Interpretable Molecule Generation via Disentanglement Learning

Yuanqi Du
Dept of Computer Science
George Mason University
ydu6@gmu.edu

Xiaojie Guo
Dept of Information Sciences and Technology
George Mason University
xguo7@gmu.edu

Amarda Shehu
Dept of Computer Science
George Mason University
ashehu@gmu.edu

Liang Zhao*
Dept of Information Sciences and Technology
George Mason University
lzhao9@gmu.edu

ABSTRACT

Designing molecules with specific structural and functional properties (e.g., drug-likeness and water solubility) is central to advancing drug discovery and material science, but it poses outstanding challenges both in wet and dry laboratories. The search space is vast and rugged. Recent advances in deep generative models are motivating new computational approaches building over deep learning to tackle the molecular space. Despite rapid advancements, state-of-the-art deep generative models for molecule generation have many limitations, including lack of interpretability. In this paper we address this limitation by proposing a generic framework for interpretable molecule generation based on novel disentangled deep graph generative models with property control. Specifically, we propose a disentanglement enhancement strategy for graphs. We also propose new deep neural architecture to achieve the above learning objective for inference and generation for variable-size graphs efficiently. Extensive experimental evaluation demonstrates the superiority of our approach in various critical aspects, such as accuracy, novelty, and disentanglement.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Applied computing** → **Molecular structural biology**; **Bioinformatics**.

KEYWORDS

Graph neural network, molecule generation, disentangled representation learning.

ACM Reference Format:
Yuanqi Du, Xiaojie Guo, Amarda Shehu, and Liang Zhao. 2020. Interpretable Molecule Generation via Disentanglement Learning. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '20), September 21–24, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3388440.3414709

*Corresponding Author

BCB '20, September 21–24, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3414709

1 INTRODUCTION

Our ability to design molecules with specific structural and functional properties is central to advancing drug discovery and material science [39]. As the current COVID-19 pandemic is acutely demonstrating, it is hard to design novel drugs in an expeditious manner. In particular, decades of research in medicinal chemistry show that finding novel drugs remains an outstanding challenge [33]. The search space is vast, with some studies estimating 10^{60} drug-like molecules being synthetically-accessible [29]. The space is also highly rugged; small perturbations in the chemical structure may result in great changes in desired properties. While high-throughput technologies have improved significantly, the space is too large to address the de-novo design of molecules exclusively in the wet laboratory. Computational approaches provide a promising complementary approach.

While for many years computational screening was primarily dominated by similarity search [36], recent advances in deep generative models are showing promise in finally tackling de-novo molecule design. The first efforts addressed the problem as a string generation one [8, 19]. These works leverage the "molecular-input line-entry system" (SMILES) representation, which is a linear string representation of molecules and their active structures [38]. SMILES is a formal grammar that describes molecules with an alphabet of characters; for instance, 'c' and 'C' denote aromatic and aliphatic carbon atoms, 'O' denotes the oxygen atom, '-' denotes single bonds, '=' denotes double bonds, and so on. While designed to be human-readable, the SMILES representation is not designed to capture molecular similarity, which prevents generative models, such as variational auto-encoders (VAE), from learning smooth molecular embeddings. More importantly, essential chemical properties, such as molecular validity, cannot be expressed and preserved by the SMILES representation.

Recent advances in deep generative models that can generate graphs have opened a new research direction for de-novo molecular design. Specifically, these models leverage more expressive representations of molecules via the concept of graphs. The atoms can be represented as nodes, and the bonds that connect them in a molecule constitute edges connecting nodes in the graph. Graph-generative models hold much promise in generating credible

molecules [15, 27, 35]. It is worth noting that the latter is captured rigorously, by subjecting a generated molecule to the sanitization checks in RDKit [20].

Current state-of-the-art deep generative models for molecule generation consist of two complementary subtasks: (1) the encoding, which refers to learning to represent molecules in a continuous manner that facilitates the preservation or optimization of their properties; (2) the decoding, which refers to learning to map an optimized continuous representation back into a reconstructed or novel molecular graph. Despite promising results, these current models have some limitation.

The learned latent representations are not disentangled and so do not expose how the underlying factors control molecular properties. Unpacking this black box of the molecule generation process via interpretation of the latent representations is critical yet unexplored by existing methods.

In this paper we address the above limitation by proposing a Disentangled Molecule VAE, to which we refer as D-MolVAE from now on. D-MolVAE is a novel deep generative framework that makes several contributions. First, the framework is disentangled. Second, the framework accommodates variable-size molecules. Finally, extensive experiments are carried out, demonstrating advantages over state-of-the-art methods.

This paper is organized as follows. First, we provide a brief summary of related works in deep generative models for the problem of molecule generation in Section 2. The framework is then described in detail in Section 3. Evaluation is presented in Section 4. The paper concludes with a summary and future research directions in Section 5.

2 RELATED WORK

2.1 Deep Graph-Generative Models

A significant number of deep generative model-based methods are proposed for learning and sampling from structured data (i.e., graphs). Most existing models are based on the VAE framework [32, 35] or generative adversarial networks (GANs) [2, 11], and others [22, 40]. For instance, GraphRNN [40] builds an autoregressive generative model based on a generative recurrent neural network (RNN) by representing the graph as a sequence and generating nodes. In contrast, GraphVAE [35] represents each graph in terms of its adjacent matrix and feature vectors of nodes. A VAE model is then utilized to learn the distribution of the graphs conditioned on a latent representation at the graph level. Other works [9, 17] encode the nodes of each graph into node-level embeddings and predict the links between each pair of nodes to generate a graph.

2.2 Deep Learning for Molecule Generation

Early deep learning-based works in [8, 34] built generative models of SMILES strings with recurrent decoders. However, these models could generate invalid SMILES not representing any molecules. To this end, later works [4, 19] improved the decoder with syntactic and semantic constraints by context-free and attribute grammars; these, however, also do not fully capture chemical validity. Other methods based on active learning [14] and reinforcement learning [10] guide the model to generate valid SMILES through additional training signals.

Very recently, recent advances in graph-generative models have opened up a new avenue for molecule generation. For example, work in [35] generates molecular graphs by predicting their adjacency matrices. Work in [22] generates molecules through a constrained graph generative model that enforces validity by generating the molecule atom by atom.

2.3 Disentangled Representation Learning

Disentangled representation learning based on VAE has gained considerable attention, recently, particularly in the domain of image representation learning [1, 3, 12, 16]. The goal is to learn representations that separate out the underlying explanatory factors responsible for formalizing the data. Such representations have been shown to enhance generalizability as well as improve robustness against adversarial attack [1].

Disentangled representations are inherently more interpretable, and can thus potentially facilitate debugging and auditing [5]. A number of approaches have been prompted to modify the VAE objective by adding, removing, or altering the weight of individual terms to improve the disentanglement properties of the latent representations [1, 3, 7, 16, 18, 25, 41]. However, the best way of learning representations that disentangle the latent factors behind a graph remains largely unexplored. Though few work [26] are proposed for interpreting the graph representations, they are not focus on graph generation task. In addition, utilizing disentanglement learning for molecule generation is critical yet seldom explored. We investigate this direction of research in this paper to enhance the interpretability of the process of molecule generation.

3 METHODS

Before relating details of the proposed D-MolVAE framework, we formalize the prolem.

3.1 Problem Formulation

Define a molecule as a graph of atoms $G = (\mathcal{V}, \mathcal{E}, E, F)$, where $\mathcal{V}$ is the set of N nodes (i.e., atoms) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of M edges (i.e. bonds that connect pairs of atoms). $e_{i,j} \in \mathcal{E}$ is an edge connecting nodes $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$. $E \in \mathbb{R}^{N \times N \times K}$ refers to the edge type tensor (i.e. bond type), where $E_{i,j} \in \mathbb{R}^{1 \times K}$ is an one-hot vector encoding the type of edge $e_{i,j}$. K is the total number of the edge types. $F \in \mathbb{R}^{N \times K'}$ refers to a node (i.e., atom) feature matrix, where $F_i \in \mathbb{R}^{1 \times K'}$ is the one-hot encoding vector denoting the type of atom v_i and K' is the total number of the atom types.

Our goal is to develop a deep generative model that can learn the joint distribution of the molecule G and a set of generative disentangled latent variables Z to discover the factors (e.g. molecule properties) in formalizing a molecule, such that the observed molecule graph G can be generated as $p(G|Z)$. By *disentanglement*, we mean the individual variables inside Z are independent from each other.

The proposed problem goes beyond the existing molecule generation problem with enhanced interpretability over the generation process. Despite the significance of this problem, to achieve it, however, is highly difficult due to several major technical challenges: (1) Difficulty in handling the dilemma between disentanglement and reconstruction quality for the molecule graph generation. (2)

Inefficiency in encoding and decoding molecules with different sizes. It is always challenging to learn the deep generative models that can generate graphs with variable sizes.

In order to solve the proposed problem and address the above challenges, we now propose the objective function of D-MolVAE which can enforce the disentanglement of the learned representations. First, a VAE generative objective with the disentanglement constraint is derived, which handles the first challenge described in Section 3.1. Furthermore, to handle the second challenge, a new disentangled graph VAE is proposed based on our variable-size edge-to-edge and edge-to-node convolution operators, detailed in Section 3.3.

3.2 Overall Objective

3.2.1 Disentangled Deep Generative Models for Molecule Graphs. Inspired by the disentanglement representation learning in the image domain [12], a suitable objective in learning $p(G|Z)$ is to maximize the marginal (log-)likelihood of the observed graph G in expectation over the whole distribution of latent variables set $Z \in \mathbb{R}^{N\times L}$ as $\max_\theta \mathbb{E}_{p_\theta(Z)}[p_\theta(G|Z)]$ where θ is the parameter of this distribution. Here, N is the number of nodes and L is the dimension of distinct latent factors.

For a given molecule graph, we describe the inferred posterior configurations of the latent variables Z using a probability distribution $q_\phi(Z|G)$. Our aim is to ensure that the inferred latent variables Z from $q_\phi(Z|G)$ capture all the generative factors in a disentangled manner. To encourage this disentanglement characteristic in the inferred $q_\phi(Z|G)$, we can introduce a constraint by trying to match it to a prior $p(Z)$ that both controls the capacity of the latent information bottleneck, and embodies the statistical independence mentioned above. This can be achieved if we set the prior to be an isotropic unit Gaussian, i.e. $p(Z) = \mathcal{N}(\mathbf{0}, I)$, leading to the constrained optimization problem in Eq. 1:

$$\max_{\theta,\phi} \ \mathbb{E}_{G\sim\mathcal{D}}[\mathbb{E}_{q_\phi(Z|G)} log p_\theta(G|Z)] \qquad (1)$$
$$s.t. \ D_{KL}(q_\phi(Z|G)||p(Z)) \leq \epsilon.$$

where ϵ specifies the strength of the applied constraint; $\mathcal{D}$ refers to the observed set of molecules and $D_{KL}(\cdot)$ denotes the Kullback–Leibler divergence (KLD) between two distributions.

3.2.2 Realization of Disentanglement Constraint. The constraint in Eq. 1 is intractable and hard to be optimized through stochastic gradient descent. We propose to effectively enforce the constraint by transferring it to a regularization term. We propose the *Disentanglement Inferred Prior* term.

In dealing with the constraint, since there is no explicit upper bound on the KLD between $q_\phi(Z|G)$ and $p(Z)$, the reconstruction error and KLD can be jointly minimized in the objective as:

$$\max_\theta \mathbb{E}_{p_\theta(Z)}[\log p_\theta(G|Z)] - \lambda D_{KL}(q_\phi(Z|G)||p(Z)) \qquad (2)$$
$$s.t. \forall x_1 \leq x_2 : F_j(x_1) \leq F_j(x_2), \ \ x_1, x_2 \sim q_\phi(Z^{(j)}|G), \ Z^{(j)} \subseteq Z$$

However, as proved by Esmaeili et al. [7], the enforcement on the disentanglement (i.e., the second term) will influence the optimization of reconstruction loss (i.e.the first term), as mentioned in the first challenge in Section 3.1. Thus, we first decompose the second term in the objective as:

$$\begin{aligned} &- D_{KL}(q_\phi(Z|G)||p(Z)) \\ &= - \mathbb{E}_{q_\phi(Z,G)} log \frac{q_\phi(Z|G)}{q_\phi(Z)} - D_{KL}(q_\phi(Z)||p(Z)), \end{aligned} \qquad (3)$$

where the first term in the second row minimizes the mutual information $I(Z, G)$ in the inference model, while maximizing the second term (i.e., inferred priors) enforces the distance between $q_\phi(Z)$ and $p(Z)$.

Since the first term actually represents the mutual information between the latent Z and the graphs G, which will lead to poor reconstructions when enforcing disentanglement as mentioned in the first challenge. Thus, to solve the trade-off problems between the disentanglement of Z and G, we discard it and maximize its lower bound instead as:

$$\max_\theta \mathbb{E}_{p_\theta(Z)}[\log p_\theta(G|Z)] - \lambda D_{KL}(q_\phi(Z)||p(Z)) \qquad (4)$$
$$s.t. \forall x_1 \leq x_2 : F_j(x_1) \leq F_j(x_2), \ \ x_1, x_2 \sim q_\phi(Z^{(j)}|G), \ Z^{(j)} \subseteq Z$$

Considering that $Z = \{z_1, ..., z_N\}$, then the *Disentanglement Inferred Prior* term can be further written as:

$$\begin{aligned} -D_{KL}(q_\phi(Z)||p(Z)) &= \mathbb{E}_{q_\phi(Z,G)}(log \frac{p(Z)}{q_\phi(Z)}) \\ &= \mathbb{E}_{q_\phi(Z,G)}(log \frac{\prod_i^N p(Z_i)}{\prod_i^N q_\phi(Z_i)}) \\ &= - \sum_i^N D_{KL}(q_\phi(Z_i)||p(Z_i)). \end{aligned} \qquad (5)$$

3.3 Architecture of D-MolVAE

The construction of the overall model extends the conventional VAE consisting of an encoder and a decoder, where the encoder learns the mean and standard deviation of the latent representation of the input and the decoder decodes the sampled latent representation to reconstruct the input. The graph encoder is used to model the prior distributions $q_\phi(Z|G)$ by generating the mean μ and standard variation σ of this learned distribution.

The graph decoder is utilized to model $p_\theta(G|Z)$, of which the output is parameterizing the learned distribution. The latent representation is then sampled by the inferred mean μ and standard derivation σ of the learnt distribution. The details of each components are described as follows. The encoder and decoder of the proposed model is based on the work proposed by Liu et al. [22].

3.3.1 Molecule Encoder. To model the prior distributions $q_\phi(Z|G)$ expressed in the objective, an encoder is constructed based on a graph neural network (GNN). The GNN embeds each node in an input graph G into L-dimension latent space following distribution $q_\phi(Z|G)$ parameterized by mean μ_i and standard deviation vectors σ_i for each node v_i, which is the output of the GNN. As a result, by sampling from the modelled distribution, $Z = \{Z_1, ..., Z_N\}$ are obtained variables containing the node representation vectors for all the nodes.

3.3.2 Molecule Decoder. The molecule decoder models the distribution $p_\theta(G|Z)$ by generating the molecule graph G conditioning on the latent representation variables Z that are sampled from the learned distribution in the encoder. The process proceeds in an auto-regressive style. In each step a focus node is chosen to be visited, and then the edges are generated related to this focus node. The nodes are ordered by using the breadth-first traversal.

The molecule decoder mainly contains three steps, namely *node initialization*, *node update* and *edge selection and labelling*.

Node Initialization. We first define N as an upper bound on the number of nodes in the final generated graph. An initial state $h_i^{(t=0)}$ is assigned with each node v_i in a set of initially unconnected nodes. Specifically, $h_i^{(t=0)}$ is the concatenation as $[Z_i, \tau_i]$, where τ_i is an one-hot vector indicating the atom type. τ_i is derived from Z_i by sampling from the softmax output of a learned mapping $\tau_i \sim f(Z_i)$ where $f(\cdot)$ is a multiple layer perception (MLP). From these node-level states, we can calculate global representations $H(t)$, which is the average representation of nodes in the connected component at generation step t. In addition to N working nodes, a special "stop node" is also initialized to a learned representation h_{end} for managing algorithm termination detailed as below.

Edge Selection and Labeling. At each step t, a focus node v_i is picked from the queue of nodes. Then an edges $e_{i,j}$ is selected from node v_i to node v_j with label $E_{i,j}$. Specifically, for each non-focus node v_j, we construct a feature vector $\eta_{i,j}^{(t)} = [h_i^{(t)}, h_j^{(t)}, d_{i,j}, H(t), H(0)]$, where $d_{i,j}$ is the graph distance (i.e. path) between two nodes v_i, v_j. We use these representations to produce a distribution over candidate edges:

$$p(e_{i,j}, E_{i,j}|\eta_{i,j}^{(t)}) = p(E_{i,j}|\eta_{i,j}^{(t)}, e_{i,j}) \cdot p(e_{i,j}|\eta_{i,j}^{(t)}) \tag{6}$$

The parameters of the distribution are calculated as softmax outputs from neural networks, that is, $f_{\text{node}}(\cdot)$ which determines the target node for an edge, and $f_{\text{bond}}(\cdot)$ which determines the type of the edge:

$$p(e_{i,j}|\eta_{i,j}^{(t)}) = \frac{M_{i,j}^{(t)} \exp(f_{\text{node}}(\eta_{i,j}^{(t)}))}{\sum_k^N M_{i,k}^{(t)} \exp(f_{\text{node}}(\eta_{i,k}^{(t)}))}, \tag{7}$$

$$p(E_{i,j} = l|\eta_{i,j}^{(t)}) = \frac{m_{i,j,l}^{(t)} \exp([f_{\text{bond}}(\eta_{i,j}^{(t)})]_l)}{\sum_u^L m_{i,j,u}^{(t)} \exp([f_{\text{bond}}(\eta_{i,j}^{(t)})]_u)}, \tag{8}$$

where l refers to one type of the edge and $[f_{\text{bond}}(\eta_{i,j}^{(t)})]_u$ refers to the u-th entry in the output of function $f_{\text{bond}}(\cdot)$. $M_{i,j}^{(t)}$ and $m_{i,j,l}^{(t)}$ are binary masks that forbid edges that violate constraints on constructing syntactically valid molecules, the construction of which for the molecule generation is introduced in the following sections. New edges are sampled one by one from the above learned distributions. Any nodes that are connected to the graph for the first time during this edge selection are added to the node queue.

Node Update. Whenever we obtain a new graph $G^{(t+1)}$ at step t, the previous node states $h_i^{(t)}$ is discarded and a new node representations $h_i^{(t+1)}$ for each node is calculated by taking their (possibly changed) neighborhood into account. To this end, a standard gated graph neural network (GGNN) is utilized through S steps, which is defined as a recurrent operation over messages $r_i^{(s)}$ as:

$$r_i^{(s+1)} = GRU[r_i^{(s)}, \sum_{j \in \mathcal{V}} \text{MLP}(r_j^{(s)})] \tag{9}$$

$$h_i^{(t+1)} = r_i^{(S)}, \tag{10}$$

where $r_i^{(0)} = h_i^{(0)}$ and the sum runs over all edges in the current graph. It worth noting that since $h_i^{(t+1)}$ is computed from $h_i^{(0)}$ rather than $h_i^{(t)}$, the representation $h_i^{(t+1)}$ is independent of the generation history of $G^{(t+1)}$.

Termination. In the edge generation process of each node, the edges to a node v_i is kept added until an edge to the stop node is selected. Then we move the focus from the node v_i, and regard v_i as a "closed" node. The next focus node is then selected from the focus queue. In this way, a single connected component is grown in a breadth-first manner. The node and edge generations continue until the queue of nodes is empty. It is worth noting that there may be some unconnected nodes left at the end, which will be discarded from the final graphs.

Valency Masking. To construct syntactically-valid molecules, we additionally utilize a valency mask. Namely, the valency of an atom indicates the number of bonds that an atom can make in a stable molecule. In the molecule graph, each atom type has a fixed valency. For example, node type "H" (a hydrogen atom) has a valency of 1, and node type "O" (an oxygen atom) has a valency of 2. Throughout the generation process, two types of masks $M_{i,j}^{(t)}$ and $m_{i,j,l}^{(t)}$ are used to guarantee that the bonds b_i of each atom never exceeds its valency b_i^*. After the generation is finished, if $b_i < b_i^*$, $b_i^* - b_i$ hydrogen atoms are added to be linked to atom v_i. As a result, the generated molecules are always syntactically-valid. More specifically, $M_{i,j}^{(t)}$ also handles avoidance of edge duplication and self loops, and is defined as:

$$\begin{aligned} M_{i,j}^{(t)} = \mathbb{I}(b_i < b_i^*) \times \mathbb{I}(b_j < b_j^*) \times \mathbb{I}(e_{i,j} \text{ not exist}) \\ \times \mathbb{I}(i \neq j) \times \mathbb{I}(v_i \text{ is focus}) \end{aligned} \tag{11}$$

where $\mathbb{I}(\cdot)$ is an indicator function, and as a special case, connections to the stop node are always unmasked. Further, when selecting the label for a chosen edge, we must again avoid violating the valency constraint, so we define $m_{i,j,l}^{(t)} = M_{i,j}^{(t)} \times \mathbb{I}(b_j^* - b_j < l)$, where l refer to the bond type and $l = 1, 2, 3$ indicates single, double and triple bond types respectively.

4 RESULTS

We evaluate our D-MolVAE model via metrics in qualitative and quantitative experiments in comparison with other related deep generative model for molecule generation [8, 19, 21, 23, 32, 35]. All experiments are conducted on a 64-bit machine with an NVIDIA GPU (GeForce RTX 2080ti, 1545MHz, 11GB GDDR6).

4.1 Models Utilized for Comparative Evaluation

We implement and evaluate the following 6 current, state-of-the-art deep generative frameworks for molecule generation:

- *CGVAE* [23]: This is a VAE model, in which both encoder and decoder are graph-structured and enforce a validity constraint.
- *GraphGMG* [21]: This is a deep auto-regressive graph model that generates the nodes of a graph sequentially.
- *SMILES-LSTM* [37]: This is an LSTM model that utilizes the SMILES representation.
- ChemVAE [8]: This is a generative model that converts discrete representations of molecules to and from a multidimensional continuous representation.
- GrammarVAE [19]: This is a VAE-based model that enforces syntactic and semantic constraints over SMILES strings via context free and attribute grammars.
- GraphVAE [35]: This is a generic deep generative model for graph generation.

We note that CGVAE [23] has a similar encoder and decoder as the proposed D-MolVAE model but does not contain the disentaglement we propose here. So, the comparison allows evaluating the added benefit of disentanglement.

4.2 Datasets

We consider two molecule datasets commonly used as benchmarks in molecule generation literature, QM9 [28, 31] and ZINC [13]. We split the training/validation set as follows. For QM9, we use 120k/20k as training/validation set. For ZINC, we use 60k/10k as training/validation set.

- The *QM9 Dataset* [28, 31] consists of ~134k stable small organic molecules with up to 9 heavy atoms (Carbon (C), Oxygen (O), Nitrogen (N) and Fluorine (F)).
- The *ZINC Dataset* [13] includes ~250k drug-like chemical compounds with an average of ~23 heavy atoms. The structures of the molecules in this dataset are more complex than those of the molecules in the QM9 dataset.

4.3 Metrics Employed for Evaluation

4.3.1 Quantitative Metrics to Evaluate the Quality of the Learned Distribution. The distribution of molecules learned by our model is evaluated via the following metrics.

Novelty: This metric measures the fraction of generated molecules that are not in the training dataset.

Uniqueness: This metric measures the fraction of generated molecules after and before removing duplicates.

Validity: This metric measures the fraction of generated molecules that are chemically valid.

4.3.2 Metrics to Compare the Learned Distribution and the Reference Distribution. We additionally utilize several other metrics to compare the training/reference dataset to the generated dataset. Specifically, a distribution of a variable of interest is computed from the training and the generated dataset, respectively, and the distributions are compared via two popular metrics in the Machine Learning (ML) community, the Maximum Mean Discrepancy (MMD) [40] and KLD [40]. When utilizing MMD, we focus on the following three variables that allow summarizing a graph (we recall that the molecules are represented as a graph): *node degree*, *clustering coefficient*, and *average orbit count*; the latter counts the number of 4-orbits in a graph. These three variables are routinely used to summarize distributions of graphs in graph-generative deep learning works [22, 40]. When utilizing KLD, we focus on benchmark molecular properties, such as cLogP and Drug-likeness [23].

4.3.3 Metrics to Evaluate Disentanglement. We employ four popular metrics to evaluate disentanglement.

β-M[12] measures disentanglement by examining the accuracy of a linear classifier that predicts the index of a fixed factor of variation.

F-M[16] addresses several issues by using a majority voting classifier on a different feature vector that represents a corner case in the β-M.

MOD [30] measures whether each latent variable depends on at most a factor describing the maximum variation using their mutual information.

DCI [6] computes the entropy of the distribution obtained by normalizing the importance of each dimension of the learned representation for predicting the value of a factor of variation. All implementation details are as in [24].

4.3.4 Qualitative Evaluation of Generated Molecules. In addition to the quantitative evaluations above, we visualize generated molecules from the proposed model and the model utilized for comparison. The distributions of some graph statistics in the generated molecules and real molecules are also drawn and thus compared visually.

4.3.5 Qualitative Evaluation for Disentanglement. We demonstrate qualitatively that our proposed model consistently discover more latent factors (molecular properties) and disentangle them in a cleaner fashion. By jointly changing the value of one variable in each of the node latent representation continuously and fixing the remaining variables, we can visualize the corresponding variation of molecular properties in the generated graphs. These properties are selected due to their low correlation, which are ideal for the disentanglement experiment setting that requires the independent semantic factors.

4.4 Experimental Results

4.4.1 Evaluating the Quality of Generated Molecules. Table 1 shows the performance of six state-of-the-art model and our proposed model, D-MolVAE, in terms of *novelty*, *uniqueness*, and *validity*. We train those model on our two datasets and sample 30k molecules from the trained model (In terms of the GraphGMG model [21], we obtained 20k generated molecules from the GraphGMG authors). As Table 1 shows, D-MolVAE achieves superior performance in terms of all the three metrics over the other 6 model.

Table 1 allows making several observations. First, our model, D-MolVAE, achieves 100% validity; that is, 100% of generated molecules are chemically-valid. This is due to the masking mechanism in our model which can guarantee the validity of a generated molecule. In contrast, ChemVAE achieves the lowest performance, with 10% only valid molecules on the QM9 dataset and 17% valid molecules on the ZINC dataset. Second, D-MolVAE also generates up to 99.99%

Table 1: Novelty, uniqueness, and validity are measured on the molecule datasets generated by the various model under comparison. The highest value achieved on a metric is highlighed in bold font.

Dataset	Metric	D-MolVAE	CGVAE	GraphGMG	LSTM	ChemVAE	GrammarVAE	GraphVAE
QM9	% Validity	**100.00**	**100.00**	-	94.78	10.00	30.00	61.00
	% Novelty	**97.36**	96.33	-	82.98	90.00	95.44	85.00
	% Unique	97.80	**98.03**	-	96.94	67.50	9.30	40.90
ZINC	% Validity	**100.00**	**100.00**	89.20	96.80	17.00	31.00	14.00
	% Novelty	99.99	**100.00**	89.10	**100.00**	98.00	**100.00**	**100.00**
	% Unique	99.88	99.95	99.41	**99.97**	30.98	10.76	31.60

novel molecules, which is higher than other methods. The higher novelty is due to the disentangled representations which can fully explore molecular patterns. In comparison to CGVAE, which shares a similar architecture but without disentanglement enforcement, the proposed model have equal or better performance in novelty. This demonstrates that adding the disentanglement regularization does not influence the reconstruction error and so does not sacrifice the quality of generated molecules. Finally, our model and CGVAE have the highest performance, over 99% in terms of uniqueness. ChemVAE, GrammarVAE, and GraphVAE have the lowest performance.

4.4.2 Comparing the Learned Distribution to the Reference Distribution. Given the results shown above, we now focus on the comparison of our model with the top-performing one among the comparison model, namely CGVAE. As described above, we measure the distance between the generated and the training datasets in terms of molecular properties and graph statistics, as shown in Table 2. MMD is used when comparing distributions of graph statistics, and KLD is used when comparing distributions of molecular properties.

Table 2: Comparison of training and generated distributions of graph properties via MMD and KLD. (CC refers to the Clustering Coefficient.

Dataset	Metric	D-MolVAE	CGVAE
QM9	MMD(Degree)	0.0838	0.0167
	MMD(CC)	0.0175	0.0097
	MMD(Orbit)	0.0079	0.0018
	KLD(cLogP)	0.35	0.08
	KLD(cLogS)	0.18	0.06
	KLD(Drug-like)	0.18	0.07
	KLD(Rel PSA)	0.18	0.04
	KLD(PSA)	0.30	0.03
ZINC	MMD(Degree)	0.0034	0.0023
	MMD(CC))	0.0005	0.0013
	MMD(Orbit)	0.0001	0.0005
	KLD(cLogP)	0.67	0.67
	KLD(cLogS)	0.74	0.74
	KLD(Drug-like)	1.29	1.29
	KLD(Rel PSA)	0.79	0.78
	KLD(PSA)	0.59	0.56

Table 2 shows the difference between the generated molecules and those in training set in terms of various molecular properties. The smaller the value is, the more similar the generated properties is to those in the training set. As can be seen in this table, both our model and CGVAE can reasonably preserve some patterns of the distribution of the molecule properties in the training set. Comparing with CGVAE, our method preserves more on the ZINC dataset while less on QM9 dataset. And this table generally shows that both our method and CGVAE have done a good job in balancing the information preservation and novelty of the generated molecules. Also notice that in addition to this, our model has additional functionality including the enhancement of the disentanglement and capability in controlling properties, which goes beyond what CGVAE can achieve. To further see this, please refer to Table 3 and Section 4.4.5.

4.4.3 Visualization of Generated Molecules. Figure 1 shows the two-dimensional structure of some molecules generated by CGVAE and our model, in comparison to the training data.

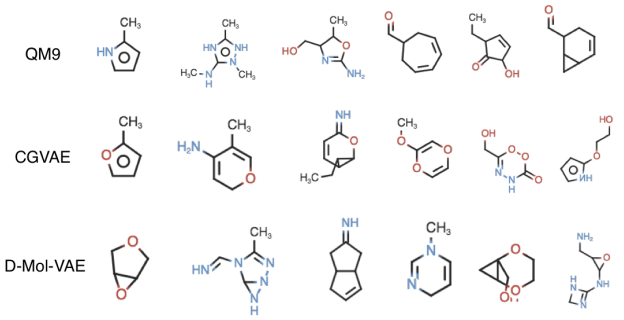

Figure 1: Molecules from QM9 dataset are shown in the top row. The next row shows molecules generated by CGVAE. The last row shows molecules generated by D-MolVAE.

We relate the entire distribution of generated molecules in Figure 2 in terms of the molecular properties cLogP and Drug-likeness. These distributions are superimposed over the distributions of the corresponding property over the training dataset for comparison. Figure 2 shows that our D-MolVAE model captures the distributions of the molecular properties in the training dataset.

4.4.4 Quantitative Evaluation of Disentanglement Learning. Table 3 illustrates the evaluation of our model's disentanglement score via the β-M, F-M, MOD, and DCI metrics. Table 3 shows that our

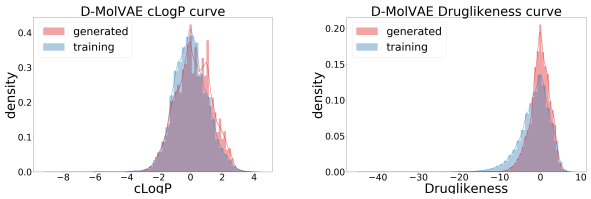

Figure 2: Distribution of the molecular properties CLogP and Drug-likeness(from top to down) in the generated versus the training dataset for our model, D-MolVAE

model achieve the best overall disentanglement scores. Specifically, in terms of the QM9 dataset with smaller molecules, D-MolVAE achieves an F-M score of 61.2%, whereas CGVAE achieves only 57.0%. Both models achieve comparable MOD scores, with D-MolVAE achieving the the highest. All models achieve a $\beta - M$ of 100%. CGVAE outperforms our model in terms of the DCI score. Interestingly, our model outperform the baseline CGVAE model on almost all the four metrics on the ZINC dataset with larger molecules. Altogether, these results show that the proposed D-MolVAE successfully learns the disentangled latent representations.

Table 3: Quantitative evaluation of disentanglement on QM9 and ZINC datasets.

Dataset	Model	β-M(%)	F-M(%)	DCI	Mod
QM9	CGVAE	**100**	57.0	**0.055**	0.239
	D-MolVAE	**100**	**61.2**	0.023	**0.261**
ZINC	CGVAE	**100**	48.0	**0.011**	0.195
	D-MolVAE	**100**	**52.4**	0.010	**0.197**

4.4.5 Qualitative Evaluation of Disentanglement Learning. In Figure 3 we show how generated molecules change when the value of latent variable z_0 traverses from -4 to 4. The cLogP scores assigned to the latent variable z_0 in the training process are shown at the bottom of each molecule. Similar performance can be seen in Figure 4.

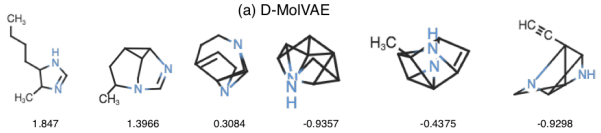

Figure 3: Molecules generated by D-MolVAE (top panel) and MD-MolVAE (bottom panel) as we increase the value of the latent variable z_0 by enforcing the cLogP property.

5 CONCLUSION

This paper proposes a disentangled VAE model for interpretable molecule generation. To learn the disentangled latent representation, we derive a new disentangled objective which consists of a

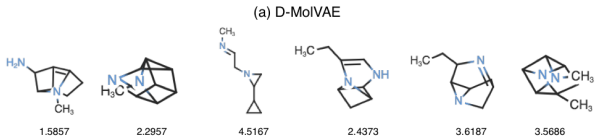

Figure 4: Molecules generated by D-MolVAE (top panel) and MD-MolVAE (bottom panel) as we increasethe value of the latent variable z_0 by enforcing the drug-likeness property.

reconstruction loss and a constraint requirement. To enable the optimization of the proposed objective, we further propose one regularization term, namely inferior priors term, in realizing the constraint. The proposed model is validated on two real-world molecule datasets for two tasks: one is molecule generation, and the other is disentangled representation learning. Both the quantitative and qualitative evaluation results show the promise of disentangled representation learning in interpreting molecule during the generation process.

We consider the proposed work an important first step that opens a line of work in explaining current deep-learning-based models designed for important problems in drug discovery, biology, material science, and other disciplines and domains. Here we highlight some directions for potential future work.

Beyond interpreting the molecule generation process, it is important to precisely control the properties of generated molecules. Given the specific values of several properties, one could decode back the latent variables into a molecule that preserves the exact required scores of properties.

We note that current methods are only concerned with global properties of molecules (or their graph representations), such as ClogP, drug-likeness, and others. Preserving local properties of an atom or a cluster of atoms (e.g., an aromatic hydrocarbon) has not been explored. Interpreting both local and global factors in formalizing a molecule can be helpful in designing novel drug structures and enhancing the understanding of the contribution of each element in the overall molecular property.

It is also interesting to generalize the proposed methods to additional applications in bioinformatics. We consider applying (and adapting) our techniques to other structured data, such as brain networks, protein structures, and more, and investigating the capabilities of interpretable graph-generative models in these domains.

6 ACKNOWLEDGMENTS

The work is supported by the National Science Foundation Grant No. 1942594, No. 1755850, No. 1907805 and a Jeffress Trust Awards Program in Interdisciplinary Research Award. This material is additionally based upon work by AS supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep Variational Information Bottleneck. In *5th International Conference on Learning*

Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

[2] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. 2018. NetGAN: Generating Graphs via Random Walks. In *International Conference on Machine Learning*. 609–618.

[3] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*. 2610–2620.

[4] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. 2018. Syntax-directed variational autoencoder for structured data. *arXiv preprint arXiv:1802.08786* (2018).

[5] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[6] Cian Eastwood and Christopher KI Williams. 2018. A framework for the quantitative evaluation of disentangled representations. (2018).

[7] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem van de Meent. 2019. Structured Disentangled Representations. *Proceedings of Machine Learning Research* 89 (2019).

[8] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* 4, 2 (2018), 268–276.

[9] Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative Generative Modeling of Graphs. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. 2434–2444.

[10] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. 2017. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843* (2017).

[11] Xiaojie Guo, Lingfei Wu, and Liang Zhao. 2018. Deep graph translation. *arXiv preprint arXiv:1805.09980* (2018).

[12] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[13] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. 2012. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling* 52, 7 (2012), 1757–1768.

[14] David Janz, Jos van der Westhuizen, and José Miguel Hernández-Lobato. 2017. Actively learning what makes a discrete sequence valid. *arXiv preprint arXiv:1708.04465* (2017).

[15] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction Tree Variational Autoencoder for Molecular Graph Generation. In *Proceedings of the 35th International Conference on Machine Learning-Volume 80*. JMLR. org, 2323–2332.

[16] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983* (2018).

[17] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[18] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2018. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[19] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. 2017. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1945–1954.

[20] G. Landrum. 2006. *Rdkit: Open-source cheminformatics*. https://www.rdkit.org/

[21] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter W. Battaglia. 2018. Learning Deep Generative Models of Graphs. *CoRR* abs/1803.03324 (2018). arXiv:1803.03324 http://arxiv.org/abs/1803.03324

[22] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. 2018. Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*. 7795–7804.

[23] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. 2018. Constrained Graph Variational Autoencoders for Molecule Design. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 7795–7804. http://papers.nips.cc/paper/8005-constrained-graph-variational-autoencoders-for-molecule-design.pdf

[24] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359* (2018).

[25] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. 2018. Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*. 6114–6125.

[26] Emmanuel Noutahi, Dominique Beani, Julien Horwood, and Prudencio Tossou. 2019. Towards interpretable sparse graph representation learning with laplacian pooling. *arXiv preprint arXiv:1905.11577* (2019).

[27] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. 2019. MolecularRNN: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372* (2019).

[28] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. 2014. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data* 1 (2014), 140022.

[29] J.-L. Reymond, L. Rudiggkeit, L. Blum, and R. van Deursen. 2012. The enumeration of chemical space. *Comput Mol Sci* 2, 5 (2012), 717–733.

[30] Karl Ridgeway and Michael C Mozer. 2018. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*. 185–194.

[31] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. 2012. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of chemical information and modeling* 52, 11 (2012), 2864–2875.

[32] Bidisha Samanta, Abir De, Niloy Ganguly, and Manuel Gomez-Rodriguez. 2018. Designing random graph models using variational autoencoders with applications to chemical design. *arXiv preprint arXiv:1802.05283* (2018).

[33] P. Schneider and G. Schneider. 2016. De Novo Design at the Edge of Chaos. *J Medicinal Chem* 59, 9 (2016), 4077–4086.

[34] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. 2018. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science* 4, 1 (2018), 120–131.

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

[36] D. Stumpfe and B. Bajorath. 2011. Similarity Searching. *Comput Mol Sci* 1, 2 (2011), 260–282.

[37] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

[38] D. Weininger. 1988. SMILES, a chemical language and information system. *J Chem Information and Comput Sci* 28, 1 (1988), 31–36.

[39] G. M. Whitesides. 2015. Reinventing Chemistry. *Angew Chem Int Ed Engl* 54, 11 (2015), 3196–209.

[40] Jiaxuan You, Rex Ying, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. *arXiv preprint arXiv:1802.08773* (2018).

[41] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2019. Infovae: Information maximizing variational autoencoders. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.