Multi-Objective Molecule Generation using Interpretable Substructures

Wengong Jin [1]   **Regina Barzilay** [1]   **Tommi Jaakkola** [1]

Abstract

Drug discovery aims to find novel compounds with specified chemical property profiles. In terms of generative modeling, the goal is to learn to sample molecules in the intersection of multiple property constraints. This task becomes increasingly challenging when there are many property constraints. We propose to offset this complexity by composing molecules from a vocabulary of substructures that we call molecular rationales. These rationales are identified from molecules as substructures that are likely responsible for each property of interest. We then learn to expand rationales into a full molecule using graph generative models. Our final generative model composes molecules as mixtures of multiple rationale completions, and this mixture is fine-tuned to preserve the properties of interest. We evaluate our model on various drug design tasks and demonstrate significant improvements over state-of-the-art baselines in terms of accuracy, diversity, and novelty of generated compounds.

1. Introduction

The key challenge in drug discovery is to find molecules that satisfy multiple constraints, from potency, safety, to desired metabolic profiles. Optimizing these constraints simultaneously is challenging for existing computational models. The primary difficulty lies in the lack of training instances of molecules that conform to all the constraints. For example, for this reason, Jin et al. (2019a) reports over 60% performance loss when moving beyond the single-constraint setting.

In this paper, we propose a novel approach to multi-property molecular optimization. Our strategy is inspired by fragment-based drug discovery (Murray & Rees, 2009) often followed by medicinal chemists. The idea is to start

with substructures (e.g., functional groups or later pieces) that drive specific properties of interest, and then combine these building blocks into a target molecule. To automate this process, our model has to learn two complementary tasks: (1) identification of the building blocks that we call rationales, and (2) assembling multiple rationales together into a fully formed target molecule. In contrast to competing methods, our generative model does not build molecules from scratch, but instead assembles them from automatically extracted rationales already implicated for specific properties (see Figure 1).

We implement this idea using a generative model of molecules where the rationale choices play the role of latent variables. Specifically, a molecular graph $\mathcal{G}$ is generated from underlying rationale sets $\mathcal{S}$ according to:

$$P(\mathcal{G}) = \sum_{\mathcal{S}} P(\mathcal{G}|\mathcal{S})P(\mathcal{S}) \tag{1}$$

As ground truth rationales (e.g., functional groups or subgraphs) are not provided, the model has to extract candidate rationales from molecules with the help of a property predictor. We formulate this task as a discrete optimization problem efficiently solved by Monte Carlo tree search. Our rationale conditioned graph generator, $P(\mathcal{G}|\mathcal{S})$, is initially trained on a large collection of real molecules so that it is capable of expanding any subgraph into a full molecule. The mixture model is then fine-tuned using reinforcement learning to ensure that the generated molecules preserve all the properties of interest. This training paradigm enables us to realize molecules that satisfy multiple constraints without observing any such instances in the training set.

The proposed model is evaluated on molecule design tasks under different combinations of property constraints.[1] Our baselines include state-of-the-art molecule generation methods (Olivecrona et al., 2017; You et al., 2018a). Across all tasks, our model achieve state-of-the art results in terms of accuracy, novelty and diversity of generated compounds. In particular, we outperform the best baseline with 38% absolute improvement in the task with three property constraints. We further provide ablation studies to validate the benefit of our architecture in the low-resource scenario. Finally, we show that identified rationales are chemically meaningful in a toxicity prediction task (Sushko et al., 2012).

[1]MIT CSAIL. Correspondence to: Wengong Jin <wengong@casil.mit.edu>.
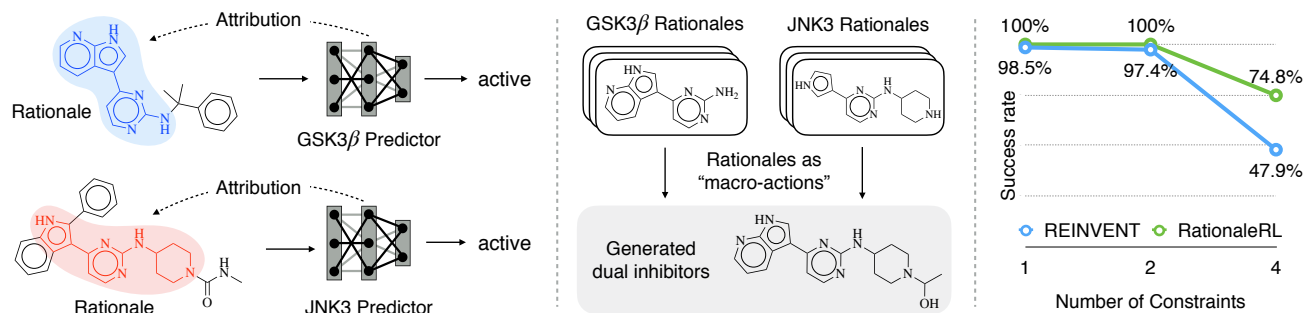
---

[1]https://github.com/wengong-jin/multiobj-rationale

Figure 1. Illustration of RationaleRL. *Left*: To generate a dual inhibitor against biological targets GSK3β and JNK3, our model first identifies rationale substructures $\mathcal{S}$ for each property. Note that rationales are not provided as domain knowledge. *Middle*: The model learns to compose multiple rationales $\mathcal{S}$ into a complete molecule $\mathcal{G}$. *Right:* Our method achieves much higher success rate than the current state-of-the-art molecule design method REINVENT (Olivecrona et al., 2017)) under four property constraints.

2. Related Work

Reinforcement Learning One of the prevailing paradigms for drug design is reinforcement learning (RL) (You et al., 2018a; Olivecrona et al., 2017; Popova et al., 2018), which seeks to maximize the expected reward defined as the sum of predicted property scores using the property predictors. Their approach learns a distribution $P(\mathcal{G})$ (a neural network) for generating molecules. Ideally, the model should achieve high success rate in generating molecules that meet all the constraints, while maintaining the diversity of $P(\mathcal{G})$.

The main challenge of RL is the reward sparsity, especially when there are multiple competing constraints. For illustration, we test a state-of-the-art reinforcement learning method (Olivecrona et al., 2017) under four property constraints: biological activity to GSK3β, JNK3, drug-likeness and synthetic accessibility (Li et al., 2018). As shown in Figure 1, initially the success rate and diversity is high when given only one of the constraints, but they decrease dramatically when all the property constraints are added. The reason of this failure is that the property predictor (i.e., reward function) remains black-box and the model has limited understanding of why certain molecules are desirable.

Our framework offsets this complexity by understanding property landscape through rationales. At a high level, the rationales are analogous to *options* (Sutton et al., 1999; Stolle & Precup, 2002), which are macro-actions leading the agent faster to its goal. The rationales are automatically discovered from molecules with labeled properties.

Molecule Generation Previous work have adopted various approaches for generating molecules under specific property constraints. Roughly speaking, existing methods can be divided along two axes — representation and optimization. On the representation side, they either operate on SMILES strings (Gómez-Bombarelli et al., 2018; Segler et al., 2017; Kang & Cho, 2018) or directly on molecular graphs (Simonovsky & Komodakis, 2018; Jin et al., 2018;

Samanta et al., 2018; Liu et al., 2018; De Cao & Kipf, 2018; Ma et al., 2018; Seff et al., 2019). On the optimization side, the task has been formulated as reinforcement learning (Guimaraes et al., 2017; Olivecrona et al., 2017; Popova et al., 2018; You et al., 2018a; Zhou et al., 2018), continuous optimization in the latent space learned by variational autoencoders (Gómez-Bombarelli et al., 2018; Kusner et al., 2017; Dai et al., 2018; Jin et al., 2018; Kajino, 2018; Liu et al., 2018), or graph-to-graph translation (Jin et al., 2019b). In contrast to existing approaches, our model focuses on the multi-objective setting of the problem and offers a different formulation for molecule generation based on rationales.

Interpretability Our rationale based generative model seeks to provide transparency (Doshi-Velez & Kim, 2017) for molecular design. The choice of rationales $P(\mathcal{S})$ is visible to users and can be easily controlled by human experts. Prior work on interpretability primarily focuses on finding rationales (i.e., explanations) of model predictions in image and text classification (Lei et al., 2016; Ribeiro et al., 2016; Sundararajan et al., 2017) and molecule property prediction (McCloskey et al., 2019; Ying et al., 2019; Lee et al., 2019). In contrast, our model uses rationales as building blocks for molecule generation.

3. Proposed Approach: RationaleRL

Molecules are represented as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with atoms $\mathcal{V}$ as nodes and bonds $\mathcal{E}$ as edges. The goal of drug discovery is to find novel compounds satisfying given property constraints (e.g., drug-likeness, binding affinity, etc.). Without loss of generality, we assume the property constraints to be of the following form:

$$\text{Find molecules} \quad \mathcal{G}$$
$$\text{Subject to} \quad r_i(\mathcal{G}) \geq \delta_i; \quad i = 1, \cdots, M \qquad (2)$$

For each property i, the property score $r_i(\mathcal{G}) \in [0, 1]$ of molecule $\mathcal{G}$ must be higher than threshold $\delta_i \in [0, 1]$. A
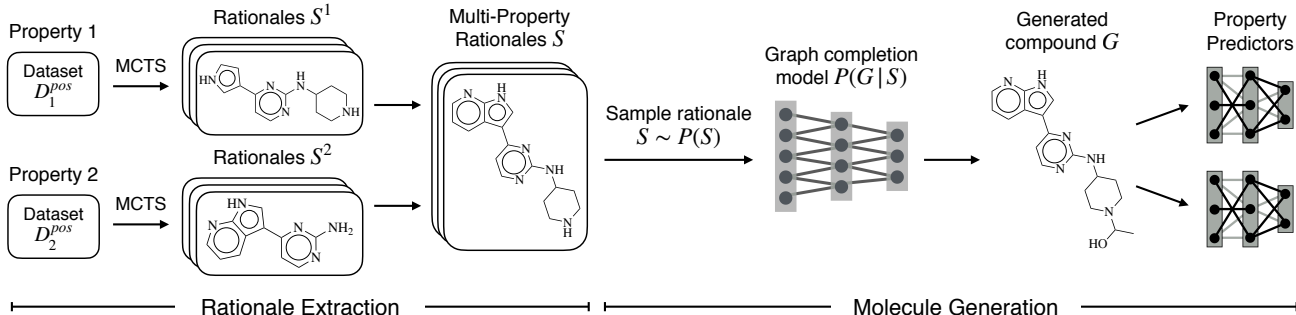
Figure 2. Overview of our approach. We first construct rationales for each individual property and then combine them as multi-property rationales. The method learns a graph completion model $P(\mathcal{G}|\mathcal{S})$ and rationale distribution $P(\mathcal{S})$ in order to generate positive molecules.

molecule $\mathcal{G}$ is called *positive* to property i if $r_i(\mathcal{G}) \geq \delta_i$ and *negative* otherwise.

Following previous work (Olivecrona et al., 2017; Popova et al., 2018), $r_i(\mathcal{G})$ is output of property prediction models (e.g., random forests) which effectively approximate empirical measurements. The prediction model is trained over a set of molecules with labeled properties gathered from real experimental data. The property predictor is then *fixed* throughout the rest of the training process.

Overview Our model generates molecules by first sampling a rationale $\mathcal{S}$ from the vocabulary $V_{\mathcal{S}}^{[M]}$, and then completing it into a molecule $\mathcal{G}$. The generative model is defined as

$$P(\mathcal{G}) = \sum_{\mathcal{S} \in V_{\mathcal{S}}^{[M]}} P(\mathcal{S})P(\mathcal{G}|\mathcal{S}) \tag{3}$$

As shown in Figure 2, our model consists of three modules:

- **Rationale Extraction**: Construct rationale vocabulary $V_{\mathcal{S}}^i$ each individual property i and combines these rationales for multiple properties $V_{\mathcal{S}}^{[M]}$ (see §3.1).

- **Graph Completion** $P(\mathcal{G}|\mathcal{S})$: Generate molecules $\mathcal{G}$ using multi-property rationales $S^{[M]} \in V_{\mathcal{S}}^{[M]}$. The model is first pre-trained on natural compounds and then fine-tuned to generate molecules satisfying multiple constraints (see §3.2 for its architecture and §3.3 for fine-tuning).

- **Rationale Distribution** $P(\mathcal{S})$: The rationale distribution $P(\mathcal{S})$ is learned based on the properties of complete molecules $\mathcal{G}$ generated from $P(\mathcal{G}|\mathcal{S})$. A rationale $\mathcal{S}$ is sampled more frequently if it is more likely to be expanded into a positive molecule $\mathcal{G}$ (see §3.3).

3.1. Rationale Extraction from Predictive Models

Single-property Rationale We define a rationale $\mathcal{S}^i$ for a single property i as a subgraph of some molecule $\mathcal{G}$ which causes $\mathcal{G}$ to be active (see Figure 1). To be specific, let $V_{\mathcal{S}}^i$ be the vocabulary of such rationales for property i. Each rationale $\mathcal{S}^i \in V_{\mathcal{S}}^i$ should satisfy the following two criteria to be considered as a rationale:

1. The size of $\mathcal{S}^i$ should be small (less than 20 atoms).

2. Its predicted property score $r_i(\mathcal{S}^i) \geq \delta_i$.

For a single property i, we propose to extract its rationales from a set of *positive* molecules $\mathcal{D}_i^{pos}$ used to train the property predictor. For each molecule $\mathcal{G}_i^{pos} \in \mathcal{D}_i^{pos}$, we find a rationale subgraph with high predicted property and small size ($N_s = 20$):

$$\begin{aligned}\text{Find subgraph} \quad & \mathcal{S}^i \subset \mathcal{G}_i^{pos} \\ \text{Subject to} \quad & r_i(\mathcal{S}^i) \geq \delta_i, \\ & |\mathcal{S}^i| \leq N_s \text{ and } \mathcal{S}^i \text{ is connected}\end{aligned} \tag{4}$$

Solving the above problem is challenging because rationale $\mathcal{S}^i$ is discrete and the potential number of subgraphs grows exponentially to the size of $\mathcal{G}_i^{pos}$. To limit the search space, we have added an additional constraint that $\mathcal{S}^i$ has to be a connected subgraph.[2] In this case, we can find a rationale $\mathcal{S}^i$ by iteratively removing some peripheral bonds while maintaining its property. Therefore, the key is learning to prune the molecule.

This search problem can be efficiently solved by Monte Carlo Tree Search (MCTS) (Silver et al., 2017). The root of the search tree is $\mathcal{G}^{pos}$ and each state s in the search tree is a subgraph derived from a sequence of bond deletions. To ensure that each subgraph is chemically valid and stays connected, we only allow deletion of one peripheral non-aromatic bond or one peripheral ring from each state. As shown in Figure 3, a bond or a ring a is called peripheral if $\mathcal{G}^{pos}$ stays connected after deleting a.

During search process, each state s in the search tree contains edges (s, a) for all legal deletions a. Following Silver et al. (2017), each edge (s, a) stores the following statistics:

- $N(s, a)$ is the visit count of deletion a, which is used for exploration-exploitation tradeoff in the search process.

---

[2]This assumption is valid in many cases. For instance, rationales for toxicity (i.e., toxicophores) are connected subgraphs in most cases (Sushko et al., 2012).
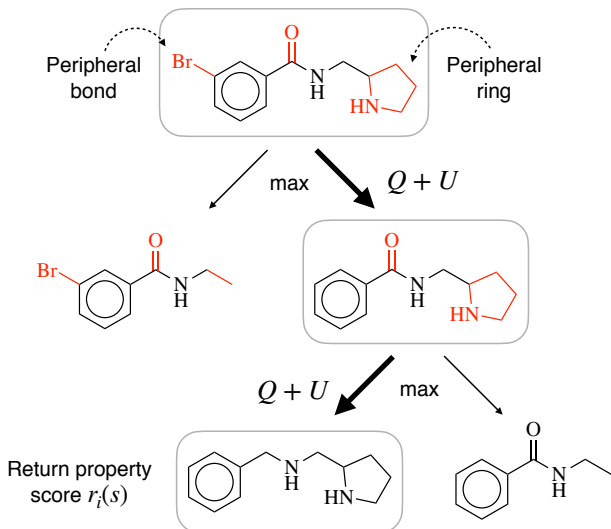
Figure 3. Illustration of Monte Carlo tree search for molecules. Peripheral bonds and rings are highlighted in red. In the forward pass, the model deletes a peripheral bond or ring from each state which has maximum $Q + U$ value (see Eq.(6)). In the backward pass, the model updates the statistics of each state.

- $W(s, a)$ is total action value which indicates how likely the deletion a will lead to a good rationale.

- $Q(s, a)$ is the mean action value: $W(s, a) / N(s, a)$

- $R(s, a) = r_i(s')$ is the predicted property score of the new subgraph s' derived from deleting a from s.

Guided by these statistics, MCTS searches for rationales in multiple iterations. Each iteration consists of two phases:

1. *Forward pass*: Select a path $s_0, \cdots, s_L$ from the root s_0 to a leaf state s_L with less than N atoms and evaluate its property score $r_i(s_L)$. At each state s_k, an deletion a_k is selected according to the statistics in the search tree:

$$a_k = \arg\max_a Q(s_k, a) + U(s_k, a) \quad (5)$$

$$U(s_k, a) = c_{\text{puct}} R(s_k, a) \frac{\sqrt{\sum_b N(s_k, b)}}{1 + N(s_k, a)} \quad (6)$$

where c_{puct} determines the level of exploration. This search strategy is a variant of the PUCT algorithm (Rosin, 2011). It initially prefers to explore deletions with high $R(s, a)$ and low visit count, but asympotically prefers deletions that are likely to lead to good rationales.

2. *Backward pass*: The edge statistics are updated for each state s_k. Specifically, $N(s_k, a_k) \leftarrow N(s_k, a_k) + 1$ and $W(s_k, a_k) \leftarrow W(s_k, a_k) + r_i(s_L)$.

In the end, we collect all the leaf states s with $r_i(s) \geq \delta_i$ and add them to the rationale vocabulary V_S^i.

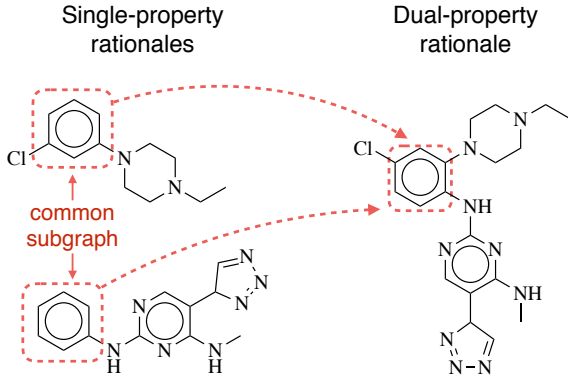**Multi-property Rationale** For a set of M properties, we

Figure 4. Illustration of multi-property rationale construction. Given two single-property rationales, we first find their maximum common substructure (MCS). If their MCS is not empty, we superpose one rationale on another so that their MCS coincides.

can similarly define its rationale $\mathcal{S}^{[M]}$ by imposing M property constraints at the same time, namely

$$\forall i : r_i(\mathcal{S}^{[M]}) \geq \delta_i, i = 1, \cdots, M$$

In principle, we can apply MCTS to extract rationales from molecules that satisfy all the property constraints. However, in many cases there are no such molecules available. To this end, we propose to construct multi-property rationales from single-property rationales extracted by MCTS. Specifically, each multi-property rationale $\mathcal{S}^{[M]}$ is merged from single-property rationales $\mathcal{S}^1, \cdots, \mathcal{S}^M$. We merge two rationales $\mathcal{S}^i$ and $\mathcal{S}^j$ by first finding their maximum common substructure (MCS) and then superposing $\mathcal{S}^i$ on $\mathcal{S}^j$ so that their MCS coincides (see Figure 4).[3] This gives us a set of candidate rationales:

$$C_{\mathcal{S}}^M = \bigcup_{(\mathcal{S}^1, \cdots, \mathcal{S}^M)} \text{MERGE}(\mathcal{S}^1, \cdots, \mathcal{S}^M) \quad (7)$$

where $(\mathcal{S}^1, \cdots, \mathcal{S}^M) \in V_{\mathcal{S}}^1 \times \cdots \times V_{\mathcal{S}}^M$. Note that the output of MERGE is a set as there are multiple ways of superposing two rationales. Finally, the vocabulary of multi-property rationales is the subset of $C_{\mathcal{S}}^M$ which satisfies all the property constraints:

$$V_{\mathcal{S}}^{[M]} = \{\mathcal{S} \in C_{\mathcal{S}}^M \mid r_i(\mathcal{S}^{[M]}) \geq \delta_i, \forall i\} \quad (8)$$

For notational convenience, we will denote both single and multi-property rationales as $\mathcal{S}$ from now on.

3.2. Graph Completion

This module is a variational autoencoder which completes a full molecule $\mathcal{G}$ given a rationale $\mathcal{S}$. Since each rationale $\mathcal{S}$

---

[3]The MCS of two (or multiple) rationales is computed using RDKit (Landrum, 2006).

can be realized into many different molecules, we introduce a latent variable $\boldsymbol{z}$ to generate diverse outputs:

$$P(\mathcal{G}|\mathcal{S}) = \int_{\boldsymbol{z}} P(\mathcal{G}|\mathcal{S}, \boldsymbol{z}) P(\boldsymbol{z}) d\boldsymbol{z} \qquad (9)$$

where $P(\boldsymbol{z})$ is the prior distribution. Different from standard graph generation, our graph decoder must generate graphs that contain subgraph $\mathcal{S}$. Our VAE architecture is adapted from existing atom-by-atom generative models (You et al., 2018b; Liu et al., 2018) to incorporate the subgraph constraint. For completeness, we present our architecture here:

Encoder Our encoder is a message passing network (MPN) which learns the approximate posterior $Q(\boldsymbol{z}|\mathcal{G}, \mathcal{S})$ for variational inference. Let $\boldsymbol{e}(a_u)$ be the embedding of atom u with atom type a_u, and $\boldsymbol{e}(b_{uv})$ be the embedding of bond (u, v) with bond type b_{uv}. The MPN computes atom representations $\{\boldsymbol{h}_v | v \in \mathcal{G}\}$.

$$\{\boldsymbol{h}_v\} = \text{MPN}_e \left(\mathcal{G}, \{\boldsymbol{e}(a_u)\}, \{\boldsymbol{e}(b_{uv})\}\right) \qquad (10)$$

For simplicity, we denote the MPN encoding process as $\text{MPN}(\cdot)$, which is detailed in the appendix. The atom vectors are aggregated to represent $\mathcal{G}$ as a single vector $\boldsymbol{h}_\mathcal{G} = \sum_v \boldsymbol{h}_v$. Finally, we sample latent vector $\boldsymbol{z}_\mathcal{G}$ from $Q(\boldsymbol{z}|\mathcal{G}, \mathcal{S})$ with mean $\boldsymbol{\mu}(\boldsymbol{h}_\mathcal{G})$ and log variance $\boldsymbol{\Sigma}(\boldsymbol{h}_\mathcal{G})$:

$$\boldsymbol{z}_\mathcal{G} = \boldsymbol{\mu}(\boldsymbol{h}_\mathcal{G}) + \exp(\boldsymbol{\Sigma}(\boldsymbol{h}_\mathcal{G})) \cdot \epsilon; \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \qquad (11)$$

Decoder The decoder generates molecule $\mathcal{G}$ according to its breadth-first order. In each step, the model generates a new atom and all its connecting edges. During generation, we maintain a queue $\mathcal{Q}$ that contains frontier nodes in the graph who still have neighbors to be generated. Let $\mathcal{G}_t$ be the partial graph generated till step t. To ensure $\mathcal{G}$ contains $\mathcal{S}$ as subgraph, we set the initial state of $\mathcal{G}_1 = \mathcal{S}$ and put all the peripheral atoms of $\mathcal{S}$ to the queue $\mathcal{Q}$ (only peripheral atoms are needed due to the rationale extraction algorithm).

In t^{th} generation step, the decoder first runs a MPN over current graph $\mathcal{G}_t$ to compute atom representations $\boldsymbol{h}_v^{(t)}$:

$$\{\boldsymbol{h}_v^{(t)}\} = \text{MPN}_d \left(\mathcal{G}_t, \{\boldsymbol{e}(a_u)\}, \{\boldsymbol{e}(b_{uv})\}\right) \qquad (12)$$

The current graph $\mathcal{G}_t$ is represented as the sum of its atom vectors $\boldsymbol{h}_{\mathcal{G}_t} = \sum_{v \in \mathcal{G}_t} \boldsymbol{h}_v^{(t)}$. Suppose the first atom in $\mathcal{Q}$ is v_t. The decoder decides to expand $\mathcal{G}_t$ in three steps:

1. Predict whether there will be a new atom attached to v_t:

$$\boldsymbol{p}_t = \text{sigmoid}(\text{MLP}(\boldsymbol{h}_{v_t}^{(t)}, \boldsymbol{h}_{\mathcal{G}_t}, \boldsymbol{z}_\mathcal{G})) \qquad (13)$$

where $\text{MLP}(\cdot, \cdot, \cdot)$ is a ReLU network whose input is a concatenation of multiple vectors.

2. If $\boldsymbol{p}_t < 0.5$, discard v_t and move on to the next node in $\mathcal{Q}$. Stop generation if $\mathcal{Q}$ is empty. Otherwise, create a new atom u_t and predict its atom type:

$$\boldsymbol{p}_{u_t} = \text{softmax}(\text{MLP}(\boldsymbol{h}_{v_t}^{(t)}, \boldsymbol{h}_{\mathcal{G}_t}, \boldsymbol{z}_\mathcal{G})) \qquad (14)$$

3. Predict the bond type between u_t and other frontier nodes in $\mathcal{Q} = \{q_1, \cdots, q_n\}$ ($q_1 = v_t$). Since atoms are generated in breadth-first order, there are no bonds between u_t and atoms not in $\mathcal{Q}$.

To fully capture edge dependencies, we predict the bonds between u_t and atoms in $\mathcal{Q}$ sequentially and update the representation of u_t when new bonds are added to $\mathcal{G}_t$. In the kth step, we predict the bond type of (u_t, q_k) as follows:

$$b_{u_t, q_k} = \text{softmax}\left(\text{MLP}(\boldsymbol{g}_{u_t}^{(k-1)}, \boldsymbol{h}_{q_k}^{(t)}, \boldsymbol{h}_{\mathcal{G}_t}, \boldsymbol{z}_\mathcal{G})\right) \qquad (15)$$

where $\boldsymbol{g}_{u_t}^{(k-1)}$ is the new representation of u_t after bonds $\{(u_t, q_1), \cdots, (u_t, q_{k-1})\}$ have been added to $\mathcal{G}_t$:

$$\boldsymbol{g}_{u_t}^{(k-1)} = \text{MLP}\left(\boldsymbol{e}(a_{u_t}), \sum_{j=1}^{k-1} \text{MLP}(\boldsymbol{h}_{q_j}^{(t)}, \boldsymbol{e}(b_{q_j, u_t}))\right)$$

3.3. Training Procedure

Our training objective is to maximize the expected reward of generated molecules $\mathcal{G}$, where the reward is an indicator of $r_i(\mathcal{G}) \geq \delta_i$ for all properties $1 \leq i \leq M$

$$\sum_\mathcal{G} \mathbb{I}\left[\bigwedge_i r_i(\mathcal{G}) \geq \delta_i\right] P(\mathcal{G}) + \lambda \mathbb{H}[P(\mathcal{S})] \qquad (16)$$

We incorporate an entropy regularization term $\mathbb{H}[P(\mathcal{S})]$ to encourage the model to explore different types of rationales. The rationale distribution $P(\mathcal{S})$ is a categorical distribution over the rationale vocabulary. Let $\mathbb{I}[\mathcal{G}] = \mathbb{I}[\bigwedge_i r_i(\mathcal{G}) \geq \delta_i]$. It is easy to show that the optimal $P(\mathcal{S})$ has a closed form solution:

$$P(\mathcal{S}_k) \propto \exp\left(\frac{1}{\lambda} \sum_\mathcal{G} \mathbb{I}[\mathcal{G}] P(\mathcal{G}|\mathcal{S}_k)\right) \qquad (17)$$

The remaining question is how to train graph generator $P(\mathcal{G}|\mathcal{S})$. The generator seeks to produce molecules that are realistic and positive. However, Eq.(16) itself does not take into account whether generated molecules are realistic or not. To encourage the model to generate realistic compounds, we train the graph generator in two phases:

- *Pre-training* $P(\mathcal{G}|\mathcal{S})$ using real molecules.
- *Fine-tuning* $P(\mathcal{G}|\mathcal{S})$ using policy gradient with reward from property predictors.

The overall training algorithm is shown in Algorithm 1.

3.3.1. PRE-TRAINING

In addition to satisfying all the property constraints, the output of the model should constitute a realistic molecule. For this purpose, we pre-train the graph generator on a large set of molecules from ChEMBL (Gaulton et al., 2017). Each training example is a pair $(\mathcal{S}, \mathcal{G})$, where $\mathcal{S}$ is a random connected subgraph of a molecule $\mathcal{G}$ with up to N

Algorithm 1 Training method with n property constraints.

1: **for** $i = 1$ to M **do**
2: $\quad V_S^i \leftarrow$ rationales extracted from existing molecules $\mathcal{D}_i^{pos}$ positive to property i. $\qquad$ (see §3.1)
3: **end for**
4: Construct multi-property rationales $V_S^{[M]}$.
5: Pre-train $P(\mathcal{G}|\mathcal{S})$ on the pre-training dataset $\mathcal{D}^{pre}$.
6: Fine-tune model $P(\mathcal{G}|\mathcal{S})$ on $\mathcal{D}^f$ for L iterations using policy gradient.
7: Compute $P(\mathcal{S})$ based on Eq.(17) using fine-tuned model $P(\mathcal{G}|\mathcal{S})$.

---

atoms. The task is to learn to expand a subgraph into a full molecule. In particular, we train the generative model $P(\mathcal{G}|\mathcal{S})$ to maximize the likelihood of the pre-training dataset $\mathcal{D}^{pre} = \{(\mathcal{S}_i, \mathcal{G}_i)\}_{i=1}^n$.

3.3.2. FINE-TUNING

After pre-training, we further fine-tune the graph generator on property-specific rationales $\mathcal{S} \in \mathcal{V}_S$ in order to maximize Eq.(16). The model is fine-tuned through multiple iterations using policy gradient (Sutton et al., 2000). Let $P_{\theta^t}(\mathcal{G}|\mathcal{S})$ be the model trained till t^{th} iteration. In each iteration, we perform the following two steps:

1. Initialize the fine-tuning set $\mathcal{D}^f = \emptyset$. For each rationale $\mathcal{S}_i$, use the current model to sample K molecules $\{\mathcal{G}_i^1, \cdots, \mathcal{G}_i^K\} \sim P_{\theta^t}(\mathcal{G}|\mathcal{S}_i)$. Add $(\mathcal{G}_i^k, \mathcal{S}_i)$ to set $\mathcal{D}^f$ if $\mathcal{G}_i^k$ is predicted to be positive.

2. Update the model $P_\theta(\mathcal{G}|\mathcal{S})$ on the fine-tuning set $\mathcal{D}^f$ using policy gradient method.

After fine-tuning $P(\mathcal{G}|\mathcal{S})$, we compute the rationale distribution $P(\mathcal{S})$ based on Eq.(17).

4. Experiments

We evaluate our method (RationaleRL) on molecule design tasks under various combination of property constraints. In our experiments, we consider the following two properties:

- GNK3β: Inhibition against glycogen synthase kinase-3 beta. The GNK3β prediction model is trained on the dataset from Li et al. (2018), which contains 2665 positives and 50K negative compounds.

- JNK3: Inhibition against c-Jun N-terminal kinase-3. The JNK3 prediction model is also trained on the dataset from Li et al. (2018) with 740 positives and 50K negatives.

Following Li et al. (2018), the property prediction model is a random forest using Morgan fingerprint features (Rogers & Hahn, 2010). We set the positive threshold $\delta_i = 0.5$. For each property, we split its property dataset into 80%, 10% and 10% for training, validation and testing. The test AUROC score is 0.86 for both GSK3β and JNK3.

Multi-property Constraints We also consider combinations of property constraints that are biologically relevant:

- GNK3β + JNK3: Jointly inhibiting JNK3 and GSK-3β may provide potential benefit for the treatment of Alzheimer's disease (Li et al., 2018). There exist 316 dual inhibitors already available in the dataset.

- GNK3β + JNK3 + QED + SA: We further require the generated dual inhibitors to be drug like and synthetically accessible (i.e., easy to be synthesized). These two properties are quantified by QED (Bickerton et al., 2012) and synthetic accessibility (SA) score (Ertl & Schuffenhauer, 2009). In particular, we require QED ≥ 0.6 and SA ≤ 4.0.

Evaluation Metrics Our evaluation effort measures various aspects of molecule design. For each method, we generate $n = 5000$ molecules and compute the following metrics:

- **Success**: The fraction of sampled molecules predicted to be positive (i.e., satisfying all property constraints). A good model should have a high success rate. Following previous work (Olivecrona et al., 2017; You et al., 2018a), we only consider the success rate under property predictors, as it is hard to obtain real property measurements for GSK3β and JNK3.

- **Diversity**: It is also important for a model generate diverse range of positive molecules. To this end, we measure the diversity of generated positive compounds by computing their pairwise molecular distance $\mathrm{sim}(X, Y)$, which is defined as the Tanimoto distance over Morgan fingerprints of two molecules.

$$\text{Diversity} = 1 - \frac{2}{n(n-1)} \sum_{X,Y} \mathrm{sim}(X,Y)$$

- **Novelty**: Crucially, a good model should discover novel positive compounds. In this regard, for each generated positive compound $\mathcal{G}$, we find its nearest neighbor $\mathcal{G}_{\text{SNN}}$ from positive molecules in the training set. We define the novelty as the fraction of molecules with nearest neighbor similarity lower than 0.4 (Olivecrona et al., 2017):

$$\text{Novelty} = \frac{1}{n} \sum_{\mathcal{G}} \mathbf{1} \left[\mathrm{sim}(\mathcal{G}, \mathcal{G}_{\text{SNN}}) < 0.4 \right]$$

Baselines We compare our method against the following state-of-the-art generation methods for molecule design:

- JT-VAE (Jin et al., 2018) is a generative model which generate molecules based on substructures such as rings and bonds. The model contains auxiliary property predictors over the VAE latent space. During testing, we perform multi-objective optimization in the latent space using the gradient from the property predictors.

Table 1. Results on molecule design with one or two property constraints.

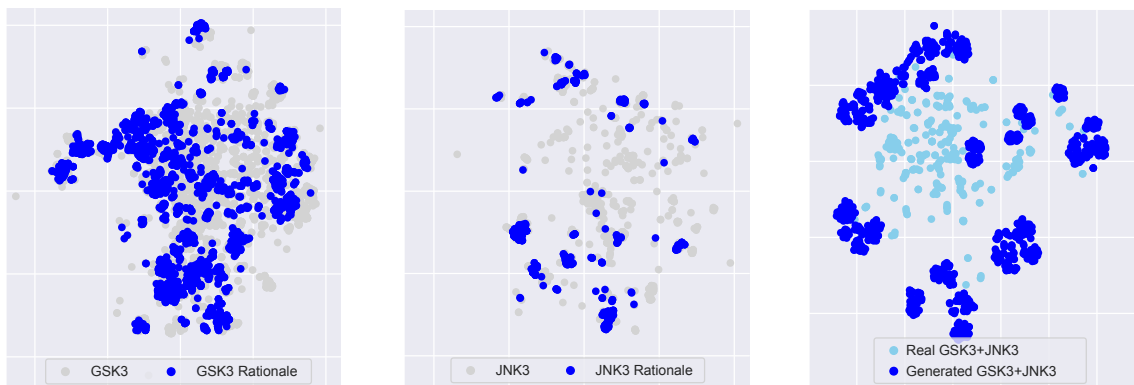| Method	GSK3β			JNK3			GSK3β + JNK3		
	Success	Novelty	Diversity	Success	Novelty	Diversity	Success	Novelty	Diversity
JT-VAE	32.2%	11.8%	0.901	23.5%	2.9%	0.882	3.3%	7.9%	**0.883**
GCPN	42.4%	11.6%	**0.904**	32.3%	4.4%	**0.884**	3.5%	8.0%	0.874
GVAE-RL	33.2%	76.4%	0.874	57.7%	62.6%	0.832	40.7%	80.3%	0.783
REINVENT	99.3%	**61.0%**	0.733	98.5%	31.6%	0.729	97.4%	39.7%	0.595
RationaleRL	**100%**	53.4%	0.888	**100%**	**46.2%**	0.862	**100%**	**97.3%**	0.824

Figure 5. **Left & middle**: t-SNE plot of the extracted rationales for GSK3β and JNK3. For both properties, rationales mostly covers the chemical space populated by existing positive molecules. **Right**: t-SNE plot of generated GSK3β+JNK3 dual inhibitors.

Table 2. Molecule design with four property constraints. The novelty and diversity of JT-VAE, GVAE-RL and GCPN are not reported due to their low success rate.

Method	GSK3β + JNK3 + QED + SA		
	Success	Novelty	Diversity
JT-VAE	1.3%	-	-
GVAE-RL	2.1%	-	-
GCPN	4.0%	-	-
REINVENT	47.9%	56.1%	0.621
RationaleRL	**74.8%**	**56.8%**	**0.701**

- REINVENT (Olivecrona et al., 2017) is a RL model generating molecules based on their SMILES strings (Weininger, 1988). To generate realistic molecules, their model is pre-trained over one million molecules from ChEMBL and then finetuned under property reward.

- GCPN (You et al., 2018a) is a RL model which generates molecular graphs atom by atom. It uses GAN (Goodfellow et al., 2014) to help generate realistic molecules.

- GVAE-RL is a graph variational autoencoder which generates molecules atom by atom. The graph VAE architecture is the same as our model, but it generates molecules from scratch without using rationales. The model is pre-

trained on the same ChEMBL dataset and fine-tuned for each property using policy gradient. This is an ablation study to show the importance of using rationales.

Rationales Details of the rationales used in our model:

- GSK3β, JNK3: For single properties, the rationale size is required to be less than 20 atoms. For each positive molecule, we run 20 iteration of MCTS with $c_{puct} = 10$. There are 6299 and 417 rationales for GSK3β and JNK3.

- GSK3β + JNK3: We construct two-property rationales by merging the single-property rationales (see Figure 4). For computational efficiency, we only consider rationales $\mathcal{S}_k$ receiving the highest $P(\mathcal{S}_k)$ within each property. This gives us two rationale subsets with $|V_{\mathcal{S}}^{\text{GSK3}\beta}| = 238$ and $|V_{\mathcal{S}}^{\text{JNK3}}| = 39$. In total, there are 9719 two-property rationales merged from $V_{\mathcal{S}}^{\text{GSK3}\beta}$ and $V_{\mathcal{S}}^{\text{JNK3}}$.

- GSK3β + JNK3 + QED + SA: We construct four-property rationales by selecting the two-property rationales whose QED ≥ 0.6 and SA ≤ 4.0. There are in total 61 different four-property rationales.

Model Setup We pre-train all the models on the same ChEMBL dataset, which contains 1.02 million training examples. On the four-property generation task, our model is fine-tuned for $L = 50$ iterations, with each rationale being expanded for $K = 200$ times.
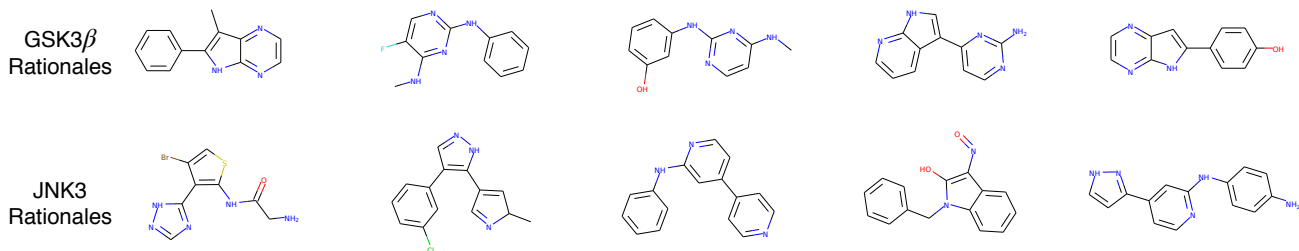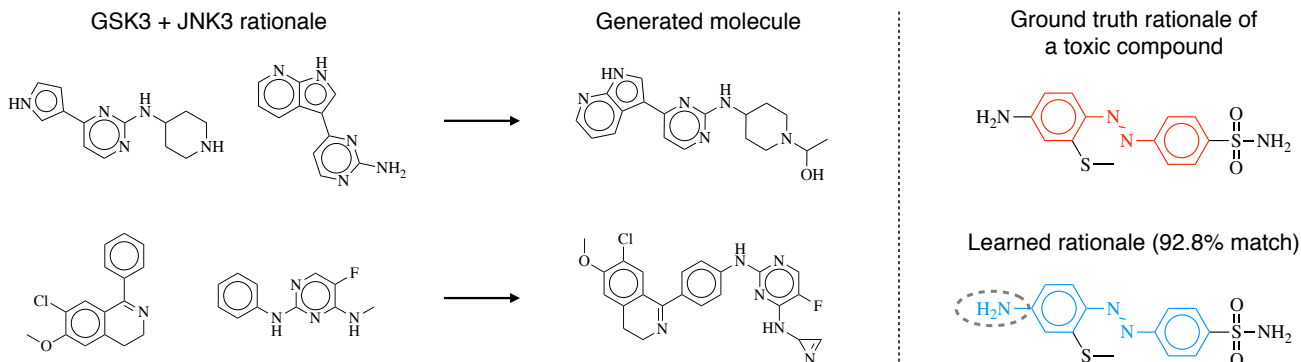
Figure 6. Sample rationales of GSK3β (top) and JNK3 (bottom).

Figure 7. **Left**: Examples of molecules generated in the GSK3β+JNK3+QED+SA task. The model learns to combine two disjoint rationale graphs into a complete molecule. **Right**: Example structural alerts in the toxicity dataset. The ground truth rationale (Azobenzene) is highlighted in red. Our learned rationale almost matches the ground truth (error highlighted in dashed circle).

4.1. Results

The results are reported in Table 1 and 2. On the single-property generation task, both REINVENT and RationaleRL demonstrate nearly perfect success rate since there is only one constraint. On the two-property generation task, our model achieves 100% success rate while maintaining 97.3% novelty and 0.824 diversity score, which is much higher than REINVENT. Meanwhile, GCPN has a low success rate (3.5%) due to reward sparsity.

Table 2 shows the results on the four-property generation task, which is the most challenging. The difference between our model the baselines become significantly larger. In fact, GVAE-RL and GCPN completely fail in this task due to reward sparsity. Our model outperforms REINVENT with a wide margin (success rate: 74.8% versus 47.9%; diversity: 0.701 versus 0.621).

Ablation study In all tasks, our method significantly outperforms GVAE-RL, which has the same generative architecture but does *not* utilize rationales and generates molecules from scratch. Thus we conclude the importance of rationales for multi-property molecular design.

Visualization We further provide visualizations to help understand our model. In Figure 5, we plotted a t-SNE (Maaten & Hinton, 2008) plot of the extracted rationales for GSK3β

and JNK3. For both properties, rationales mostly cover the chemical space populated by existing positive molecules. The generated GSK3β+JNK3 dual inhibitors are distributionally close to the true dual inhibitors in the training set. In Figure 6, we show samples of rationales extracted by MCTS. In Figure 7, we show examples of generated molecules that satisfy all the four constraints (GSK3β+JNK3+QED+SA).

4.2. Property Predictor Applicability

Since the reported success rate is based on property predictors, it is possible for the generated molecules to exploit flaws of the property predictors. In particular, the predicted properties may be unreliable when the distribution of generated compounds is very different from the distribution of molecules used to train the property predictors.

This issue is alleviated by our framework because our generated compounds are built from rationales extracted from true positive compounds used to train the property predictors. Therefore, our generated compounds are closer to the true compounds than compounds generated from scratch. To show this, we adopt Frechet ChemNet Distance (FCD)[4] to measure distributional discrepancy between generated molecules and positive compounds in the training set (Preuer

---

[4]The FCD score is analogous to the Frechet Inception Distance (Heusel et al., 2017) used in image synthesis.

Table 3. Frechet ChemNet Distance (FCD) between generated compounds and true positive molecules in the training set.

	GSK3β	JNK3	GSK3β+JNK3
REINVENT	28.63	25.16	36.75
RationaleRL	**6.88**	**9.68**	**25.52**

Table 4. Rationale accuracy on the toxicity dataset. Our rationales are more faithful to the property of interest.

Method	Partial Match	Exact Match
Integrated Gradient	0.857	39.4%
MCTS Rationale	**0.861**	**46.0%**

et al., 2018). As shown in Table 3, the FCD of RationaleRL is much lower than REINVENT, which means our generated compounds are distributionally closer to the true compounds. As a result, the predicted properties are more reliable for molecules generated by our model.

4.3. Faithfulness of Rationales

While our rationales are mainly extracted for generation, it is also important for them to be chemically relevant. In other words, the extracted rationales should accurately explain the property of interest. As there is no ground truth rationales available for JNK3 and GSK3β, we turn to an auxiliary toxicity dataset for evaluating rationale quality.

Data The toxicity dataset contains 125K molecules randomly selected from ChEMBL. Each molecule is labeled as toxic if it contains structural alerts (Sushko et al., 2012) — chemical substructures that is correlated with human or environmental hazards (see Figure 7).[5] Under this setup, the structural alerts are ground truth rationales and we evaluate how often the extracted rationales match them. The dataset is split into 105K for training and 20K for testing. In total, there are 26.5K toxic molecules and 164 types of structural alerts. We train a graph convolutional network (Yang et al., 2019) to predict toxicity, which achieves 0.99 AUROC score on the test set.

Results We compare our MCTS based rationale extraction with integrated gradient (Sundararajan et al., 2017), which has been applied to explain property prediction models (McCloskey et al., 2019). We report two metrics: *partial match* AUC (attribution AUC metric used in McCloskey et al. (2019)) and *exact match* accuracy which measures how often a rationale graph exactly matches the true rationale in the molecule. As shown in Table 4, our method significantly outperforms the baseline in terms of exact matching. The extracted rationales has decent overlap with true rationales,

with 0.86 partial match on average. Therefore, our model is capable of finding rationales faithful to the properties.

5. Conclusion

In this paper, we developed a rationale based generative model for molecular design. Our model generates molecules in two phases: 1) identifying rationales whose presence indicate strong positive signals for each property; 2) expanding rationale graphs into molecules using graph generative models and fine-tuning it towards desired combination of properties. Our model demonstrates strong improvement over prior reinforcement learning methods in various tasks.

Acknowledgements

References

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90, 2012.

Dai, H., Tian, Y., Dai, B., Skiena, S., and Song, L. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786*, 2018.

De Cao, N. and Kipf, T. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

Gaulton, A., Hersey, A., Nowotka, M., Bento, A. P., Chambers, J., Mendez, D., Mutowo, P., Atkinson, F., Bellis, L. J., Cibrián-Uhalte, E., et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2017.

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., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

---

[5]Structural alerts used in our paper are from surechembl.org/ knowledgebase/169485-non-medchem-friendly-smarts

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Guimaraes, G. L., Sanchez-Lengeling, B., Farias, P. L. C., and Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *International Conference on Machine Learning*, 2018.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical graph-to-graph translation for molecules. *arXiv preprint arXiv:1907.11223*, 2019a.

Jin, W., Yang, K., Barzilay, R., and Jaakkola, T. Learning multimodal graph-to-graph translation for molecular optimization. *International Conference on Learning Representations*, 2019b.

Kajino, H. Molecular hypergraph grammar with its application to molecular optimization. *arXiv preprint arXiv:1809.02745*, 2018.

Kang, S. and Cho, K. Conditional molecular design with deep generative models. *Journal of chemical information and modeling*, 59(1):43–52, 2018.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.

Landrum, G. Rdkit: Open-source cheminformatics. *Online). http://www. rdkit. org. Accessed*, 3(04):2012, 2006.

Lee, G.-H., Jin, W., Alvarez-Melis, D., and Jaakkola, T. S. Functional transparency for structured data: a game-theoretic approach. *arXiv preprint arXiv:1902.09737*, 2019.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

Li, Y., Zhang, L., and Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *arXiv preprint arXiv:1801.07299*, 2018.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. L. Constrained graph variational autoencoders for molecule design. *Neural Information Processing Systems*, 2018.

Ma, T., Chen, J., and Xiao, C. Constrained generation of semantically valid graphs via regularizing variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 7113–7124, 2018.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

McCloskey, K., Taly, A., Monti, F., Brenner, M. P., and Colwell, L. J. Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences*, 116(24):11624–11629, 2019.

Murray, C. W. and Rees, D. C. The rise of fragment-based drug discovery. *Nature chemistry*, 1(3):187, 2009.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

Popova, M., Isayev, O., and Tropsha, A. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7): eaap7885, 2018.

Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., and Klambauer, G. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Rosin, C. D. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3): 203–230, 2011.

Samanta, B., De, A., Jana, G., Chattaraj, P. K., Ganguly, N., and Gomez-Rodriguez, M. Nevae: A deep generative model for molecular graphs. *arXiv preprint arXiv:1802.05283*, 2018.

Seff, A., Zhou, W., Damani, F., Doyle, A., and Adams, R. P. Discrete object generation with reversible inductive construction. In *Advances in Neural Information Processing Systems*, pp. 10353–10363, 2019.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329*, 2017.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Simonovsky, M. and Komodakis, N. Graphvae: Towards generation of small graphs using variational autoencoders. *arXiv preprint arXiv:1802.03480*, 2018.

Stolle, M. and Precup, D. Learning options in reinforcement learning. In *International Symposium on abstraction, reformulation, and approximation*, pp. 212–223. Springer, 2002.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328. JMLR. org, 2017.

Sushko, I., Salmina, E., Potemkin, V. A., Poda, G., and Tetko, I. V. Toxalerts: a web server of structural alerts for toxic chemicals and compounds with potential adverse reactions, 2012.

Sutton, R. S., Precup, D., and Singh, S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 9240–9251, 2019.

You, J., Liu, B., Ying, R., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *arXiv preprint arXiv:1806.02473*, 2018a.

You, J., Ying, R., Ren, X., Hamilton, W. L., and Leskovec, J. Graphrnn: A deep generative model for graphs. *arXiv preprint arXiv:1802.08773*, 2018b.

Zhou, Z., Kearnes, S., Li, L., Zare, R. N., and Riley, P. Optimization of molecules via deep reinforcement learning. *arXiv preprint arXiv:1810.08678*, 2018.