable by known biochemical reactions. Rather than employing abstract architectures such as GANs to suggest synthetic accessibility or drug-likeliness, which are known to be unstable and difficult to converge alongside policy training, we use the chemical library CReM [40] to construct our environment such that all next possible molecules can be obtained by one step of interchanging chemical fragments with our current molecule. This approach is considerably more reliable and interpretable compared to DL approaches. A detailed description of the CReM library can be found in the Appendix.

CReM Action   At each time step t, we use CReM to enumerate the next set of valid molecules $\boldsymbol{v}_{t+1}$ based on current molecule s_t. Under this setting, the transition dynamics are fixed, the underlying set A of the action space can be defined as equal to S of the state space, and action a_t is induced by the direct selection of next state s_{t+1}.

2.2   Objective

The objective is to maximize the expected reward of generated molecules. The reward is composed of two parts– a main reward assigned to the final product s_T, and innovation rewards assigned to the intermediate products $\{s_t\}_{1 \leq t \leq T}$. The relative importance between them can be specified based on individual interests. Below are the details of these two rewards in our experiments.

Docking Score   Docking scores are computed by docking programs that use the three-dimensional structure of the protein to predict the most stable bound conformations of the small molecules of interest, targeting a pre-defined functional site. An efficient GPU implementation of an automated docking tool is used in the experiments, see the Appendix for more details.
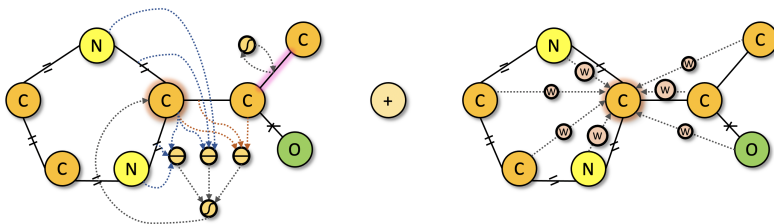
3

Figure 1: Spatial Graph Attention.

Curiosity Score Curiosity scores are generated by assessing the similarity of a given molecule to all the previously explored molecules (including intermediates) with two deep neural networks. The lower the similarity is, the higher the curiosity score. See Section 3.3 for details.

3 Proposed Method

3.1 Spatial Graph Attention

We introduce a hybrid GNN called Spatial Graph Attention (sGAT) to acquire feature vectors $\boldsymbol{h}_t \in \mathbb{R}^{d_h}$ representing states. Two different types of information graphs extracted from a connected object are heterogeneous and thus handled differently in forward passes as described in the following sections. The two hidden representations acquired respectively are aggregated at the end of each layer. See Figure 1 for an overview.

3.1.1 Graph Attention on an Attribution Graph

The attribution graph of a molecule with n atoms and e bonds is given by the triple $(\boldsymbol{A}, \boldsymbol{N}, \boldsymbol{E})$, where $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ is the node adjacency matrix, $\boldsymbol{N}$ is the node attribution matrix of dimension $n \times d_n$ and $\boldsymbol{E}$ is the edge attribution matrix of dimension $e \times d_e$. Each entry A_{ij} of $\boldsymbol{A}$ is 1 if a bond exists between atom i and j, and 0 otherwise. Each row vector $\boldsymbol{n}_i$ of $\boldsymbol{N}$ is a concatenation of the properties of atom i, including its atomic number, mass, etc., with the categorical properties being one-hot encoded. $\boldsymbol{E}$ is formed similarly with the bond attributes. We denote a row vector of $\boldsymbol{E}$ as $\boldsymbol{e}_{ij}$ when it corresponds to the bond between atom i and j.

We then define a multi-head forward propagation that handles richer graph information than the attention layers used in GATs [53]. Let $h_{\boldsymbol{n}_k} \in \mathbb{R}^{1 \times d_{h_n}}$ denote a given representation for $\boldsymbol{n}_k$, $h_{\boldsymbol{e}_{ij}} \in \mathbb{R}^{1 \times d_{h_e}}$ denote a given representation for $\boldsymbol{e}_{ij}$, then the m-th head attention α_{ij}^m from node j to node i ($i \neq j$) is given by

$$\alpha_{ij}^m = softmax_j \left(\mathop{\Big|\Big|}_{k: a_{ik}=1} \left\{ \sigma([h_{\boldsymbol{n}_i} \boldsymbol{W}_{n,m} \parallel h_{\boldsymbol{e}_{ik}} \boldsymbol{W}_{e,m} \parallel h_{\boldsymbol{n}_k} \boldsymbol{W}_{n,m}] \cdot att_m{}^T) \right\} \right) \quad (3)$$

where $\parallel$ stands for concatenation (here column-wise); σ is some non-linear activation; $\boldsymbol{W}_{n,m} \in \mathbb{R}^{d_{h_n} \times d_{w_n}}$, $\boldsymbol{W}_{e,m} \in \mathbb{R}^{d_{h_e} \times d_{w_e}}$ are the m-th head weight matrices for nodes and edges respectively; $att_m \in \mathbb{R}^{1 \times (2d_{w_n} + d_{w_e})}$ is the m-th head attention weight. The representations after a feed-forward operation are consequently given as follow:

$$h'_{\boldsymbol{n}_i} = aggr_{1 \leq m \leq n_m} \left\{ \sigma \left(\left(\sum_{j: A_{ij}=1} \alpha_{ij}^m \cdot h_{\boldsymbol{n}_j} + h_{\boldsymbol{n}_i} \right) \boldsymbol{W}_{n,m} \right) \right\} \quad (4)$$

$$h'_{\boldsymbol{e}_{ij}} = aggr_{1 \leq m \leq n_m} \left\{ \sigma \left(h_{\boldsymbol{e}_{ij}} \boldsymbol{W}_{e,m} \right) \right\} \quad (5)$$

where n_m is the total number of attention heads and $aggr$ denotes an aggregation method, most commonly $mean$, sum, or $concat$ [16]. In principle, a single-head operation on nodes is essentially graph convolution with the adjacency matrix $\hat{\boldsymbol{A}} = \widetilde{\boldsymbol{A}} + \boldsymbol{I}$ where $\widetilde{\boldsymbol{A}}$ is attention-regularized according to (3). This approach sufficiently encodes edge attributes while still being a node-centric convolution mechanism, for which efficient frameworks like Pytorch-Geometric [11] have been well established.
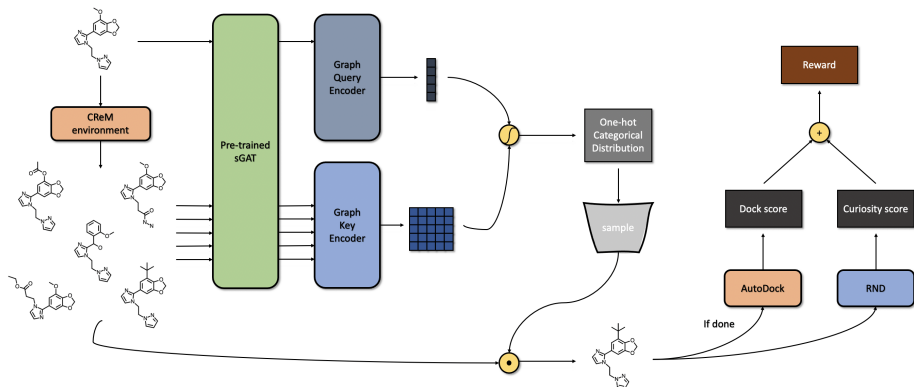
4

Figure 2: An overview of the Distilled Graph Attention Policy Network during a single step of the generating process.

Hypergraph Attention We also extended our attention mechanism to hypergraphs and provided support for the exploitation of molecular hyperstructure (e.g. [22]) if required. In this case, instead of an adjacency matrix we have a hyperedge index matrix $\boldsymbol{H} \in \{0,1\}^{e \times n}$ where each entry h_{ij} is 1 if hyperbond i connects atom j. A detailed description of the original hypergraph attention formulation can be found in [1]. While we anticipate that this feature will be useful in future design challenges, we do not bench mark it here, and leave this for future work.

3.1.2 Weighted Graph Convolution on Geometric Graphs

In addition to attributions and logical adjacency, one might also wish to make the most of molecule's spatial structure, as it informs the molecular volume and the spatial distribution of interaction sites — shape and chemical complementarity to the receptor binding site is essential for an effective association. On such structure, the convolution of feature vectors can be conducted by weighing over the Euclidean distance between atoms. Taking computational complexity into account, the following paragraph provides the details.

Spatial Convolution Let $\boldsymbol{G} = \left(d_{ij}^{-1}\right)_{i,j \leq n}$ be the inverse distance matrix where d_{ij} is the Euclidean distance between node i and j for $\forall i \neq j$, and $d_{ii}^{-1} := 0$. $\boldsymbol{G}$ can then be seen as an adjacency matrix with weighted "edge"s indicating nodes' spatial relations, and the forward propagation is thus given by

$$\boldsymbol{H}_n'' = \sigma \left(\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{G}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} + \boldsymbol{I} \right) \boldsymbol{H}_n \boldsymbol{W}_n \right) \tag{6}$$

where $\widetilde{\boldsymbol{G}}$ is optionally sparsified and attention-regularized from $\boldsymbol{G}$ to be described below; $\widetilde{\boldsymbol{D}} = diag_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} \widetilde{G}_{ij} \right\}$; $\boldsymbol{H}_n$ is the row concatenation of $\{h_{\boldsymbol{n}_i}\}_{1 \leq i \leq n}$; $\boldsymbol{W}_n \in \mathbb{R}^{d_{h_n} \times d_{w_n}}$ is the weight matrix; the identity matrix $\boldsymbol{I}$ is added after regularization since diagonal entries and the rest are heterogeneous. In reality, $\boldsymbol{G}$ induces $O(n)$ of convolution operations on each node and can drastically increase training time when the number of nodes is high. Therefore, one might want to derive $\widetilde{\boldsymbol{G}}$ by enforcing a cut-off around each node's neighborhood [38], or preserving an $O(n)$ number of largest entries in $\boldsymbol{G}$ and dropping out the rest. Depending on how abstract the spatial information is in real-world cases, $\boldsymbol{G}$ should be regularized by attention as described in the previous section. If attention is applied, spatial convolution is principally fully-connected graph attention with the Euclidean distance as a one-dimensional edge attribution.

3.2 Graph Attention Policy Network

In this section we introduce Graph Attention Policy Network (GAPN) that is tailored to environments that possess a dynamic range of actions. Note that $\rho(\cdot | s_t, a_t)$ is a degenerate distribution for fixed transition dynamics and the future trajectory $\boldsymbol{\tau} \sim p(S_{t+1}, S_{t+2}, \ldots | S_t = s_t)$ is strictly equal in distribution to $\boldsymbol{a} \sim \pi(A_t, A_{t+1}, \ldots | S_t = s_t)$, hence simplified as the latter in the following sections.

5

To learn the policy more effectively and efficiently, we tested pre-training an sGAT with an expert dataset and sharing its first n_g layers with the Graph Attention Policy Network (GAPN) in the later experiments. These shared layers are not updated during the training of RL. See Figure 2 for an overview.

3.2.1 Action Selection

At each time step t, we sample the next state s_{t+1} from a random variable S_{t+1} constructed by applying a retrieval-system-inspired attention mechanism [52]. In our experiments, the scaled dot-product attention behaved more unstable and achieved worse results than the final layer approach. Further, we are not convinced that the minimal efficiency improvement makes it worth replacing the latter. Thus, the construction of S_{t+1} is given by:

$$S_{t+1} \sim OHC\left\{ softmax\left(\underset{g \in \boldsymbol{g}_{t+1}}{\|} \{L_{final}(E_Q(g_t) \| E_K(g))\} \right) \right\} \cdot \boldsymbol{v}_{t+1} \tag{7}$$

where $OHC\{p_1, \ldots, p_{n_v}\}$ is a one-hot categorical distribution with n_v categories; $\|$ is column concatenation here; $g_t, \boldsymbol{g}_{t+1}$ are the embeddings for s_t and $\boldsymbol{v}_{t+1}$ acquired by graph pre-training; E_Q, E_K are two sGAT+MLP graph encoders with output feature dimension d_k; $L_{final} : \mathbb{R}^{b \times 2d_k} \to \mathbb{R}^b$ is the final feed-forward layer.

There could be a number of ways to determine stopping time T for the generating process. One of the most intuitive ways is to append s_t itself to $\boldsymbol{v}_{t+1}$, and terminate the generation if s_t is selected as s_{t+1}. In experiments, we let every trajectory reach a threshold for maximum time step (i.e. T is constant) for the following two reasons. Firstly, it encourages the process to navigate more effectively through the state/action space instead of taking meaningless steps or getting stuck in a cycle. Secondly, with parallelization (further described in Section 3.4), it enables sparse reward evaluations, which addresses a major performance bottleneck. Note that constant trajectory length is feasible because a threshold for the maximum time step can be significantly lower for fragment-based action space compared to node-by-node and edge-by-edge action spaces.

3.2.2 Actor-Critic Algorithm

For the purpose of obeying causal logic and reducing variance, advantages are frequently used instead of raw rewards in policy iterations. Applying an actor-critic algorithm on reward-to-go yields the following Q-function and value function:

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi\left[\sum_{t'=t}^{T} \gamma^{t'-t} \cdot r(S_{t'}, A_{t'}) \bigg| s_t, a_t \right] \tag{8}$$

$$V^\pi(s_t) = \mathbb{E}_\pi\left[Q^\pi(s_t, A_t) | s_t \right] \tag{9}$$

the advantage at time step t is then given by:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \tag{10}$$

For a more detailed description of actor-critic algorithm in RL, see [14].

3.2.3 Proximal Policy Optimization

We use Proximal Policy Optimization (PPO, [47]), a state-of-the-art policy gradient technique, to train our network. PPO holds a leash on policy updates whose necessity is elaborated in trust region policy optimization (TRPO, [48]), yet much simplified compared to the latter. Furthermore, it enables multiple epochs of minibatch updates within one episode. The objective function is given as follow:

$$J^*(\theta) = \max_\theta J(\theta)$$

$$= \max_\theta \mathbb{E}_{\mu, \pi_\theta^{old}}\left[\sum_{t=1}^{T} \min\left\{ r_t(\theta) A^{\pi_\theta^{old}}(s_t, a_t), clip_\epsilon(r_t(\theta)) A^{\pi_\theta^{old}}(s_t, a_t) \right\} \right] \tag{11}$$

where $r_t(\theta) = \pi_\theta^{new}(a_t|s_t)/\pi_\theta^{old}(a_t|s_t)$ and $clip_\epsilon(x) = \min\{\max\{1-\epsilon, x\}, 1+\epsilon\}$. During policy iterations, π^{new} is updated each epoch and π^{old} is cloned from π^{new} each episode.

3.3 Exploration with Random Network Distillation

Previous sequential graph generating frameworks often suffer from mode collapse [37, 58, 61]. To establish a curiosity-driven policy, we seek to employ a simple and efficient exploration approach that can be naturally incorporated into our architecture. We perform Random Network Distillation (RND, [3]) on graphs or pre-trained feature graphs to fulfill this need. We initialize two random functions $\hat{f}_\psi, f^*$ with neural networks that map input graphs to feature vectors in $\mathbb{R}^{d_r}$. $\hat{f}_\psi$ is trained to match the output of f^*:

$$\psi^* = \arg\min_\psi \ \mathbb{E}_{s \sim \hat{p}_{next}} \|\hat{f}_\psi(s) - f^*(s)\| \tag{12}$$

where $\hat{p}_{next}$ is the empirical distribution of all the previously selected next states, i.e. the states that have been explored. Thus, when a new state s' is generated and given to the networks, $\|\hat{f}_\psi(s') - f^*(s')\|$ is designed to be small if s' has high similarity to previous states and large if it is outside the domain they have covered. We record running errors in a buffer and construct the surrogate innovation reward as:

$$r_i(s') = clip_\delta \left(\left(\|\hat{f}_\psi(s') - f^*(s')\| - m_b \right) \Big/ \sqrt{v_b} \right) \tag{13}$$

where m_b and v_b are the first and second central moment inferred from the running buffer, $clip_\delta(x) = \min\{\max\{-\delta, x\}, \delta\}$. Similar to imposing relative weight between the main reward and the innovation reward, δ is customized based on how ambitious the curiosity is intended to be.

3.4 Parallelization and Synchronized Evaluation

Interacting with the environment and obtaining rewards through external software programs are the two major performance bottlenecks in ours as well as RL problems in general. An advantage of our environment settings, as stated in Section 3.2.1, is that a constant trajectory length is feasible. Moreover, the costs for environmental interactions are about the same for different input states. To take advantage of this, we parallelize environments on subprocesses and execute batched operations on one GPU process, which enables synchronized and sparse reward evaluations that reduces the the number of calls to the docking program. For future experiments where such conditions might be unrealistic, we also provided options for asynchronous Parallel-GPU and Parallel-CPU samplers (described in [51]) in addition to the Parallel-GPU sampler used in our experiment.

4 Experiments

We evaluated our model against state-of-the-art generative models with the objective of discovering novel inhibitors targeting SARS-CoV-2 NSP15. The implementation can be found in our GitHub repository [†].

4.1 Setup

Dataset   The atomic coordinates of an apo form of the receptor were derived from the crystal structure of NSP15 (PDB ID 6W01) [24]. To train all our models on the task of optimizing docking scores, we used a dataset of SMILES IDs taken from a set of more than six million compounds from the MCULE molecular library - a publicly available dataset of purchasable molecules [27].

Docking Reward   Using AutoDockTools [36], the search space for docking on NSP15 was defined as a box with dimensions 30x30x26 Å, encompassing the amino acid side chains that form the catalytic and binding site of uridine in NSP15 endoribonuclease. Ligands that bind in this region with high affinity will likely impair protein function. See the figure in Appendix B for a visualization. Maximizing docking scores is a challenging task, and thus far deep generative models have had little success under this objective [6].
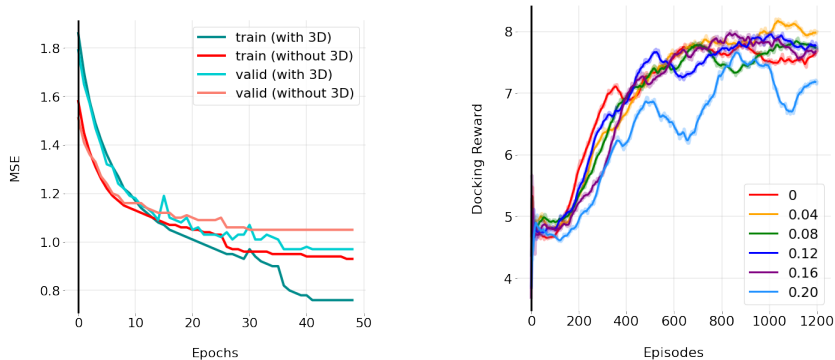

[†]`https://github.com/njchoma/DGAPN`

Figure 3: Demonstrations for two of our key features: spatial convolution on 3D graph and training with surrogate curiosity scores as innovation rewards. Left plot shows training and validation MSE in sGAT pre-training with and without 3D graph. Right plot shows DGAPN training with different relative weights on innovation rewards, with 0 being no innovation reward, i.e. GAPN.

Environment Each call to the environment with the current molecular state returns a variable-sized list of next possible states. In this case, next states consist of candidate molecules that are fragment-based modifications of the current molecule by the CReM package, which can be additive, subtractive, or both. For training efficiency, we limit the number of candidate molecules to 15-20, while during evaluation we expand this number to 128.

Starting distribution Our framework is provided with a 'warm start' by instantiating with a random sample of molecules from a set of existing 'expert' molecules from the dataset (though it can also start from a single atom). Together with pre-trained sGAT, it is expected to partially make up for the lack of imitation learning.

Model We experimented on several configurations and found the following network configurations to provide consistent results: 2 GNN layers with 256 hidden neurons, 4 MLP layers with 256 hidden neurons for actor and critic, 3 MLP layers with 128 hidden neurons for RND. In our case, although sparsifying the 3D graph G is not necessary since the average number of nodes is low enough for the gather and scatter operations (GS) used by Pytorch-Geometric to experience no noticeable difference in runtime [11], it was still carried out because we discovered that proper cutoffs improved the validation loss.

4.2   Results

On the task of generating ligands that present high estimated binding affinity for the SARS-CoV-2 Nsp15 target, DGAPN outperformed all state-of-the-art models in terms of the average score and diversity of generated molecules, and achieved the best synthetic accessibility score. As for highest scores being discovered, our model also outperformed all other models except for the one outlier GCPN generated. See Table 1 for a summary of the results. More descriptions of the models we compared to can be found in section 5. Detailed definitions of the metrics as well as snapshots and scatter plots of the generated molecules can be found in the Appendix.

In the table, GAPN w/ pre-train has 4 embedding layers shared from a pre-trained sGAT with supervised learning. DGAPN is GAPN with additional curiosity scores learned by RND with a relative weight of 0.1. The curiosity scores were incorporated as innovation rewards after a 100 episode delay and cut off after 2000 episodes. QED [2] is an indicator of drug-likeness, SA [10] is the synthetic accessibility of molecules, FCD [43] is the Fréchet ChemNet Distance that measures if generated molecules are similar to ground truth molecules by comparing neural network activations. QED is better when the score is higher, while SA and FCD are the opposite. Of these three metrics that aim to measure the realism of molecules, although ours did not achieved the best QED, another generative model with fragment-based action space, JTVAE, reached the highest score 0.70 among these methods. Furthermore, our models achieved the best SAs 2.89, 3.14 and 3.22, followed by JTVAE's 3.34, showing on an experimental level that fragment-based action space is beneficial for

realism to a certain degree. As for FCD, MolDQN attained the best result. Note that all models have achieved almost perfect chemical validity, while all RL algorithms achieved 100% validity. Thus, we did not include it as a metric in the table.

Table 1: Top docking scores and overall metrics in evaluations

| | Dock Score | | | | QED | SA | FCD | Diversity |
	mean	1st	2nd	3rd				
REINVENT	-5.6	-10.22	-9.76	-9.50	0.57	7.8	7e-3	0.11
JTVAE	-5.6	-8.56	-8.39	-8.39	**0.70**	3.34	2e-1	0.13
GCPN*	-4.8	**-16.53**	-10.72	-10.6	0.45	7.64	6e-5	0.04
MolDQN	-6.7	-10.88	-10.51	-10.36	0.16	6.47	**3e-5**	0.01
GAPN	-7.5	-10.80	-10.75	-10.44	0.39	3.14	4e-2	0.13
GAPN w/ pre-train	-7.1	-10.24	-10.19	-10.12	0.47	**2.89**	3e-2	0.13
DGAPN	**-8.3**	**-12.78**	**-12.12**	**-11.72**	0.36	3.22	9e-3	**0.15**

Regarding our primary objectives, which are optimizing docking scores and diversity, our final architecture achieved significantly better results than state-of-the-art models. With exploration, DGAPN is able to acquire a mean that is 24% higher than the best state-of-the-art model and discover the 2nd, 3rd and 4th best scores -12.78, -12.12 and -11.72. None of the other models are able to pass -11 with the exception of an outlier score from GCPN of -16.53. Interestingly enough, GCPN also has the worst average among all RL algorithms. In experiments, we did not find pre-training sGAT layers to be much helpful with the molecular docking task, although it did improved SA. As for spatial convolution and innovation rewards, we did find them to be particularly helpful in training, but the latter needs to be properly weighted (See Figure 3). As for diversity, it is clearly observed that our framework addressed the mode collapse problems of state-of-the-art RL algorithms to a great extent, achieving a score of 0.15. While the largest influence on diversity seems to come from our design of the starting distribution and action space, the distilled model with curiosity score has offered an additional positive impact on discovery.

5   Related Work

REINVENT   The REINVENT architecture consists of two recurrent neural network (RNN) architectures, generating molecules as tokenized SMILE strings. In this setting, the "Prior network" is trained with maximum likelihood estimation on a set of canonical SMILE strings, while the "Agent network" is trained with a policy gradient and rewarded using a combination of task scores and Prior network estimations [37].

JTVAE   The Junction Tree Variational Autoencoder (JT-VAE) trains two encoder / decoder networks in building a fixed-dimension latent space representation of molecules, where one network captures junction tree structure of each molecule, while the other is responsible for fine grain connectivity. Novel molecules with desired properties are then generated using Bayesian optimization on the latent space, while the decoder networks stochastically reconstruct the associated molecules [21].

GCPN   Graph Convolutional Policy Network (GCPN) is a policy gradient RL architecture for de novo molecular generation. The network defines domain-specific modifications on molecular graphs so that chemical validity is maintained at each episode. Additionally, the model optimizes for realism with adversarial training and expert-pretraining using trajectories generated from known molecules in the ZINC library [58].

MOLDQN   Molecule Deep Q-Networks (MolDQN) is a Q-learning model using Morgan fingerprint as representations of molecules. To achieve molecular validity, chemical modifications e.g. atom additions, bond additions, and bond removals are directly defined for each episode. To enhance exploration of chemical space, MolDQN learns H independent Q-functions, each of which is trained

---

*The GCPN we are using is our re-implementation of the original framework in PyTorch, excluding the adversarial training.

on separate sub-samples of the training data, and samples uniformly from these Q-functions during each episode [61].

6 Conclusions

In this work, we introduced a spatial graph attention mechanism and a curiosity-driven policy network to discover novel molecules optimized for targeted objectives. We identified candidate antiviral compounds designed to inhibit the SARS-CoV-2 protein NSP15, leveraging extensive molecular docking simulations. Our framework advances the state of the art in the optimization of molecules with antiviral potential, as measured by molecular docking scores. We note that a valuable extension of our work would be to focus on lead-optimization — the refinement of molecules already known to bind the protein of interest through position-constrained modification. Such knowledge-based and iterative refinements may help to work around limitations of the accuracy of molecular docking predictions.

The pipeline we developed was optimized for molecular design in the context of molecules targeted to proteins. This task is of widespread interest in drug discovery, and, more broadly, synthetic biology. Future work will focus on generalizing the framework we present for other design tasks where targets are amenable to graph-based representations, particularly when training data can be generated through simulation or high-throughput experimentation. Using such models for AI-guided experimental design has potential to automate the process of scientific discovery — permitting the fitted model to optimize experimentation.

References

[1] Song Bai, Feihu Zhang, and Philip HS Torr. "Hypergraph convolution and hypergraph attention". In: *Pattern Recognition* 110 (2021), p. 107637.

[2] G Richard Bickerton et al. "Quantifying the chemical beauty of drugs". In: *Nature chemistry* 4.2 (2012), pp. 90–98.

[3] Yuri Burda et al. "Exploration by random network distillation". In: *arXiv preprint arXiv:1810.12894* (2018).

[4] Dexiong Chen, Laurent Jacob, and Julien Mairal. "Convolutional kernel networks for graph-structured data". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1576–1586.

[5] Jun Chen and Haopeng Chen. "Edge-Featured Graph Attention Network". In: *arXiv preprint arXiv:2101.07671* (2021).

[6] Tobiasz Cieplinski et al. *We Should at Least Be Able to Design Molecules That Dock Well*. 2020. arXiv: 2006.16955 [q-bio.BM].

[7] Hanjun Dai et al. "Syntax-Directed Variational Autoencoder for Structured Data". In: (Feb. 2018). arXiv: 1802.08786 [cs.LG].

[8] Tomasz Danel et al. "Spatial graph convolutional networks". In: *International Conference on Neural Information Processing*. Springer. 2020, pp. 668–675.

[9] Nicola De Cao and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs". In: *arXiv preprint arXiv:1805.11973* (2018).

[10] Peter Ertl and Ansgar Schuffenhauer. "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions". In: *Journal of cheminformatics* 1.1 (2009), pp. 1–11.

[11] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[12] Anna Gaulton et al. "ChEMBL: a large-scale bioactivity database for drug discovery". en. In: *Nucleic Acids Res.* 40.Database issue (Jan. 2012), pp. D1100–7.

[13] Rafael Gómez-Bombarelli et al. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". en. In: *ACS Cent Sci* 4.2 (Feb. 2018), pp. 268–276.

[14] Ivo Grondman et al. "A survey of actor-critic reinforcement learning: Standard and natural policy gradients". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 1291–1307.

[15] Gabriel Lima Guimaraes et al. "Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models". In: (May 2017). arXiv: 1705.10843 [stat.ML].

[16] William L Hamilton, Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *arXiv preprint arXiv:1706.02216* (2017).

[17] Sheng You Huang, Sam Z Grinter, and Xiaoqin Zou. "Scoring functions and their evaluation methods for protein-ligand docking: Recent advances and future directions". In: *Physical Chemistry Chemical Physics* 12.40 (2010), pp. 12899–12908.

[18] Ruth Huey et al. "A semiempirical free energy force field with charge-based desolvation". In: *Journal of Computational Chemistry* 28.6 (2007), pp. 1145–1152.

[19] James P Hughes et al. "Principles of early drug discovery". In: *British journal of pharmacology* 162.6 (2011), pp. 1239–1249.

[20] Woosung Jeon and Dongsup Kim. "Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors". In: *Scientific reports* 10.1 (2020), pp. 1–11.

[21] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. "Junction tree variational autoencoder for molecular graph generation". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 2323–2332.

[22] Hiroshi Kajino. "Molecular hypergraph grammar with its application to molecular optimization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3183–3191.

[23] Seokho Kang and Kyunghyun Cho. "Conditional Molecular Design with Deep Generative Models". en. In: *J. Chem. Inf. Model.* 59.1 (Jan. 2019), pp. 43–52.

[24] Youngchang Kim et al. "Crystal structure of Nsp15 endoribonuclease NendoU from SARS-CoV-2". In: *Protein Science* 29.7 (2020), pp. 1596–1605.

[25] Youngchang Kim et al. "Tipiracil binds to uridine site and inhibits Nsp15 endoribonuclease NendoU from SARS-CoV-2". In: *Communications Biology* 4.1 (2021), pp. 1–11.

[26] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[27] Robert Kiss, Mark Sandor, and Ferenc A Szalai. "http://Mcule.com: a public web service for drug discovery". en. In: *J. Cheminform.* 4.Suppl 1 (2012), P17.

[28] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. "Grammar Variational Autoencoder". In: (Mar. 2017). arXiv: 1703.01925 [stat.ML].

[29] Scott Legrand et al. "GPU-Accelerated Drug Discovery with Docking on the Summit Supercomputer: Porting, Optimization, and Application to COVID-19 Research". In: (Aug. 2020). arXiv: 2007.03678.

[30] Yujia Li et al. "Learning deep generative models of graphs". In: *arXiv preprint arXiv:1803.03324* (2018).

[31] Bowen Liu et al. "Retrosynthetic reaction prediction using neural sequence-to-sequence models". In: *ACS central science* 3.10 (2017), pp. 1103–1113.

[32] Qi Liu et al. "Constrained graph variational autoencoders for molecule design". In: *arXiv preprint arXiv:1805.09076* (2018).

[33] Tengfei Ma, Jie Chen, and Cao Xiao. "Constrained generation of semantically valid graphs via regularizing variational autoencoders". In: *arXiv preprint arXiv:1809.02630* (2018).

[34] Christopher Morris et al. "Weisfeiler and leman go neural: Higher-order graph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 4602–4609.

[35] Garret M Morris et al. "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility". In: *Journal of Computational Chemistry* 30.16 (2009), pp. 2785–2791.

[36] Garrett M Morris et al. "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility". en. In: *J. Comput. Chem.* 30.16 (Dec. 2009), pp. 2785–2791.

[37] Marcus Olivecrona et al. "Molecular de-novo design through deep reinforcement learning". en. In: *J. Cheminform.* 9.1 (Sept. 2017), p. 48.

[38] Hongbin Pei et al. "Geom-gcn: Geometric graph convolutional networks". In: *arXiv preprint arXiv:2002.05287* (2020).

[39]   Monica C Pillon et al. "Cryo-EM structures of the SARS-CoV-2 endoribonuclease Nsp15 reveal insight into nuclease specificity and dynamics". In: *Nature Communications* 12.1 (2021), pp. 1–12.

[40]   Pavel Polishchuk. "CReM: chemically reasonable mutations framework for structure generation". In: *Journal of Cheminformatics* 12 (2020), pp. 1–18.

[41]   Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. "Estimation of the size of drug-like chemical space based on GDB-17 data". In: *Journal of computer-aided molecular design* 27.8 (2013), pp. 675–679.

[42]   Mariya Popova, Olexandr Isayev, and Alexander Tropsha. "Deep reinforcement learning for de novo drug design". en. In: *Sci Adv* 4.7 (July 2018), eaap7885.

[43]   Kristina Preuer et al. "Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery". In: *Journal of chemical information and modeling* 58.9 (2018), pp. 1736–1741.

[44]   Bidisha Samanta et al. "Nevae: A deep generative model for molecular graphs". In: *Journal of machine learning research. 2020 Apr; 21 (114): 1-33* (2020).

[45]   Diogo Santos-Martins et al. "Accelerating AutoDock4 with GPUs and gradient-based local search". In: *Journal of Chemical Theory and Computation* 17.2 (2021), pp. 1060–1073.

[46]   Petra Schneider et al. "Rethinking drug design in the artificial intelligence era". In: *Nature Reviews Drug Discovery* 19.5 (2020), pp. 353–364.

[47]   John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[48]   John Schulman et al. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.

[49]   Marwin H S Segler et al. "Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks". en. In: *ACS Cent Sci* 4.1 (Jan. 2018), pp. 120–131.

[50]   Martin Simonovsky and Nikos Komodakis. "Graphvae: Towards generation of small graphs using variational autoencoders". In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 412–422.

[51]   Adam Stooke and Pieter Abbeel. "rlpyt: A research code base for deep reinforcement learning in pytorch". In: *arXiv preprint arXiv:1909.01500* (2019).

[52]   Ashish Vaswani et al. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[53]   Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[54]   Xiao Wang et al. "Heterogeneous graph attention network". In: *The World Wide Web Conference*. 2019, pp. 2022–2032.

[55]   David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[56]   Xiufeng Yang et al. "ChemTS: an efficient python library for de novo molecular generation". en. In: *Sci. Technol. Adv. Mater.* 18.1 (Nov. 2017), pp. 972–976.

[57]   Ying Yang et al. "Efficient Exploration of Chemical Space with Docking and Deep-Learning". In: (2021).

[58]   Jiaxuan You et al. "Graph convolutional policy network for goal-directed molecular graph generation". In: *arXiv preprint arXiv:1806.02473* (2018).

[59]   Jiaxuan You et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5708–5717.

[60]   Matthew D Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: abs/1212.5701 (2012). arXiv: 1212.5701.

[61]   Zhenpeng Zhou et al. "Optimization of molecules via deep reinforcement learning". In: *Scientific reports* 9.1 (2019), pp. 1–10.

A Appendix

A. Measure Theory Construction of Markov Decision Process

Let $(S, \mathcal{S})$ and $(A, \mathcal{A})$ be two measurable spaces called the state space and action space; functions $\Pi : S \times \mathcal{A} \to \mathbb{R}$ and $T : S \times A \times \mathcal{S} \to \mathbb{R}$ are said to be a policy and a transition probability respectively if

1. For each $s \in S$, $E \to \Pi(s, E)$ is a probability measure on $(A, \mathcal{A})$; for each $(s, a) \in S \times A$, $F \to T(s, a, F)$ is a probability measure on $(S, \mathcal{S})$.

2. For each $E \in \mathcal{A}$, $s \to \Pi(s, E)$ is a measurable function from $(S, \mathcal{S}) \to (\mathbb{R}, \mathcal{B})$; for each $F \in \mathcal{S}$, $(s, a) \to T(s, a, F)$ is a measurable function from $(S \times A, \mathcal{S} \otimes \mathcal{A}) \to (\mathbb{R}, \mathcal{B})$.

We say a sequence of random variable duples (S_t, A_t) defined on the two measurable spaces is a Markov decision chain if

$$P(A_t \in E \mid \sigma(S_0, A_0, S_1, A_1, \ldots, S_t)) = \Pi(S_t, E) \tag{14}$$

$$P(S_{t+1} \in F \mid \sigma(S_0, A_0, S_1, A_1, \ldots, S_t, A_t)) = T(S_t, A_t, F) \tag{15}$$

A function $r : S \times \mathcal{A} \to \mathbb{R}$ is said to be the reward function w.r.t. the Markov decision chain if $r(s_t, E_t) = \mathbb{E}_{\Pi, T} [R(s_{t+1}) \mid S_t = s_t, A_t \in E_t]$ where $R : S \to \mathbb{R}$ is its underlying reward function.

With an abuse of notation, we define $\pi(a|s) := \Pi(s, \{a\})$, $\rho(s'|s, a) := T(s, a, \{s'\})$ and let $r(s, a)$ denote $r(s, \{a\})$.

B. Reinforcement Learning Environment and Reward Evaluation Details

B. 1. Environment - CReM

Chemically Reasonable Mutations (CReM) is an open-source fragment-based framework for chemical structure modification. Compared to atom-by-atom structure modification frameworks, CReM explores less of chemical space but guarantees chemical validity for each modification, because only fragments that are in the same chemical context are interchangeable. Compared to reaction-based frameworks, CReM enables a larger exploration of chemical space but may explore chemical modifications that are less synthetically feasible. Fragments are generated from the ChEMBL database [12] and for each fragment, the chemical context is encoded for several context radius sizes in a SMILES string and stored along with the fragment in a separate database. For each query molecule, mutations are enumerated by matching the context of its fragments with those that are found in the CReM fragment-context database [40]. In this work, we use the mutate function to enumerate possible modifications and use the default context radius size of 3 to find replacements.

B. 2. Evaluation - AutoDock-GPU

Docking programs use the three-dimensional structure of the protein (i.e., the receptor) to predict the most stable bound conformations of the small molecules (i.e., its putative ligands) of interest, often targeting a pre-defined functional site, such as the catalytic site. An optimization algorithm within a scoring function is employed towards finding the ligand conformations that likely correspond to binding free energy minima. The scoring function is conformation-dependent and typically comprises physics-based empirical or semi-empirical potentials that describe pair-wise atomic terms, such as dispersion, hydrogen bonding, electrostatics, and desolvation [17, 18]. AutoDock is a computational simulated docking program that uses a Lamarckian genetic algorithm to predict native-like conformations of protein-ligand complexes and a semi-empirical scoring function to estimate the corresponding binding affinities. Lower values of docking scores (i.e., more negative) indicate stronger predicted interactions [45]. The opposite value of the lowest estimated binding affinity energy obtained for each molecule forms the reward.

AutoDock-GPU is an extension of AutoDock to leverage the highly-parallel architecture of GPUs and was implemented in our framework. Within AutoDock-GPU, ADADELTA, a gradient-based method, is used for local refinement [60]. The structural information of the receptor (here, the NSP15 protein) used by AutoDock-GPU is processed prior to running the framework. In this preparatory step, AutoDockTools [36] was used to define the search space for docking on NSP15 (PDB ID 6W01;
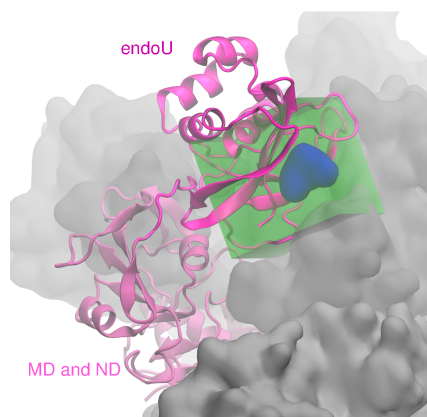
13

Figure 4: The search space in NSP15 defined for molecular docking (green box). An NSP15 protomer, which was used as the receptor in the calculations, is shown (cartoon backbone representation, in pink/magenta). The nucleotide density located at the catalytic site is depicted (blue surface). Other protomers forming the homo-hexamer are shown as grey surfaces. PDB IDs 6WLC and 6WXC were used in this illustration [25]. Abbreviations: EndoU, Poly-U specific endonuclease domain; MD, Middle domain; ND, N-terminal domain.

Figure 4) and to generate the PDBQT file of the receptor, which contains the atomic coordinates, partial charges, and AutoDock atom types. AutoGrid4 [35] was used to pre-calculate grid maps of interaction energy at the binding site for the different atom types defined in CReM.

C. QED, SA, and FCD

C. 1. Quantitative Estimate of Druglikeness

(QED) is defined as

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

where d_i are eight widely used molecular properties. Specifically, they are molecular weight (MW), octanol-water partition coefficient (ALOGP), number of hydrogen bond donors (HBD), number of hydrogen bond acceptors (HBA), molecular polar surface area (PSA), number of rotatable bonds (ROTB), the number of aromatic rings (AROM), and number of structural alerts. For each d_i,

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

each $a_i, \ldots, f_i$ are given by a supplementary table in [2].

C. 2. Synthetic Accessibility

(SA) is defined as

$$SA = \texttt{fragmentScore} - \texttt{complexityPenalty}$$

The fragment score is calculated as a sum of contributions from fragments of 934,046 PubChem already-synthesized chemicals. The complexity penalty is computed from a combination of ring-ComplexityScore, stereoComplexityScore, macroCyclePenalty, and the sizePenalty, which are as follows:

$$\texttt{ringComplexityScore} = \log(\texttt{nRingBridgeAtoms} + 1) + \log(\texttt{nSprioAtoms} + 1)$$
$$\texttt{stereoComplexityScore} = \log(\texttt{nStereoCenters} + 1)$$
$$\texttt{macroCyclePenalty} = \log(\texttt{nMacroCycles} + 1)$$
$$\texttt{sizePenalty} = \texttt{nAtoms}^{1.005} - \texttt{nAtoms}$$

C. 3. Fréchet ChemNet Distance

(FCD) is defined as the distance between a Gaussian distribution $p_w(.)$ of real-world molecules with mean and covariance $(\boldsymbol{m}_w, \boldsymbol{C}_w)$, and a Gaussian distribution $p_w(.)$ of molecules from a generative model with mean and covariance $(\boldsymbol{m}, \boldsymbol{C})$. FCD is given by

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\,(\boldsymbol{C} + \boldsymbol{C}_w - 2(\boldsymbol{C}\boldsymbol{C}_w)^{1/2}).$$

D. Snapshots and Scatter plots of generated molecules

See Figure 5 and Figure 6.

(a) Reinvent: -10.2, -9.8, -9.5

(b) JTVAE: -8.6, -8.4, -8.4

(c) GCPN: -16.5, -10.7, -10.6

(d) MolDQN: -10.9, -10.5, -10.3

(e) CReM Greedy: -11.6, -10.6, -10.5

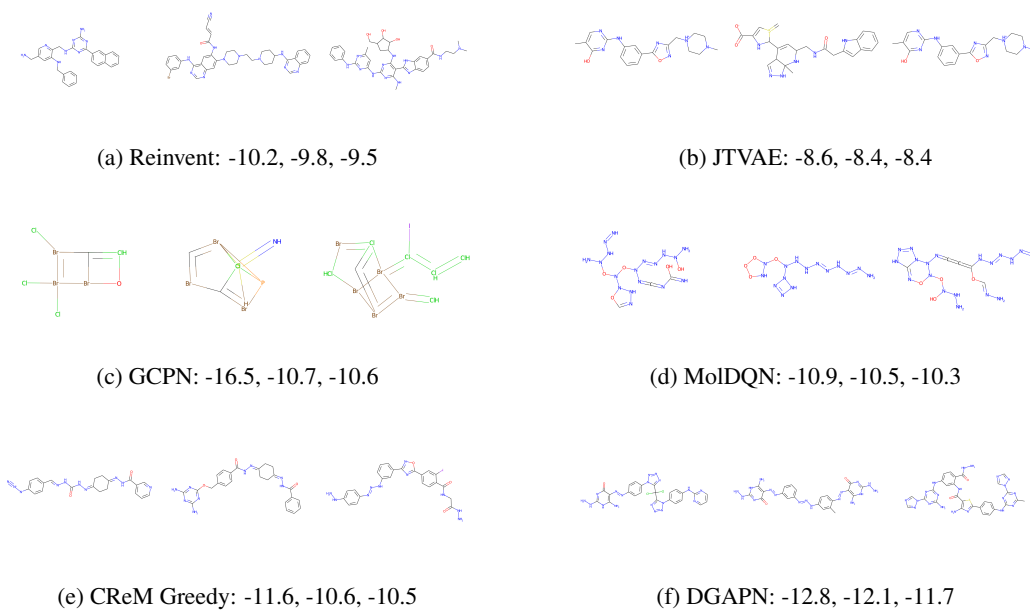(f) DGAPN: -12.8, -12.1, -11.7

Figure 5: Top 3 molecules found by each algorithm and greedy strategy with CReM environment.
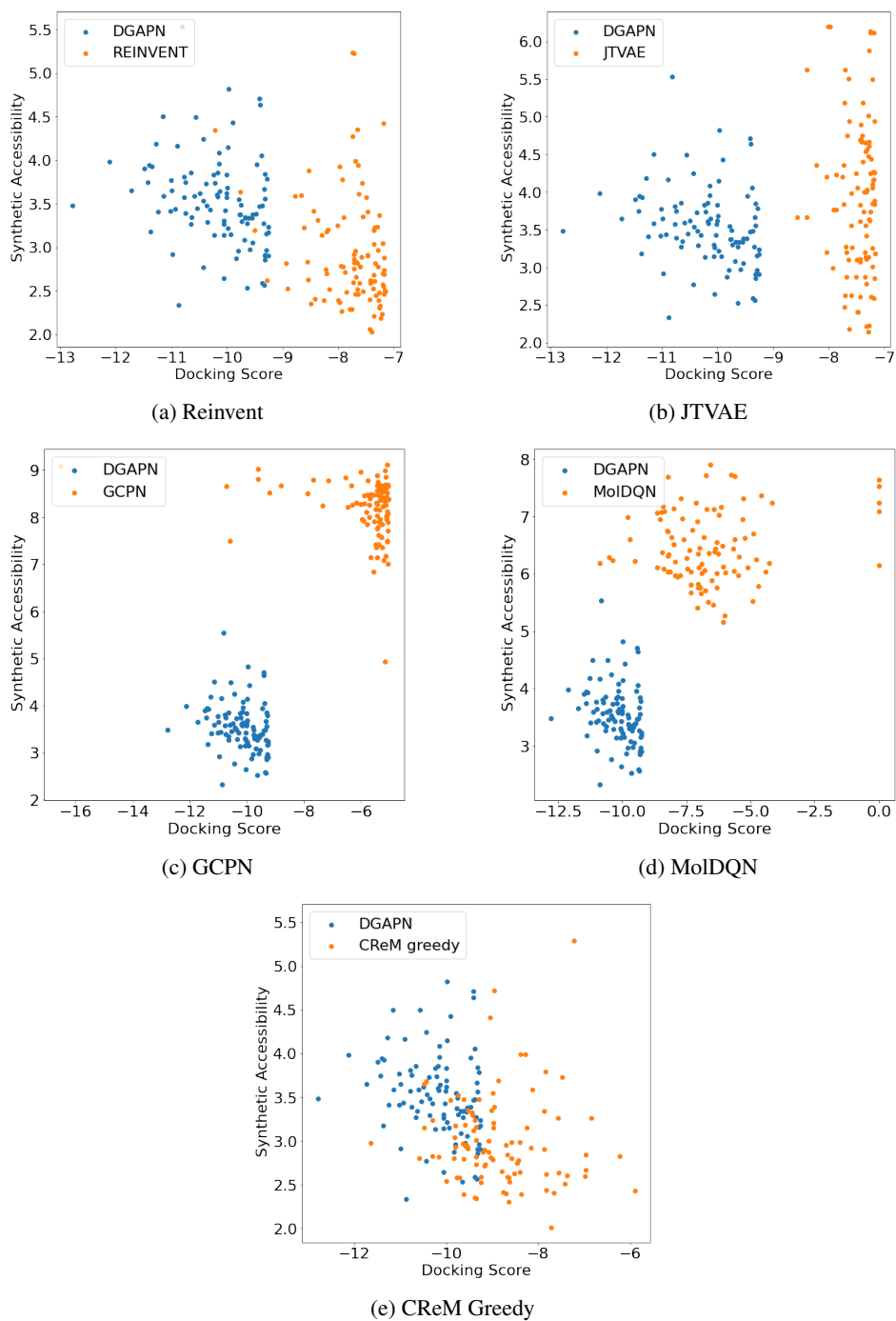
Figure 6: Comparison of top 100 molecules found by each algorithm and greedy strategy with CReM environment to DGAPN in terms of docking score (lower the better) and synthetic accessibility (lower the better).