ChemSpacE: Toward Steerable and Interpretable Chemical Space Exploration

Yuanqi Du  
Cornell University  
yd392@cornell.edu

Xian Liu  
CUHK  
alvinliu@ie.cuhk.edu.hk

Shengchao Liu  
Mila & Université de Montréal  
liusheng@mila.quebec

Jieyu Zhang  
University of Washington  
jieyuz2@cs.washington.edu

Bolei Zhou  
CUHK  
bzhou@ie.cuhk.edu.hk

Abstract

Discovering new structures in the chemical space is a long-standing challenge and has important applications to various fields such as chemistry, material science, and drug discovery. Deep generative models have been used in de novo molecule design to embed molecules in a meaningful latent space and then sample new molecules from it. However, the steerability and interpretability of the learned latent space remain much less explored. In this paper, we introduce a new task named molecule manipulation, which aims to align the molecular properties of the generated molecule and its latent activation in order to achieve interactive molecule editing. Then we develop a method called Chemical Space Explorer (ChemSpacE), which identifies and traverses interpretable directions in the latent space that align with molecular structures and property changes. Specifically, ChemSpacE leverages the properties of the learned latent space by generative models and utilizes linear models to identify such directions and thus is highly efficient in terms of training/inference time, data, and the number of oracle calls. Experiments show that ChemSpacE can efficiently steer the latent spaces of multiple state-of-the-art molecule generative models for interactive molecule discovery.

1 Introduction

Designing new molecules with desired properties is a critical problem with a range of applications in drug discovery and material science [1]. Traditional pipelines require exhaustive human efforts and domain knowledge, which are difficult to scale up. Recent studies exploit deep generative models to solve this problem by encoding molecules into a meaningful latent space, from which random samples are drawn and decoded to new molecules [2]. Such deep molecule generative models facilitate the design and development of drugs and materials [3, 4].

Despite the promising results of deep generative models for molecule generation, much less effort has been made to understand the learned representations. Most of the existing models are based on deep neural networks, which are known to be black-box lacking transparency [5]. Outside of the molecule generation domain, many attempts have been made to improve the interpretability of deep learning models from various aspects, e.g., representation space [6], model space [7], and latent space [8, 9]. In the molecule generation field, interpretability can be studied in two ways: (1) the interpretation of the learned latent space where traversing the value of latent vectors could lead to smooth molecular property change and (2) the interpretation of the chemical space that adjusting the molecular property could observe smooth structure change of molecules.

Preprint. Under review.
In addition, it remains challenging to generate molecules with desired properties. Previous works tackle the problem with latent space optimization-based, reinforcement learning-based, and searching-based methods to achieve property control of the generated molecules [10, 11]. Specifically, reinforcement learning-based algorithm [12] equips the model with rewards designed to encourage the molecule generative models to generate molecules with specific molecular properties. Latent space optimization-based algorithm takes advantage of the learnt latent space by molecule generative models and optimize the molecular properties via Bayesian Optimization [13]. Searching-based algorithm instead searches directly from the discrete and high-dimensional chemical space for molecules with optimal properties [14]. However, these work often have three major issues. (1) They require many times of expensive oracle calls to provide feedback (i.e., property scores) of the intermediate molecules during the searching or optimization process [15]. (2) They often only focus on the outcome of the process while ignoring the intermediate steps of the process which is rather essential for chemists and pharmacologists to understand the chemical instances and rules. (3) They stick to local gradients while putting less focus on global directions in the chemical/latent space. (4) The molecular properties considered in the existing work are limited to a small set of molecular properties, such as penalized logP (octanol-water partition coefficient), QED (drug-likeness), DRD2 activity (binding affinity), etc [10, 11, 13, 16].

To tackle the above challenges, we formulate a new task, molecule manipulation, which aims at manipulating the properties of generated molecules by leveraging the steerability and interpretability of molecule generative models. Broad observations of the latent space learned by molecule generative models have been observed [17, 18]: (1) molecules sharing similar structures/properties tend to cluster in the latent space, (2) interpolating two molecules in the latent space lead to smooth changes in molecular structures/properties, we develop ChemSpace Explorer, a model-agnostic and efficient method to manipulate molecules with smooth changes of molecular structures and properties which has critical applications in molecule optimization, chemical space interpretation, etc. Specifically, ChemSpace Explorer first identifies the property separation hyperplane which defines the binary boundary corresponding to some molecular property (e.g., drug-like or drug-unlike) in the learned latent space of a generative model. Based on the identified property separation hyperplane, we estimate the latent directions that govern molecular properties, which enable smooth change of the molecular structures and properties without re-training the given molecular generative model. To the best of our knowledge, this work is the first attempt to achieve interactive molecule discovery through the manipulating of pre-trained molecule generative models.

The experiments demonstrate that our method can effectively steer the state-of-the-art molecule generative models for molecule manipulation with a very small amount of training/inference time, data, and oracle calls. To quantitatively measure the performance of molecule manipulation, we design two new evaluation metrics named strict success rate and relax success rate, which evaluate the percentage of successful manipulations with smooth property-changing molecules over manipulations of a group of molecules. To facilitate the interactive molecule design and discovery for practitioners, we further develop an interface to visualize the real-time molecule manipulations and smooth molecular structure/property changes. Our main contributions are summarized as follows:

- We formulate molecule manipulation, a new task that steers the latent space of molecule generative models to manipulate the chemical properties of the output molecule.
- We develop an efficient model-agnostic method named ChemSpace Explorer for molecule manipulation, which can be incorporated in various state-of-the-art molecule generative models.
- Comprehensive experiments demonstrate the effectiveness of our method in quantifying the steerability of various molecule generative models. An interface is developed to exhibit the real-time molecule discovery and design.

2 Related Work

Molecule Generation. Recent studies have explored a variety of deep generative models for molecule generation [19], such as variational autoencoders (VAEs) [11], generative adversarial networks (GANs) [20], normalizing flows [10, 21, 22], energy-based models (EBMs) [23], reinforcement learning [24–26], etc [27, 28]. To be specific, JT-VAE [11] proposes a VAE-based architecture to encode both atomic graphs and structural graphs for efficient molecule generation. MolGAN [20] exploits GANs for molecule generation, where discriminators are used to encourage the model to
To leverage the steerability and interpretability of molecule generative models, we propose a new approach to observe the alignment of \( Z \rightarrow X \rightarrow P \) for molecules with properties \( p \) and a list of \( p \). Molecular properties \( P \) define the property space \( \mathcal{P} \), commonly assumed to be a Gaussian distribution \([44, 45]\). There exist property functions \( g \) that contain a generator \( g \) and a latent space. They can be formulated as:

\[
\begin{align*}
    f &: \mathcal{X} \rightarrow \mathbb{R}^D, \\
    g &: \mathbb{R}^D \rightarrow \mathcal{P},
\end{align*}
\]

In molecule generation, a generative model \( M \) encodes the molecular graph \( \mathcal{G} \) as a set of edges \( E \subseteq V \times V \) and a set of node features \( F \). They can be formulated as:

\[
\begin{align*}
    V &: \{0, 1\}^{N \times D}, \\
    E &: \{0, 1\}^{K \times D}, \\
    F &: \{0, 1\}^{K \times D},
\end{align*}
\]

Deep Molecule Generative Models. In molecule generation, a generative model \( M \) encodes the molecular graph \( \mathcal{G} \) as a latent vector \( Z \in \mathbb{R}^l \) with \( l \) being the dimension of the latent space and is capable of decoding any latent vector back to the molecular space. Specifically, variational auto-encoder (VAE) \([44]\) and flow-based model (Flow) \([45]\) are the two most commonly used models for molecule generation tasks. Both of them encode the data from molecular space to latent space, which is usually modeled as a Gaussian distribution; then they decode the latent code back to molecular space. They can be formulated as:

\[
\begin{align*}
    z &= f(x), \\
    x' &= g(z),
\end{align*}
\]

where \( x \) and \( x' \) are the ground-truth and reconstructed/sampled data respectively, and \( z \in Z \) represents a latent vector in the latent space, \( f(.) \) and \( g(.) \) are the encoder and generator/decoder of the generative model.

### 3 Preliminaries

Molecule Graph. A molecule can be presented as a graph \( \mathcal{G} = (V, E, F) \), where \( V \) denotes a set of \( N \) vertices (i.e., atoms), \( E \subseteq V \times V \) denotes a set of edges (i.e., bonds), \( F \subseteq \{0, 1\}^{N \times D} \) denotes the node features (i.e., atom types) and \( E \subseteq \{0, 1\}^{K \times D} \) denotes the edge features (i.e., bond types). The number of atom types and bond types are denoted by \( D \) and \( K \), respectively.

Deep Molecule Generative Models. In molecule generation, a generative model \( M \) encodes the molecular graph \( \mathcal{G} \) as a latent vector \( Z \in \mathbb{R}^l \) with \( l \) being the dimension of the latent space and is capable of decoding any latent vector back to the molecular space. Specifically, variational auto-encoder (VAE) \([44]\) and flow-based model (Flow) \([45]\) are the two most commonly used models for molecule generation tasks. Both of them encode the data from molecular space to latent space, which is usually modeled as a Gaussian distribution; then they decode the latent code back to molecular space. They can be formulated as:

\[
\begin{align*}
    z &= f(x), \\
    x' &= g(z),
\end{align*}
\]

where \( x \) and \( x' \) are the ground-truth and reconstructed/sampled data respectively, and \( z \in Z \) represents a latent vector in the latent space, \( f(.) \) and \( g(.) \) are the encoder and generator/decoder of the generative model.

### 4 Problem Formulation of Molecule Manipulation

To leverage the steerability and interpretability of molecule generative models, we propose a new task, molecule manipulation, which interprets and steers the latent space of the generative model in order to manipulate the properties of the output molecule. To be specific, a deep generative model contains a generator \( g: \mathbb{R}^l \rightarrow \mathcal{X} \), where \( \mathcal{X} \rightarrow \mathbb{R}^l \) stands for the \( l \)-dimensional latent space, which is commonly assumed to be Gaussian distribution \([44, 45]\). There exist property functions \( f_P \) which define the property space \( \mathcal{P} \) via \( P = f_P(X) \).

Formulation. The input to molecule manipulation is a list of \( n \) molecules \( X = \{x_1, x_2, \ldots, x_n\} \) and a list of \( m \) molecular properties \( P = \{p_1, p_2, \ldots, p_m\} \). We aim to manipulate one or more molecular properties \( p \) of a given molecule in a \( k \) consecutive steps and output the manipulated molecules with properties \( p' = \{p^{(1)}, p^{(2)}, \ldots, p^{(k)}\} \). By manipulating the given molecule, we can observe the alignment of \( Z \rightarrow \mathcal{X} \rightarrow \mathcal{P} \), where the relationship between \( Z \) and \( \mathcal{X} \) explains the latent
space of molecule generative models. The relationship between $\mathcal{X}$ and $\mathcal{P}$ reveals the correlations between molecular structures and properties. By traversing latent space, we can generate molecules with continuous structure/property changes.

**Evaluation criteria.** There are two important measures to evaluate the molecule manipulation task: smooth structure change and smooth property change. To be specific, we design two new evaluation metrics named strict success rate (SSR) and relaxed success rate (RSR) that measure the quality of the identified latent direction in controlling the molecular property. Under strict success rate, we consider a manipulation path to be successful only if we generate molecules with monotonically-changing properties and structures in consecutive $k$ steps of manipulation. The constraints are formulated as follows:

$$\phi_{SPC}(x, k, f) = 1[\forall i \in [k], \text{s.t.}, f(x^{(i)}) - f(x^{(i+1)}) \leq 0],$$

$$\phi_{SSC}(x, k, \delta) = 1[\forall i \in [k], \text{s.t.}, \delta(x^{(i+1)}, x^{(1)}) - \delta(x^{(i)}, x^{(1)}) \leq 0],$$

$$\phi_{DIV}(x, k) = 1[\exists i \in [k], \text{s.t.}, x^{(i)} \neq x^{(1)}],$$

where $f$ is a property function which calculates certain molecular property, $\delta$ denotes structure similarity between molecules $x^{(i)}$, $x^{(i+1)}$ generated in two adjacent manipulation steps. $\phi_{SPC}$ defines the strict property constraint; $\phi_{SSC}$ defines the strict structure constraint; $\phi_{DIV}$ defines the diversity constraint. The strict success rate is defined as:

$$SSR = L(P, X, k) = \frac{1}{|P| \times |X|} \sum_{p \in P, x \in X} 1[\phi_{SPC}(x, k, f, p) \land \phi_{SSC}(x, k) \land \phi_{DIV}(x, k)],$$

As monotonicity is rather strict, we propose a more relaxed definition of success rate, namely relaxed success rate, constructed via relaxed constraints, as follows:

$$\phi_{RPC}(x, k, f, \epsilon) = 1[\forall i \in [k], \text{s.t.}, f(x^{(i)}) - f(x^{(i+1)}) \leq \epsilon],$$

$$\phi_{RSC}(x, k, \delta, \gamma) = 1[\forall i \in [k], \text{s.t.}, \delta(x^{(i+1)}, x^{(1)}) - \delta(x^{(i)}, x^{(1)}) \leq \gamma],$$

$$\phi_{DIV}(x, k) = 1[\exists i \in [k], \text{s.t.}, x^{(i)} \neq x^{(1)}],$$

where $\epsilon$ is a predefined tolerance threshold that weakens the monotonicity requirement. We also provide two implementations of relaxed success rate, which defines different tolerance variables $\epsilon$ with local relaxed constraint (RSR-L) and global relaxed constraint (RSR-G). For global constraint, we obtain $\epsilon$ by calculating the possible values (ranges) of the molecular properties in the training dataset, while for local constraint, we obtain $\epsilon$ by calculating the possible values (ranges) of the molecular properties only in the specific manipulation paths. The formulation of RSR-L and RSR-G is as follows:
Note even though it is more challenging for the model to pass RSR-L with local constraint (smaller range) while evaluating the successful path, its extra benefit is to take into account the ability of the model to manipulate one molecular property (i.e., the larger the range, the higher the tolerance score, thus the better chance to achieve successful manipulation).

5 ChemSpacE for Molecule Manipulation

5.1 Latent Cluster Assumption

We examine the property of latent space learned by the generative models and have the following observations, (1) molecules with similar structures tend to cluster in the latent space, (2) interpolating two molecules $x_1$ and $x_2$, represented by latent vectors $z_1$ and $z_2$, can lead to a list of intermediate molecules whose structures/properties gradually change from $x_1$ to $x_2$. As molecular structures determine molecular properties [47], the observations imply that molecules with similar property values of certain molecular property would cluster together and interpolating two molecules with different values of the molecular property could lead to gradual changes in molecular structures. As shown in Fig. 1, there may exist two groups of molecules, drug-like and drug-unlike, where each group cluster together and linear interpolating two latent vectors with one molecule from each group could lead to a direction that crosses the property separation boundary. These observations also match the analysis from the prior work [17, 18]. To verify our assumption, we visualize the latent space of the pre-trained MoFlow model in Fig. 2. The left figure shows that molecules close in the latent space are similar in structures. The middle figure shows that interpolating two molecules in the latent space could lead to smooth structure changes. The right figure shows that the latent boundary is present for QED property in the pre-trained MoFlow model.

5.2 Identifying Latent Directions

Latent Separation Boundary. With the verifications above and the previous work of analyzing the latent space of generative models [8, 48–50], we assume that there exists a separation boundary which separates groups of molecules for each molecular property (e.g., drug-like and drug-unlike) and the normal vector of the separation boundary defines a latent direction which controls the degree of the property value (in Fig. 1). When $z$ moves toward and crosses the boundary, the molecular properties change accordingly (e.g., from drug-unlike to drug-like). A perfect separation boundary would have molecules with different properties well separated on different sides. From that, we can find a separation boundary for each molecular property with a unit normal vector $n \in \mathbb{R}^l$, such that

$$RSR - L(P, X, k, \epsilon_l, \gamma) = \frac{1}{|P| \times |X|} \sum_{p \in P, x \in X} 1[\phi_{RPC}(x_p, k, f_p, \epsilon_l) \land \phi_{RSC}(x_p, k, \gamma) \land \phi_{DIV}(x_p, k)],$$

$$RSR - G(P, X, k, \epsilon_g, \gamma) = \frac{1}{|P| \times |X|} \sum_{p \in P, x \in X} 1[\phi_{RPC}(x_p, k, f_p, \epsilon_g) \land \phi_{RSC}(x_p, k, \gamma) \land \phi_{DIV}(x_p, k)],$$

Note even though it is more challenging for the model to pass RSR-L with local constraint (smaller range) while evaluating the successful path, its extra benefit is to take into account the ability of the model to manipulate one molecular property (i.e., the larger the range, the higher the tolerance score, thus the better chance to achieve successful manipulation).
Table 1: Quantitative Evaluation of Molecule Manipulation over a variety of molecular properties (numbers reported are strict success rate in %, -R denotes model with random manipulation, -L denotes model with the largest range manipulation, -O denotes optimization-based manipulation, -C denotes model with ChemSpacE. The best performances are bold.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Models</th>
<th>Avg.</th>
<th>QED</th>
<th>LogP</th>
<th>SA</th>
<th>DRD2</th>
<th>JNK3</th>
<th>GSK3B</th>
<th>MolWt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MoFlow-R</td>
<td>1.65</td>
<td>1.50</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-L</td>
<td>3.43</td>
<td>1.50</td>
<td>1.00</td>
<td>0.50</td>
<td>0.00</td>
<td>1.50</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-O</td>
<td>N/A</td>
<td>3.50</td>
<td>6.00</td>
<td>6.50</td>
<td>2.00</td>
<td>8.00</td>
<td>8.50</td>
<td>7.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-C</td>
<td>37.52</td>
<td>12.50</td>
<td>9.00</td>
<td>10.00</td>
<td>11.00</td>
<td>45.50</td>
<td>16.50</td>
<td>10.50</td>
</tr>
<tr>
<td>QM9</td>
<td>HierVAE-R</td>
<td>29.29</td>
<td>1.00</td>
<td>1.50</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>HierVAE-L</td>
<td>30.69</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>2.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>HierVAE-C</td>
<td>66.23</td>
<td>27.00</td>
<td>32.00</td>
<td>35.00</td>
<td>41.50</td>
<td>51.50</td>
<td>30.00</td>
<td>39.50</td>
</tr>
<tr>
<td>ZINC</td>
<td>MoFlow-R</td>
<td>4.25</td>
<td>1.50</td>
<td>1.50</td>
<td>2.50</td>
<td>3.00</td>
<td>3.50</td>
<td>1.50</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>MoFlow-L</td>
<td>5.61</td>
<td>1.50</td>
<td>6.50</td>
<td>2.00</td>
<td>6.00</td>
<td>2.50</td>
<td>4.00</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-O</td>
<td>N/A</td>
<td>1.50</td>
<td>9.50</td>
<td>0.50</td>
<td>2.00</td>
<td>15.50</td>
<td>23.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>MoFlow-C</td>
<td>58.08</td>
<td>52.00</td>
<td>53.50</td>
<td>51.50</td>
<td>55.00</td>
<td>56.50</td>
<td>55.50</td>
<td>53.50</td>
</tr>
<tr>
<td>ChEMBL</td>
<td>HierVAE-R</td>
<td>25.59</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HierVAE-L</td>
<td>22.98</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HierVAE-C</td>
<td>47.70</td>
<td>0.50</td>
<td>3.00</td>
<td>3.00</td>
<td>6.00</td>
<td>7.50</td>
<td>5.50</td>
<td>4.50</td>
</tr>
</tbody>
</table>

the distance from any sample $z$ to the separation boundary as:

$$d(z, n) = n^T z$$  \hspace{1cm} (11)

**Latent Direction.** In the latent space, the molecular structure and property change smoothly towards the new property class when $z$ moves towards the separation boundary and vice versa, where we assume linear dependency between $z$ and $p$:

$$f_P(g(z)) = \alpha \cdot d(z, n),$$  \hspace{1cm} (12)

where $f_P$ is an oracle function and $\alpha$ is a degree scalar that scales the changes along that corresponding direction. Extending the method to multiple molecular properties manipulation, we have:

$$f_P(g(z)) = A N^T z,$$  \hspace{1cm} (13)

where $A = Diag(a_1, \cdots, a_m)$ is the diagonal matrix with linear coefficients for each of the $m$ molecular properties and $N = [n_1, \cdots, n_m]$ represents normal vectors for the separation boundaries of $m$ molecular properties. We have the molecular properties $P$ following a multivariate normal distribution via:

$$\mu_P = E(A N^T z) = A N^T E(z) = 0,$$  \hspace{1cm} (14)

$$\Sigma_P = E(A N^T z z^T N A^T) = A N^T E(z z^T) N A^T = A N^T N A^T.$$  \hspace{1cm} (15)

We have all disentangled molecular properties in $P$ if and only if $\Sigma_P$ is a diagonal matrix and all directions in $N$ are orthogonal with each other. Nevertheless, not all molecular properties are purely disentangled with each other. In that case, molecular properties can correlate with each other and $n_i^T n_j$ is used to denote the entanglement between the $i$-th and $j$-th molecular properties in $P$.

### 5.3 Molecule Manipulation

After we find the separation boundary and identify the latent direction, to manipulate the generated molecules with desired properties, we first move from latent vector $z$ along the direction $n$ with a degree scalar $\alpha$, and the new latent vector is

$$z' = z + \alpha n$$  \hspace{1cm} (16)

To this end, the expected property of the new manipulated molecule is

$$f_P(g(z + \alpha n)) = f_P(g(z)) + k\alpha,$$  \hspace{1cm} (17)
where $k$ is a scaling factor between molecular vector space and property. Based on our assumption to find a separation boundary for each molecular property, we could utilize any linear model (e.g. linear Support Vector Machine) [51] to find the separation boundaries which best separate the two classes of the data. For each molecular property, we train an individual model from a group of randomly sampled latent vectors and utilize a property function $f_P$ to calculate the corresponding molecular properties. Then, we find the separation boundary for each molecular property. The normal vectors $N$ of separation boundaries are finally utilized as identified latent directions that govern the molecular properties. Additionally, our method is highly efficient in terms of data, training time and offline oracle calls thanks to leveraging shallow models with only a small number of data and their pre-calculated molecular properties.

6 Experiments

6.1 Setup

Datasets. We use three molecule datasets, QM9 [52], ZINC250K [53], and ChEMBL [54]. QM9 contains 134k small organic molecules with up to 9 heavy atoms (C, O, N, F). ZINC is a free database of commercially-available compounds for drug discovery. On average, the molecules in ZINC are bigger ($\sim$23 heavy atoms) and structurally more complex than QM9. We take a sampled 250K molecules version [17] from the larger database. ChEMBL is a manually curated database of bioactive molecules with drug-like properties and contains $\sim$1.8 million molecules.

Baselines. We include two baseline methods of identifying latent direction that governs the molecular property and one optimization-based method, which optimizes the molecular property of the generated molecules via gradient ascent/descent for comparisons. Random manipulation randomly samples latent directions for molecular properties. Largest range manipulation draws latent vectors from the training set and defines the directions via calculating the direction between one molecule with the largest property score and another molecule with the smallest property score for each molecular property. Optimization-based method optimizes the molecular property of the generated molecules by searching a latent vector with the optimized molecular property via gradient ascent/descent.

Implementation Details. We take the publicly available pre-trained models from the GitHub Repository of HierVAE and MoFlow, respectively. We utilize the implementation of linear models (linear SVM) from Scikit-learn [55].

6.2 Evaluation Protocols

Pre-trained Models. We apply ChemSpacE, as well as baselines, on two state-of-the-art molecule generative models with publicly available pre-trained models. HierVAE [56] embeds molecular structure motifs into a hierarchical VAE-based generative model; MoFlow [18] designs a normalizing flow-based model which learns an invertible mapping between input molecules and latent vectors. Molecular Properties. We study molecular properties identified in the chemistry community through open-source cheminformatics software, RDKit [57] and protein binding affinity, synthesis accessibility oracles in TDC [15]. In total, we analyze 212 molecular properties from multiple dimensions, including distributions, inter-correlations, etc. Details can be found in Appendix B. Due to the page limit, we mainly report results for 7 molecular properties, including 4 very common yet important ones, drug-likeness (QED), molecular weight (MolWt), partition coefficient (LogP), synthesis accessibility (SA), and 3 binding affinity scores. For continuous molecular properties, we take the molecules with largest and smallest properties for training the linear models.

Quantitatively, we evaluate the ability of the model to manipulate the given molecular property of molecules with the proposed strict success rate and relaxed success rate-L/G metrics (see Sec. 4). We evaluate the model’s efficiency by comparing the training process of the linear models with a neural network-based predictor for a commonly used optimization-based method in terms of training/inference time, data, and number of oracle calls. Qualitatively, we visualize molecule manipulation including property distribution shift during manipulation, single and multiple property manipulations.
Table 2: Efficiency in terms of training/inference time, data, and number of oracles of ChemSpacE compared to the optimization-based method.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>Training(s)</th>
<th>Inference/Path(s)</th>
<th># Data</th>
<th># Oracle calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt-based</td>
<td>QM9</td>
<td>137.03</td>
<td>0.02</td>
<td>120k</td>
<td>120k</td>
</tr>
<tr>
<td></td>
<td>ZINC</td>
<td>1027.26</td>
<td>0.04</td>
<td>200k</td>
<td>200k</td>
</tr>
<tr>
<td>ChemSpacE</td>
<td>QM9</td>
<td>0.05</td>
<td>0</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>ZINC</td>
<td>0.95</td>
<td>0</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Speedup</td>
<td>QM9</td>
<td>2740×</td>
<td>0.02↑</td>
<td>400×</td>
<td>400×</td>
</tr>
<tr>
<td></td>
<td>ZINC</td>
<td>1080×</td>
<td>0.04↑</td>
<td>500×</td>
<td>500×</td>
</tr>
</tbody>
</table>

Figure 3: Manipulating QED, MolWt and LogP properties of sampled molecules. The backbone model is CGVAE trained on QM9 dataset.

6.3 Quantitative Evaluation of Molecule Manipulation

In Table 1 and 2, we report the quantitative evaluation results for molecule manipulation with both strict success rate and relaxed success rate-L/G and training, inference time, data, oracle calls efficiency, which are evaluated on 212 molecular properties over 1,000 randomly generated molecules. According to the table, we can obtain the following insights.

(1) Our proposed method, ChemSpacE, as the first attempt for molecule manipulation, achieves excellent performance to manipulate properties of molecules with two state-of-the-art molecule generative models (VAE-based and Flow-based). For some important molecular properties (e.g., QED), we (with MoFlow) achieve 52% manipulation strict success rate in ZINC dataset. We outperform the baseline methods 6× on average.

(2) The baseline (random manipulation) method sometimes “finds” directions that control molecular properties. As shown in Fig. 2, the molecules are well-clustered in the latent space with respect to structures that determine molecular properties [47]. However, the largest range manipulation works worse possibly due to its strong assumption in determining the direction via the molecules with extreme properties (largest property and smallest property) in the dataset.

(3) The ChemSpacE method outperforms the popular optimization-based method in both generating smooth manipulation path, time and data efficiency. In Table 2, ChemSpacE speeds up the training time for at least 1000×, required data for at least 400×, and required oracle calls for at least 400×.

Figure 4: Visualization of Molecular property distribution shift while manipulating molecules with MoFlow on QM9 dataset (0 denotes the randomly sampled base molecule and +x and −x denote manipulation directions and steps).
6.4 Qualitative Evaluation of Molecule Manipulation and Interpretation

In Fig. 4, we visualize the property distributions of QED, MolWt and LogP along a 7-step manipulation path. For each step, we draw a property distribution. The candidate molecules are at place 0 and we attempt to manipulate the molecular property to the left (lower) and the right (higher). From the figure, we can clearly observe that the property distribution shifts to the left and right accordingly when we manipulate the molecule to the left and right. For example, when we manipulate the molecules three steps to the left, the range of QED shifts from $[0, 0.7]$ to $[0, 0.5]$; when the molecules are manipulated three steps to the right, there are much more molecules that have QED > 0.5 than the base distribution. Similar trends can also be seen for MolWt and LogP properties.

**Single Property Manipulation.** To qualitatively evaluate the performance of our method for molecule manipulation, we randomly select the successful manipulation paths from all three generative models in Fig. 3. The figures show that our method successfully learns interpretable and steerable directions. For example, for HierVAE in Fig. 3, we can find that gradually increasing LogP of a molecule may lead to the removal of the heavy atoms O and N from the structure. With respect to QED, the molecule drops double bonds, as well as heavy N and O atoms, when increasing QED for the HierVAE model. A similar trend can be observed in the MoFlow model that increasing QED drops double bonds and O atoms on the left of Fig. 3.

**Multi-Property Manipulation.** When it comes to multi-property manipulation, the goal is to control multiple molecular properties of a given molecule at the same time. In Fig. 5 (left), we show how our method manipulates multiple molecular properties. For simplicity, we remove the duplicate molecules and only leave the distinct molecules during the manipulation. From the figure, we can observe some correlations between LogP and QED since when we increase QED, LogP also increases accordingly. However, it is not always the case as moving the molecules to the right in the second row does not increase the QED scores. One potential reason is that the chemical space is vast, discrete and complex, and it is nontrivial to manipulate only one property while keep others the same of a molecule. An interactive demo is provided at https://drive.google.com/drive/folders/1N036p_50fWvGZybgPJ3Vw1CNXHVepimSR?usp=sharing and shown in Fig. 5 (right).

7 Conclusion

In this work, we formulate a new task of molecule manipulation and develop an efficient method called ChemSpacE to improve the steerability and interpretability of molecular generative models. The interface illustrates the promising application of interactive molecule design and discovery. Nevertheless, some challenging chemical phenomena, such as activity cliff, are not yet studied in this work. In addition, exploring the chemical space with unbiased pre-trained generative models is still nascent, it is also worth discovering a biased latent space for more effective molecule manipulation in our future work.
References


Appendix for
“ChemSpacE: Toward Steerable and Interpretable Chemical Space Exploration”

A Molecule Generative Models

In Table 3, we summarize a list of representative molecule generative models, which span various types of deep generative models, including the type of generative models, the type of generation process and whether latent space is learned. We also provide the formulation for two types of deep generative models (VAE and Flow) in Section A that are very popular for molecule generation task.

Table 3: A summary of mainstream molecule generative models.

<table>
<thead>
<tr>
<th>Prior Work</th>
<th>Generative Model</th>
<th>Sequential</th>
<th>Latent Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGVAE [13]</td>
<td>VAE</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MRNN [29]</td>
<td>RNN</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GraphNVP [21]</td>
<td>Flow</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GCPN [12]</td>
<td>RL</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GraphAF [10]</td>
<td>Flow</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MoFlow [18]</td>
<td>Flow</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HierVAE [56]</td>
<td>VAE</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GraphEBM [23]</td>
<td>EBM</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GraphDF [22]</td>
<td>Flow</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

A.1 Molecule Generative Model Formulation

**VAE.** gets a lower bound (ELBO) for the data log probability by introducing a proposal distribution.

\[
\log p(x) = \log \int z p(x|z)p(z)dz \\
\geq \log[E_{q(z|x)}[p(x|z)] + KL(q(z|x)||p(z))] \tag{18}
\]

**Flow.** The key of Flow model is to design an invertible function with the following nice property:

\[
z_0 \sim p_0(z_0) \\
z_i = f_i(z_{i-1}) \\
z_{i-1} = f_{i-1}^{-1}(z_i) \\
p_i(z_i) = p_{i-1}(z_{i-1}) \left| \det \frac{df_i^{-1}}{dz_i} \right| = p_{i-1}(f_{i-1}^{-1}(z_i)) \left| \det \frac{df_i^{-1}}{dz_i} \right|. \tag{19}
\]

where \( f_i \) is invertible function. To be more concrete, we can take \( z_0 \) as some tractable noise distribution, like Gaussian distribution, and repeating this for \( K \) steps will lead to the data distribution, i.e.:

\[ x = z_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(z_0). \]

Thus, the log likelihood of the data is as follows:

\[
\log p(x) = \log p_K(z_K) \\
= \log p_{K-1}(z_{K-1}) - \log \left| \det \frac{df_K}{dz_{K-1}} \right| \\
= \log p_{K-2}(z_{K-2}) - \log \left| \det \frac{df_{K-1}}{dz_{K-2}} \right| - \log \left| \det \frac{df_K}{dz_{K-1}} \right| \tag{20}
\]

= ... \\
= \log p_0(z_0) - \sum_{i=1}^{K} \log \left| \det \frac{df_i}{dz_{i-1}} \right|
B Study of Molecular Properties


Inter-correlations of molecular properties. In Fig. 6, we visualize the linear correlations between each pair of molecular properties across three datasets. From the heatmaps, we can observe that there are no linear correlations between half of the molecular properties, and similar patterns are observed in ZINC and ChEMBL datasets.

Molecular Property Distributions. We visualize 7 molecular property distributions reported in section 6 in Fig. 7. From there, we can observe that the property distribution may vary a lot in terms of different datasets. Notably, the distributions of some properties, e.g., QED, are very similar in ZINC and ChEMBL datasets, while some are quite different, e.g., MolWt.

\textsuperscript{1}https://www.rdkit.org/docs/index.html
\textsuperscript{2}https://tdcommons.ai/
Figure 6: Inter-correlation heatmaps for studied molecular properties in QM9, ZINC and ChEMBL datasets.

Figure 7: Property distributions of 7 randomly selected molecular properties on QM9, ZINC and ChEMBL datasets.

C Latent Space Evaluation

To evaluate the quality of the learned latent space, we utilize three disentanglement evaluation metrics, disentanglement, completeness and informativeness [58]. To be specific, disentanglement measures the degree to which each latent dimension controls at most one molecular property, completeness measures the degree to which each molecular property is governed by at most one latent dimension, and informativeness measures the prediction accuracy of molecular properties given the latent representation. From Table 4, we find MoFlow learns a better and more disentangled latent space than CGVAE and HierVAE. One possible reason is that MoFlow (369) has a larger latent space than CGVAE (100) and HierVAE (32) since Flow restricts the latent size to be equal to the input size. Similarly, CGVAE ranks the second likely because its latent space size is larger than HierVAE.

D Molecule Manipulation Experiments

D.1 Molecule Generation Evaluation

We evaluate the Validity, Novelty and Uniqueness of the generated molecules as proposed in Kusner et al. [59] in Table 5. We can observe that ChemSpace not only improves the success rate from the baseline methods, but also in general improves the novelty and uniqueness during manipulation. Besides, in Fig. 8, we also report the SSR curves of molecule manipulations over three models on QM9 and ZINC datasets with multiple manipulation ranges (distance in the latent space), $[-1, 1]$, $[-5, 5]$, $[-10, 10]$ and $[-20, 20]$. From the figure, we can observe that the trends in each of the curves remain still when the manipulation range changes. In general, either too large or too small range is not desired, we set it as a hyper-parameter and we observe that $[-1, 1]$ is a reasonably good default value. More experiments on molecule manipulation can be found in Appendix D.
Table 4: Quantitative Evaluation of Disentanglement on Latent Space.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Models</th>
<th>Disentanglement</th>
<th>Completeness</th>
<th>Informativeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>QM9</td>
<td>MoFlow</td>
<td>0.24</td>
<td>0.57</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>HierVAE</td>
<td>0.13</td>
<td>0.27</td>
<td>0.75</td>
</tr>
<tr>
<td>ZINC</td>
<td>MoFlow</td>
<td>0.40</td>
<td>0.62</td>
<td>0.87</td>
</tr>
<tr>
<td>ChEMBL</td>
<td>HierVAE</td>
<td>0.14</td>
<td>0.41</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Figure 8: Molecule manipulation performance (average) with various manipulation ranges with three models on QM9 (top) and ZINC (bottom) datasets.

D.2 Molecule Manipulation Evaluation

In this section, we report detailed results for all manipulation ranges \([-1, 1], [-5, 5], [-10, 10], [-20, 20]\) in terms of success rate and strict success rate in Table 6. Additionally, we visualize the SSR curves of molecule manipulations over three models on QM9 and ZINC in Fig. 9 and SR/SSR.

Table 5: Quantitative Evaluation of Latent Manipulation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Models</th>
<th>Validity (%)</th>
<th>Novelty (%)</th>
<th>Uniqueness (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>QM9</td>
<td>MoFlow</td>
<td>100.00</td>
<td>98.23</td>
<td>98.27</td>
</tr>
<tr>
<td></td>
<td>MoFlow-R</td>
<td>91.60</td>
<td>91.60</td>
<td>8.06</td>
</tr>
<tr>
<td></td>
<td>MoFlow-L</td>
<td>40.75</td>
<td>40.75</td>
<td>9.32</td>
</tr>
<tr>
<td></td>
<td>MoFlow-C</td>
<td>91.63</td>
<td>88.71</td>
<td>24.23</td>
</tr>
<tr>
<td>QM9</td>
<td>HierVAE</td>
<td>100.00</td>
<td>79.39</td>
<td>95.14</td>
</tr>
<tr>
<td></td>
<td>HierVAE-R</td>
<td>100.00</td>
<td>84.53</td>
<td>28.91</td>
</tr>
<tr>
<td></td>
<td>HierVAE-L</td>
<td>100.00</td>
<td>84.05</td>
<td>27.26</td>
</tr>
<tr>
<td></td>
<td>HierVAE-C</td>
<td>100.00</td>
<td>79.66</td>
<td>34.81</td>
</tr>
<tr>
<td>ZINC</td>
<td>MoFlow</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>MoFlow-R</td>
<td>69.98</td>
<td>69.98</td>
<td>29.04</td>
</tr>
<tr>
<td></td>
<td>MoFlow-L</td>
<td>43.36</td>
<td>43.36</td>
<td>24.87</td>
</tr>
<tr>
<td></td>
<td>MoFlow-C</td>
<td>71.26</td>
<td>71.26</td>
<td>15.82</td>
</tr>
<tr>
<td>ChEMBL</td>
<td>HierVAE</td>
<td>100.00</td>
<td>94.03</td>
<td>99.45</td>
</tr>
<tr>
<td></td>
<td>HierVAE-R</td>
<td>100.00</td>
<td>84.53</td>
<td>28.91</td>
</tr>
<tr>
<td></td>
<td>HierVAE-L</td>
<td>100.00</td>
<td>93.00</td>
<td>55.09</td>
</tr>
<tr>
<td></td>
<td>HierVAE-C</td>
<td>100.00</td>
<td>94.24</td>
<td>56.58</td>
</tr>
</tbody>
</table>
Table 6: Quantitative Evaluation of Molecule Manipulation over a variety of molecular properties (numbers reported are soft success rate-L / soft success rate-G in %, -R denotes model with random manipulation, -L denotes model with largest range manipulation, -O denotes optimization-based manipulation, -C denotes model with ChemSpacE. The best performances are bold.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Models</th>
<th>Avg. QED</th>
<th>LogP</th>
<th>SA</th>
<th>DRD2</th>
<th>JNK3</th>
<th>GSK3B</th>
<th>MolWt</th>
</tr>
</thead>
<tbody>
<tr>
<td>QM9</td>
<td>MoFlow-R</td>
<td>27.21 / 32.31</td>
<td>1.50 / 2.00</td>
<td>0.00 / 3.00</td>
<td>1.00 / 3.00</td>
<td>0.00 / 46.00</td>
<td>4.00 / 4.00</td>
<td>0.00 / 15.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-L</td>
<td>29.28 / 35.20</td>
<td>3.00 / 8.00</td>
<td>1.00 / 7.00</td>
<td>1.00 / 2.00</td>
<td>0.50 / 42.50</td>
<td>6.00 / 6.00</td>
<td>0.50 / 7.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-O</td>
<td>N/A</td>
<td>4.50/6.50</td>
<td>6.50/8.50</td>
<td>8.50/13.00</td>
<td>3.00/15.00</td>
<td>10.50/10.50</td>
<td>10.50/17.50</td>
</tr>
<tr>
<td></td>
<td>MoFlow-C</td>
<td>53.97 / 61.56</td>
<td>16.00 / 28.00</td>
<td>13.50 / 28.00</td>
<td>17.50 / 39.50</td>
<td>17.50 / 72.50</td>
<td>58.50 / 58.50</td>
<td>21.50 / 49.00</td>
</tr>
<tr>
<td>ZINC</td>
<td>HierVAE-R</td>
<td>2.62 / 26.06</td>
<td>1.00 / 1.00</td>
<td>1.50 / 1.50</td>
<td>0.50 / 0.50</td>
<td>0.50 / 1.50</td>
<td>1.00 / 5.50</td>
<td>1.00 / 3.00</td>
</tr>
<tr>
<td></td>
<td>HierVAE-L</td>
<td>3.25 / 27.33</td>
<td>0.50 / 1.00</td>
<td>0.00 / 1.50</td>
<td>0.50 / 4.00</td>
<td>2.00 / 6.50</td>
<td>0.00 / 2.50</td>
<td>0.50 / 1.50</td>
</tr>
<tr>
<td></td>
<td>HierVAE-O</td>
<td>N/A</td>
<td>1.50/2.00</td>
<td>10.50/15.50</td>
<td>1.00/2.50</td>
<td>2.50/5.50</td>
<td>18.00/21.50</td>
<td>23.50/28.50</td>
</tr>
<tr>
<td></td>
<td>HierVAE-C</td>
<td>46.72 / 61.49</td>
<td>27.00 / 35.00</td>
<td>32.00 / 44.00</td>
<td>35.00 / 42.00</td>
<td>41.50 / 48.50</td>
<td>51.50 / 60.00</td>
<td>30.00 / 33.50</td>
</tr>
</tbody>
</table>

The curves of molecule manipulation with HierVAE on ChEMBL datasets in Fig. 10. The manipulation visualization of CGVAE on QED, MolWt and LogP is provided in Fig. ??.

E  ChemSpacE Demo

As shown in Fig. 11(right), we design an interactive real-time system for molecule manipulation, where the user can click random to randomly sample a molecule and freely select which model to interpret, which property to interpret, and tuning the slide bar manipulates the molecule accordingly in real-time. The demo video is anonymously provided at https://drive.google.com/drive/folders/1N036p_50fvGZybPj3vW10NXXVepimSR?usp=sharing.

18
Figure 10: Molecule manipulation performance with various manipulation ranges with HierVAE on ChEMBL dataset (left SR, right SSR) (better seen in color).

Figure 11: Optimizing molecular properties with optimization-based method.