Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Original Paper

Structural Bioinformatics

Small Molecule Generation via Disentangled Representation Learning

Yuanqi Du[1], **Xiaojie Guo**[2], **Yinkai Wang**[1], **Amarda Shehu**[1,3,4,5,*], **and Liang Zhao**[6,*]

[1]Department of Computer Science, George Mason University, Fairfax, 22030, USA and
[2]Department of Information Technology and Science, George Mason University, Fairfax, 22030, USA and
[3]Center for Advancing Human-Machine Partnerships (CAHMP), Fairfax, 22030, USA and
[4]Department of Bioengineering, George Mason University, Fairfax, 22030, USA and
[5]School of System Biology, George Mason University, Manassas, 200110, USA and
[6]Department of Computer Science, Emory University, Atlanta, 30322, USA.

*To whom correspondence should be addressed.

Associate Editor: Jinbo Xu

Received on 07/29/2021; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Expanding our knowledge of small molecules beyond what is known in nature or designed in wet laboratories promises to significantly advance cheminformatics, drug discovery, biotechnology, and material science. In-silico molecular design remains challenging, primarily due to the complexity of the chemical space and the non-trivial relationship between chemical structures and biological properties. Deep generative models that learn directly from data are intriguing, but they have yet to demonstrate interpretability in the learned representation, so we can learn more about the relationship between the chemical and biological space. In this paper, we advance research on disentangled representation learning for small molecule generation. We build on recent work by us and others on deep graph generative frameworks, which capture atomic interactions via a graph-based representation of a small molecule. The methodological novelty is how we leverage the concept of disentanglement in the graph variational autoencoder framework both to generate biologically-relevant small molecules and to enhance model interpretability.

Results: Extensive qualitative and quantitative experimental evaluation in comparison with state of the art models demonstrate the superiority of our disentanglement framework. We believe this work is an important step to address key challenges in small molecule generation with deep generative frameworks.

Availability: Training and generated data are made available at `https://ieee-dataport.org/documents/dataset-disentangled-representation-learning-interpretable-molecule-generation`. All code is made available at `https://anonymous.4open.science/r/D-MolVAE-2799/`.

Contact: liang.zhao@emory.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Expanding our knowledge of small molecules beyond what is known in nature or designed in wet laboratories promises to significantly advance drug discovery, biotechnology, and material science (Whitesides, 2015). In-silico molecule design is central to cheminformatics research but remains challenging (Schneider and Schneider, 2016). Studies estimate that 10^{60} drug-like molecules are synthetically-accessible (Reymond *et al.*, 2012). This size of chemical space is beyond the scope of even high-throughput wet-laboratory technologies.

A multi-decade journey in cheminformatics research informs us of several challenges for small molecule generation. The first concerns the poorly-understood and complex relationship between chemical and biological space. Not all molecules in the vast chemical space meet desired biological/functional properties of interest, such as water soluble, drug-likeness, and more (Ramakrishnan *et al.*, 2014). Moreover, changes to the chemical structure to optimize along a biological criterion may worsen other criteria; the search space that links chemical and biological space may be rich in with barriers separating neighboring local optima.

Until a decade ago, molecule generation, widely referred to as computational screening, was dominated by similarity search methods (Stumpfe and Bajorath, 2011). While conceptually straightforward, these methods were limited in their ability to generate novel small molecules. Advances in machine learning expedited progress. Shallow models were not very effective (Ellman, 1996; Yoshikawa *et al.*, 2018; Renz *et al.*, 2020; Xue *et al.*, 2019), as they relied heavily on domain insight to formulate and construct meaningful representations of small molecules. Due to their inherent ability to learn directly from data, deep generative models then made a debut. Initial efforts utilized a linear representation of molecules, known as SMILES (Weininger, 1988), which stands for "molecular-input line-entry system". SMILES is a formal grammar that describes molecules with an alphabet of characters; aromatic and aliphatic carbon atoms are denoted by 'c' and 'C', oxygen atoms by 'O', single bonds by '-', double bonds by '=', etc. The SMILES representation allows addressing molecule generation as a string generation problem. Deep learning methods based on the recurrent neural network (RNN) framework suddenly became useful (Gómez-Bombarelli *et al.*, 2018; Segler *et al.*, 2018; Kusner *et al.*, 2017). However, SMILES-based deep models could generate few valid molecules. In response, later works (Kusner *et al.*, 2017; Dai *et al.*, 2018) added syntactic and semantic constraints. In other works, models were guided to generate valid SMILES through active learning, reinforcement learning, and additional training signals (Janz *et al.*, 2017; Guimaraes *et al.*, 2017; Janz *et al.*, 2017; Guimaraes *et al.*, 2017). While some improvements were observed, generating valid molecules remained challenging.

Graph-generative deep models leverage a more expressive representation of a molecule via the concept of a molecular graph. The atoms are represented as vertices and the bonds as edges connecting the vertices. In deep learning literature, graph-generative models are based on the variational autoencoder (VAE) (Simonovsky and Komodakis, 2018; Samanta *et al.*, 2018; Jin *et al.*, 2018; Dai *et al.*, 2018; Blaschke *et al.*, 2018) or generative adversarial networks (GANs) (Bojchevski *et al.*, 2018; Guo *et al.*, 2018). For instance, GraphRNN (You *et al.*, 2018) builds an autoregressive generative model based on a generative RNN that generates the graph one vertex at a time. In contrast, GraphVAE (Simonovsky and Komodakis, 2018) represents each graph in terms of its adjacent matrix and feature vectors of vertices. A VAE model is then utilized to learn the distribution of the graphs conditioned on a latent representation at the graph level. Other works (Grover *et al.*, 2019; Kipf and Welling, 2016) encode the vertices into vertex-level embeddings and predict the edges between each pair of vertices to generate a graph.

The adoption of graph-generative models for small molecule generation has been rapid. Current graph generative models for molecule generation leverage the VAE framework to address two subtasks: (1) encoding: learning a low-dimensional, latent code/representation of a molecular graph; (2) decoding: learning to map the latent representation back into a (reconstructed) molecular graph. For instance, work in (Simonovsky and Komodakis, 2018) generates molecular graphs by predicting their adjacency matrices. Work in (Liu *et al.*, 2018a) generates molecules through a constrained graph generative model that enforces validity by generating a molecule one atom at a time. These works generate more valid molecules than SMILES-based models and additionally subject generated molecules to the sanitization checks in RDKit [1].

Graph-generative VAEs represent a promising platform that we leverage in this paper, but current graph-generative VAEs for small molecule generation fall short. The learned latent representation has all the latent factors entangled which limits the model transparency and interpretability. Specifically, these models do not facilitate linking the chemical space to the biological space and so do not advance our understanding of complex relationship between chemical and biological space for small molecules. Facilitating this linking is central not only for molecule generation but also for molecule optimization Alemi *et al.* (2017), an important and related task that beyond the scope of this paper.

In this paper we advance research on small molecule representation learning for molecule generation by disentanglement enhancement. Disentangled representation learning is an active research area, particularly in image representation learning (Alemi *et al.*, 2017; Chen *et al.*, 2018; Higgins *et al.*, 2017a; Kim and Mnih, 2018; Guo *et al.*, 2021) and has been shown key to improving model generalizability and robustness against adversarial attacks, and even facilitate debugging and auditing (Alemi *et al.*, 2017; Doshi-Velez and Kim, 2017). While a comprehensive review is beyond the scope of this paper, we point to recent approaches that modify the VAE objective by adding, removing, or altering the weight of individual terms in the loss function to improve disentanglement (Alemi *et al.*, 2017; Chen *et al.*, 2018; Esmaeili *et al.*, 2019; Kim and Mnih, 2018; Kumar *et al.*, 2018; Lopez *et al.*, 2018; Zhao *et al.*, 2019; Guo *et al.*, 2020; Du *et al.*, 2021a). Currently, however, we do not know the best approach to learn disentangled representations of graph data. This includes the small molecule generation domain. In a recent workshop paper (Du *et al.*, 2020), we demonstrated that learning disentangled representations results in better molecule generation over methods that do not leverage disentanglement. However, as our goal was a proof-of-concept demonstration that VAEs for disentangled representation learning achieve good assessment for small molecule generation, the study was limited to classic disentanglement and focused on few datasets of known small molecules of the same size.

Here we propose a graph-generative VAE framework that learns a disentangled code/representation, so that we may additionally elucidate how the factors that encode chemical structure control biological properties. Specifically, we design and evaluate the D-MolVAE framework, which stands for Disentangled Molecule VAE. The framework permits various mechanisms for disentanglement, resulting in several novel deep graph-generative models, which we compare to one another and many other state-of-the-art methods on benchmark datasets across several metrics.

Our experiments show that the D-MolVAE framework is effective and superior at generating valid, novel, and unique small molecules over other methods. The framework also accommodates variable-size molecules which improves its scope and applicability. Our experiments additionally show that disentanglement representation learning is valuable for better interpretation and understanding of the relationship between the chemical space and the biological space; the proposed D-MolVAE models are better able to capture the underlying graph statistics and distributions of various biological properties.

The D-MolVAE models effectively implement a trade-off between the disentanglement enhancement and the reconstruction. Our experiments show that explicit disentanglement enforcement does not hurt performance. In fact, the models are superior over many methods. Taken altogether, our findings suggest that the disentangled factors provide an advantage with respect to the quality of generated molecules, as well as the linking of the chemical and biological space. Our experiments suggest

[1] RDKit: Open-source cheminformatics; http://www.rdkit.org

several models as promising platforms for further exploring disentangled representations for improving small molecule generation.

2 Methods

We first define and formalize the problem. Then we describe the graph-generative models based on VAE framework, namely D-MolVAE, focusing the description on the variants of disentanglement terms proposed to obtain different disentangled graph-generative VAE models.

2.1 Problem Formulation

Let us represent a molecule as a graph $G = (\mathcal{V}, \mathcal{E}, E, F)$. The N atoms of the molecule constitute the N vertices V of graph G. The M bonds connecting pairs of atoms in the molecule constitute the edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where $e_{i,j} \in \mathcal{E}$ is an edge connecting vertices $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$. $G = (\mathcal{V}, \mathcal{E}, E, F)$ also contains E and F. $E \in \mathbb{R}^{N \times N \times K}$ is the edge type tensor that records the K bond types. Specifically, $E_{i,j} \in \mathbb{R}^{1 \times K}$ is an one-hot vector encoding the type of edge $e_{i,j}$. $F \in \mathbb{R}^{N \times K'}$ is the vertex type feature matrix that records the K' atom types. Specifically, $F_i \in \mathbb{R}^{1 \times K'}$ is the one-hot encoding vector denoting the type of atom v_i.

The objective in graph generative disentangled representation learning is to learn the joint distribution of G and a set of generative disentangled latent factors/variables $Z \in \mathbb{R}^{N \times L}$, such that the observed graph G can be generated as $p(G|Z)$. Note that L is the dimensionality of the latent factors. Disentanglement denotes the additional constraint that the individual variables in Z be independent from one another.

2.2 D-MolVAE Framework

Two challenges present themselves with the above formulation: (1) how to integrate the disentanglement constraint and the reconstruction quality constraint in the loss function that guides learning; (2) how to efficiently encode and decode molecules/graphs of different sizes. We first show how the first challenge is addressed in the D-MolVAE framework via a generative objective function. We show in this context that different approaches here can result in different models. Then we show how the second challenge is addressed via variable-size edge-to-edge and edge-to-vertex convolution operators in D-MolVAE.

We are inspired by disentanglement representation learning in the image domain (Higgins *et al.*, 2017a), where 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 allows explicitly denoting the parameters characterizing this distribution.

Learning $p_\theta(G|Z)$ requires the inference of its posterior $p_\theta(Z|G)$, which is intractable. So, one defines instead an approximated posterior $q_\phi(Z|G)$ that is computationally tractable. In disentangled representation learning, one needs to additionally ensure that the inferred latent variables Z from $q_\phi(Z|G)$ capture all the generative factors in a disentangled manner. This is achieved by introducing a constraint to match $q_\phi(Z|G)$ to a well-disentangled prior $p(Z)$ that controls the capacity of the latent information bottleneck and embodies the statistical independence mentioned above. An isotropic unit Gaussian suffices; that is, $p(Z) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ is an $N \times N$ identity matrix. This leads to the following constrained optimization problem:

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

In the above equation, $\mathcal{D}$ refers to the observed set of graphs (corresponding to molecules in the training dataset), $D_{KL}(\cdot)$ is the Kullback–Leibler divergence (KLD) that allows comparing two probability distributions, and ϵ is a parameter that specifies the strength of the applied constraint; that is, ϵ allows weighting how much we want the disentanglement constraint to be enforced.

Unfortunately, the above constraint formulated to achieve disentanglement is intractable. So, an aggregate objective (loss) function is formulated instead, where the above constraint and the reconstruction error in a VAE are combined together as in:

$$\max_\theta \mathbb{E}_{p_\theta(Z)}[\log p_\theta(G|Z)] - \beta D_{KL}(q_\phi(Z|G)||p(Z)) \qquad (2)$$

This aggregation is similar to the beta-VAE (Higgins *et al.*, 2017b) that first introduced the notion of disentanglement (though not for graph data). Note that β weighs how important it is to enforce the disentanglement constraint. Specifically, when $\beta = 1$, one obtains a vanilla VAE (Kingma and Welling, 2013). We direct the interested reader to work in (Higgins *et al.*, 2017b) to understand the effects of β.

2.3 Disentanglement-enhanced Models

By considering different approaches to enforce disentanglement, we obtain different instantiations of our D-MolVAE framework, namely, D-MolVAE-V, D-MolVAE-β, D-MolVAE-DIP-I, D-MolVAE-DIP-II, and D-MolVAE-VIB.

D-MolVAE-V: We extend the previous work on disentangled variational auto-encoders (Kingma and Welling, 2013; Esmaeili *et al.*, 2019) into that for graph-structured data, as follows:

$$\begin{aligned}
&\mathcal{L}(\theta, \phi, G, Z, \beta) = -D_{KL}(p_\theta(Z, G)||q_\phi(G, Z)) \\
&= \mathbb{E}_{q_\phi(Z,G)}[log \frac{p_\theta(G, Z)}{p_\theta(G)p(Z)} + log \frac{q(G)q_\phi(Z)}{q_\phi(G, Z)} + log \frac{p_\theta(G)}{q(G)} + log \frac{p(Z)}{q_\phi(Z)}] \\
&= \mathbb{E}_{q_\phi(Z,G)}[\underbrace{log \frac{p_\theta(G|Z)}{p_\theta(G)}}_{①} - \underbrace{log \frac{q_\phi(Z|G)}{q_\phi(Z)}}_{②}] \\
&\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{③} - \underbrace{D_{KL}(q_\phi(Z)||p(Z))}_{④} \qquad (3)
\end{aligned}$$

In the above, terms ③ and ④ enforce consistency between the marginal distributions over G and z. Specifically, minimizing the KLD in term ③ maximizes the marginal likelihood $\mathbb{E}_{q(G)} log p_\theta(G)$; maximizing the *disentangled inferred priors* term ④ enforces the distance between $q_\phi(Z)$ and $p(Z)$. Terms ① and ② enforce consistency between the conditional distributions. Specifically, term ① maximizes the correlation for each Z that generates each G^n; when $Z \sim q_\phi(Z|G^n)$ is sampled, the likelihood $p_\theta(G^n|Z)$ should be higher than the marginal likelihood $p_\theta(G^n)$. Meanwhile, term ② regularizes term ① by minimizing the mutual information $I(Z, G)$ in the inference model.

The D-MolVAE-V objective is defined as:

$$\begin{aligned}
\mathcal{L}(\theta, \phi, G, Z, \beta) = \mathbb{E}_{q_\phi(Z,G)}&[\underbrace{log \frac{p_\theta(G|Z)}{p_\theta(G)}}_{①} - \underbrace{log \frac{q_\phi(Z|G)}{q_\phi(Z)}}_{②}] \\
&\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{③} - \underbrace{D_{KL}(q_\phi(Z)||p(Z))}_{④} \qquad (4)
\end{aligned}$$

D-MolVAE-β: The penalty term $\beta > 1$ has proven useful to enforce the disentanglement of the latent variables without worsening reconstruction performance (Higgins *et al.*, 2017a). We emphasize that β allows balancing between reconstruction loss and KLD loss. So, our first model that introduces disentanglement for graph-based representation learning for small molecule generation is D-MolVAE-β. Its objective is similar to D-MolVAE-V. The only difference concerns the weighted KLD terms $\textcircled{1} + \textcircled{3} + \beta(\textcircled{2} + \textcircled{4})$ as follows:

$$\mathcal{L}(\theta,\phi,G,Z,\beta) = \mathbb{E}_{q_\phi(Z,G)}\underbrace{[log\frac{p_\theta(G|Z)}{p_\theta(G)}]}_{\textcircled{1}}\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{\textcircled{3}}$$
$$-\beta(\mathbb{E}_{q_\phi(Z,G)}\underbrace{[log\frac{q_\phi(Z|G)}{q_\phi(Z)}]}_{\textcircled{2}} - \underbrace{D_{KL}(q_\phi(Z)||p(Z)))}_{\textcircled{4}} \qquad (5)$$

D-MolVAE-DIP-I: It is important to note that term $\textcircled{2}$ may lead to poor reconstruction, when the disentanglement is heavily enforced by setting high values for the β parameter. To address this, the 'Disentangled Inferred Prior Variational Autoencoders' (DIPVAE) mode introduces a hyperparameter λ in term $\textcircled{4}$ (Kumar *et al.*, 2018). This term is also referred to as the 'inferred priors' term and enforces the distance between $q_\phi(z)$ and $p(z)$. The hyperparameter allows controlling the the trade-off between the reconstruction loss and the KLD term. We incorporate this idea to obtain D-MolVAE-DIP-I, whose objective function is now $\textcircled{1} + \textcircled{2} + \textcircled{3} + \lambda\textcircled{4}$, as in:

$$\mathcal{L}(\theta,\phi,G,Z,\beta) = \mathbb{E}_{q_\phi(Z,G)}[\underbrace{log\frac{p_\theta(G|Z)}{p_\theta(G)}}_{\textcircled{1}} - \underbrace{log\frac{q_\phi(Z|G)}{q_\phi(Z)}}_{\textcircled{2}}]$$
$$\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{\textcircled{3}} - \lambda\underbrace{D_{KL}(q_\phi(Z)||p(Z))}_{\textcircled{4}} \qquad (6)$$

D-MolVAE-DIP-II: Note that the $\textcircled{2}$ term represents the mutual information $I(Z,G)$ between the latent representation Z and the molecule G, which may lead to poor reconstruction. An alternative approach to balance between disentanglement and reconstruction is to discard term $\textcircled{2}$, thus obtaining DIPVAE-MolVAE-II, whose objective function now is:

$$\mathcal{L}(\theta,\phi,G,Z,\beta) = \mathbb{E}_{q_\phi(Z,G)}\underbrace{[log\frac{p_\theta(G|Z)}{p_\theta(G)}]}_{\textcircled{1}}\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{\textcircled{3}}$$
$$-\lambda\underbrace{D_{KL}(q_\phi(Z)||p(Z))}_{\textcircled{4}} \qquad (7)$$

D-MolVAE-VIB: The Variational Information Bottleneck (VIB) approach interprets the capacity of the KLD as the information bottleneck of the network (Alemi *et al.*, 2017). It proposes to add a controllable value C and a hyperparameter γ over the KLD term to control the information flowing through it. Later work demonstrates that by slowly increasing the value of C, the latent representation is able to gradually capture the semantic factors (Locatello *et al.*, 2018). Inspired by these works, we obtain our final model D-MolVAE-VIB, whose objective function is:

$$\mathcal{L}(\theta,\phi,G,Z,\beta) = \mathbb{E}_{q_\phi(Z,G)}\underbrace{[log\frac{p_\theta(G|Z)}{p_\theta(G)}]}_{\textcircled{1}}\underbrace{-D_{KL}(q(G)||p_\theta(G)))}_{\textcircled{3}}$$
$$-\gamma|\mathbb{E}_{q_\phi(Z,G)}\underbrace{[log\frac{q_\phi(Z|G)}{q_\phi(Z)}]}_{\textcircled{2}} - \underbrace{D_{KL}(q_\phi(Z)||p(Z))}_{\textcircled{4}} - C| \qquad (8)$$

2.4 Implementation Details

The variants are summarized in terms of their objectives in Table 1. The encoder and decoder architecture are summarized in Table 2. Finally, the hyperparameters used for training are related in Table 3. The rows refer to the different benchmark datasets, which we describe in Section 3. We observe that increasing β leads to a better disentangled representation, as later shown in Table 7.

Table 1. Summary of D-Mol-VAE variants in terms of their disentanglement objectives.

Model	Objectives
D-MolVAE-V	$\textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4}$
D-MolVAE-β	$\textcircled{1} + \textcircled{3} + \beta(\textcircled{2} + \textcircled{4})$
D-MolVAE-DIP-I	$\textcircled{1} + \textcircled{2} + \textcircled{3} + \lambda\textcircled{4}$
D-MolVAE-DIP-II	$\textcircled{1} + \textcircled{3} + \textcircled{4}$
D-MolVAE-VIB	$\textcircled{1} + \textcircled{3} + \gamma\textcircled{2} + \textcircled{4} + C$

Table 2. Encoders and decoders architectures. Each layer is expressed in the format as $< kernel\_size >< layer\_type >< Num\_channel >< Activation\_function >< stride\_size >$. FC refers to the fully connected layers).

Encoder	Decoder
Input: $G(\mathcal{V},\mathcal{E},E,F)$	Input$[z] \in \mathbb{R}^{100}$
FC.100 ReLU	FC.100 ReLU
GGNN.100 ReLU	GGNN.100 ReLU
GGNN.100 ReLU	GGNN.100 ReLU
FC.100	FC.bv (batch node size) FC.3 (edge)

Table 3. Hyperparameters used for training.

Dataset	Learning_rate	Batch_size	λ	Num_iteration
QM9	5e-4	64	1	10
ZINC	5e-4	8	1	5
MOSES	5e-4	4	1	5
CHEMBL	5e-4	4	1	5

3 Results

3.1 Datasets and Experimental Setup

We employ four benchmark datasets: QM9, ZINC, MOSES, and ChEMBL (Du *et al.*, 2021b). QM9 (Ramakrishnan *et al.*, 2014; Ruddigkeit *et al.*, 2012) contains around 134k stable small organic molecules with up to 9 heavy atoms (e.g. Carbon (C), Oxygen (O), Nitrogen (N) and Fluorine (F)). ZINC (Irwin *et al.*, 2012) contains approximately 250K drug-like chemical compounds with an average of 23 heavy atoms. The molecules in this dataset are more complex than in QM9. MOSES (Polykovskiy *et al.*, 2020) contains about 1.9M larger molecules with up to 30 heavy atoms. ChEMBL (Gaulton *et al.*, 2017) contains about 1.8M manually-curated bioactive molecules with drug-like properties. For QM9, we use the entire dataset, while for ZINC, MOSES, and ChEMBL which have larger molecules, we randomly sample 70k molecules from the entire dataset, and split into $6 : 1$ for training and validation. During testing, we generate 30k molecules for our experiments.

We utilize qualitative and quantitative experiments that evaluate the proposed D-MolVAE-V, D-MolVAE-β, D-MolVAE-DIP-I, D-MolVAE-DIP-II, and D-MolVAE-VIB. The models are pitched against 9 state-of-the-art deep generative models for molecule generation: *ChemVAE* (Gómez-Bombarelli *et al.*, 2018), *GrammarVAE* (Kusner *et al.*, 2017), *GraphVAE* (Simonovsky and Komodakis, 2018), *GraphGMG* (Li *et al.*, 2018), *SMILES-LSTM* (Sundermeyer *et al.*, 2012), *GraphNVP* (Madhawa *et al.*, 2019), *GRF* (Honda *et al.*, 2019), *GraphAF* (Shi *et al.*, 2019), and *CGVAE* (Liu *et al.*, 2018b). In the interest of brevity, summaries of the main computational ingredients in each of these models are related in the Supplementary Material. All experiments are conducted on a 64-bit machine with a 6 core Intel CPU i9-9820X, 32GB RAM, and an NVIDIA GPU (GeForce RTX 2080ti, 1545MHz, 11GB GDDR6).

3.2 Evaluating the Quality of Generated Molecules

Table 4 relates the comparative analysis. Each trained model is used to generate $30k$ molecules. For GraphGMG, we obtain 20K generated molecules from the GraphGMG authors. Results for ChemVAE, GrammarVAE, GraphVAE, and SMILES-LSTM are obtained from (Liu *et al.*, 2018b). The quality a generated dataset is evaluated via the 3 common metrics of *Novelty*, *Uniqueness*, and *Validity*. *Novelty* measures the fraction of generated molecules that are not in the training dataset. *Uniqueness* measures the fraction of generated molecules after and before removing duplicates. *Validity* measures the fraction of generated molecules that are chemically valid.

Table 4. Novelty, uniqueness, and validity, shown in %, are measured on a generated dataset. The highest value achieved on a metric is highlighted in boldface.

Model	QM9			ZINC		
	Validity	Novelty	Unique	Validity	Novelty	Unique
ChemVAE	10.00	90.00	67.50	17.00	98.00	30.98
GrammarVAE	30.00	95.44	9.30	31.00	100.00	10.76
GraphVAE	61.00	85.00	40.90	14.00	100.00	31.60
GraphGMG	-	-	-	89.20	89.10	99.41
SMILES-LSTM	94.78	82.98	96.94	96.80	**100.00**	**99.97**
GraphNVP	90.10	54.00	97.30	74.40	100.00	94.80
GRF	84.50	58.60	66.00	73.40	**100.00**	53.70
GraphAF	**100.00**	88.83	94.51	**100.00**	**100.00**	99.10
CGVAE	**100.00**	96.33	98.03	**100.00**	**100.00**	99.95
D-MolVAE-V	**100.00**	96.10	**99.15**	**100.00**	**100.00**	99.95
D-MolVAE-β	**100.00**	95.35	96.62	**100.00**	**100.00**	99.72
D-MolVAE-DIP-I	**100.00**	97.36	97.80	**100.00**	99.99	99.88
D-MolVAE-DIP-II	**100.00**	**98.31**	72.36	**100.00**	**100.00**	51.42
D-MolVAE-VIB	**100.00**	95.85	98.66	**100.00**	**100.00**	99.18

Table 5. Novelty, uniqueness, and validity, shown in %, are measured on a generated dataset. The highest value achieved on a metric is highlighted in boldface.

Model	MOSES			CHEMBL		
	Validity	Novelty	Unique	Validity	Novelty	Unique
CGVAE	99.97	99.97	95.33	**100.00**	99.97	99.85
D-MolVAE-V	**100.00**	**100.00**	99.70	**100.00**	**100.00**	14.85
D-MolVAE-β	**100.00**	**100.00**	99.73	**100.00**	**100.00**	99.35
D-MolVAE-DIP-I	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**99.96**
D-MolVAE-DIP-II	**100.00**	**100.00**	56.53	**100.00**	**100.00**	99.93
D-MolVAE-VIB	**100.00**	**100.00**	**100.00**	**100.00**	99.97	99.88

Table 4 allows making several observations. ChemVAE, GrammarVAE, and GraphVAE have the lowest performance. The D-MolVAE models achieve superior performance over the other models. In particular, all D-MolVAE models achieve 100% on validity on all datasets. Similar performance is observed on uniqueness, as well. Varied performance is observed on novelty, though all D-MolVAE models consistently outperform or match the performance of the other models; CGVAE is the only other model with a consistently good performance across all metrics on all datasets. This is not surprising, as our proposed models build over the CGVAE architecture but additionally enforce disentanglement. The explicit disentanglement enforcement seems to provide some benefits on higher novelty, in particular, on the QM9 dataset, over CGVAE. Taken altogether, these results suggest that the disentanglement enforcement does not reduce and actually improves performance; adding the disentanglement regularization does not influence the reconstruction error and so does not sacrifice the quality of generated molecules. It is worth noting that some of the proposed models, such as D-MolVAE-DIP-I and D-MolVAE-DIP-II, generate more novel molecules. Between the two, D-MolVAE-DIP-II generates more novel (nearly 100%) yet less unique (50%-70%) molecules due to the stronger constraint exerted by the KL Divergence term. In Table 5, we further evaluate the performance of our proposed methods and the strongest baseline, CGVAE, on two new datasets, MOSES and ChEMBL. In MOSES dataset, all the model achieve 100% validity and novelty, while D-MolVAE-VIB and D-MolVAE-DIp-I perform also 100% unique.In CHEMBL dataset, all the models achieve a comparable result except D-MolVAE-V on Unique.

3.3 Comparing the Learned Distribution to the Training Distribution

Given the above results, we now focus the comparison of our models against CGVAE. We measure the distance between the generated and the training datasets in terms of molecular properties and graph statistics, as shown in Table 6, utilizing two popular metrics, the Maximum Mean Discrepancy (MMD) (You *et al.*, 2018) and KL Divergence (KLD) (You *et al.*, 2018). MMD is used when comparing distributions of graph statistics, and KLD is used when comparing distributions of molecular properties; the molecular properties of interest are selected due to their low correlation, which is ideal for the disentanglement experiment setting that requires independent semantic factors. The correlation heatmap between commonly used molecular properties evaluated in QM9 dataset is shown in Figure 1. All these statistics are described in detail the Supplementary Material, where we also draw randomly-selected QM9 molecules over the generated dataset for each of the models.

In Table 6, the smaller the value, the more similar the generated set is to the training set on a property under comparison. Table 6 shows that all models reasonably preserve the distributions of properties in the training set. In comparison with CGVAE, our D-MolVAE models preserve more on the ZINC and MOSES dataset while less on the QM9 dataset. However, our models consistently perform well on all four datasets. The only dataset where CGVAE performs better than any of our models on about half of the properties (4/9) is on the QM9 dataset. CGVAE also performs comparably on KLD to at least one of our models on the CHEMLB dataset, but it is outperformed on MMD. On both the ZINC and the MOSES datasets, our models outperform CGVAE. In particular, D-MolVAE-VIB performs consistently well across all four datasets. The KLD between the training and the generated datasets are small, and this is further confirmed visually by plotting the distributions of the molecular properties cLogP, cLogS, PSA, rPSA and Drug-likeness for each model in Figures 2-7 in the Supplementary Material. These results make clear that our D-MolVAE models capture well the distributions of the molecular properties in the training dataset.

Altogether, these results suggest that the proposed models capture the underlying property distribution of the training dataset. Overall, all models balance well between information preservation and novelty in the generated molecules. Among all our D-MolVAE models, it is easily observed that D-MolVAE-VIB outperforms all the others along most metrics. Interestingly, even though disentanglement-enhanced models do not outperform the baselines in terms of capturing the synthesis accessibility (SA) score distribution, they generate novel molecules with higher SA score, e.g. MolVAE-VIB. This observation actually demonstrates the exploration power of the disentangled models and the better trade-off they allow us to achieve between exploration and exploitation. It is worth noting that one can choose over the disentangled models and the base models by preferences of exploration or exploitation.

Table 6. Comparing the difference between the training and generated distributions of graph properties via MMD and KLD. We abbreviate D-MolVAE by Mol, DIP by D, Degree by Deg, Clustering Coefficient by Coeff, Drug-likeness by Drug, and Rel PSA by RPSA. The best value per row is in boldface.

Dataset	Metric	CGVAE	Mol-V	Mol-β	Mol-DI	Mol-DII	Mol-VIB
QM9	MMD(Deg)	**0.0167**	0.0258	0.0541	0.0838	0.0238	0.0232
	MMD(CC)	0.0097	0.0051	0.0259	0.0175	0.0095	**0.0045**
	MMD(Orbit)	0.0018	0.0210	0.0021	0.0079	0.0031	**0.0017**
	KLD(cLogP)	0.08	0.41	0.44	0.35	0.46	**0.01**
	KLD(cLogS)	**0.06**	0.27	0.26	0.18	1.23	0.13
	KLD(Drug)	0.07	0.15	0.08	0.18	0.22	**0.04**
	KLD(RPSA)	**0.04**	0.29	0.11	0.18	0.51	**0.04**
	KLD(PSA)	**0.03**	0.07	0.07	0.30	0.09	**0.03**
	KLD(SA)	0.44	0.21	0.50	0.89	**0.16**	0.20
ZINC	MMD(Deg)	0.0023	**0.0005**	0.0043	0.0034	0.7962	0.0111
	MMD(CC))	0.0013	**0.0002**	0.0013	0.0005	0.0316	0.0363
	MMD(Orbit)	0.0005	0.0731	**0.0001**	**0.0001**	**0.0001**	0.0006
	KLD(cLogP)	0.67	0.59	**0.09**	0.67	0.30	0.23
	KLD(cLogS)	0.74	0.04	**0.09**	0.74	0.58	0.10
	KLD(Drug)	1.29	1.63	0.97	1.29	1.52	**0.01**
	KLD(RPSA)	0.78	0.47	0.31	0.79	1.17	**0.08**
	KLD(PSA)	0.56	0.06	0.14	0.59	**0.01**	0.12
	KLD(SA)	**0.56**	0.75	0.79	0.76	2.29	0.82
MOSES	MMD(Deg)	0.0052	0.0032	0.0031	0.0220	0.4520	**0.0024**
	MMD(CC))	0.0003	0.0027	0.0004	0.0005	**0.0000**	0.0002
	MMD(Orbit)	0.0009	0.0013	**0.0002**	0.0006	0.0217	0.0005
	KLD(cLogP)	0.47	**0.01**	0.96	0.12	0.37	0.25
	KLD(cLogS)	0.22	0.21	0.17	1.01	0.50	**0.16**
	KLD(Drug)	0.35	0.56	0.84	1.41	**0.33**	0.48
	KLD(RPSA)	0.04	**0.01**	0.18	0.93	0.97	0.05
	KLD(PSA)	0.07	0.22	0.36	0.71	0.58	**0.07**
	KLD(SA)	1.57	1.76	1.85	1.25	3.57	**1.09**
CHEMBL	MMD(Deg)	0.0028	0.6634	0.0022	0.0015	**0.0013**	0.0025
	MMD(CC))	0.0002	0.0010	0.0004	**0.0001**	0.0002	**0.0001**
	MMD(Orbit)	0.0004	0.0424	0.0010	**0.0002**	**0.0002**	0.0004
	KLD(cLogP)	**0.03**	0.05	0.31	0.04	0.04	**0.03**
	KLD(cLogS)	**0.04**	**0.04**	0.05	**0.04**	**0.04**	**0.04**
	KLD(Drug)	**0.01**	**0.01**	0.02	0.02	0.02	**0.01**
	KLD(RPSA)	**0.01**	0.02	**0.01**	**0.01**	**0.01**	**0.01**
	KLD(PSA)	**0.23**	0.24	0.25	0.24	0.25	**0.23**
	KLD(SA)	**0.07**	0.08	0.09	0.08	0.08	0.08

Table 7. Evaluation of disentanglement across all top models on each of the datasets. ↑ indicates that a higher value on a metric is better.

Dataset	Model	β-M(%)↑	F-M(%)↑	DCI↑	Mod↑
QM9	CGVAE	**100**	57.0	0.055	0.239
	Mol-V	**100**	50.0	0.019	0.233
	Mol-β	**100**	56.0	0.0466	0.223
	Mol-DI	**100**	61.2	0.023	**0.261**
	Mol-DII	**100**	62.0	0.0972	0.241
	Mol-VIB	**100**	**72.0**	**0.1282**	0.243
ZINC	CGVAE	**100**	48.0	0.011	0.195
	Mol-V	**100**	44.0	0.016	0.163
	Mol-β	**100**	52.0	0.016	0.151
	Mol-DI	**100**	52.4	0.010	**0.197**
	Mol-DII	**100**	50.0	0.019	0.188
	Mol-VIB	**100**	**58.0**	**0.036**	0.189
MOSES	CGVAE	**100**	38.0	0.059	0.184
	Mol-V	**100**	44.0	0.060	0.189
	Mol-β	**100**	46.0	0.061	0.186
	Mol-DI	**100**	**58.0**	0.062	0.209
	Mol-DII	**100**	50.0	0.071	0.212
	Mol-VIB	**100**	54.0	**0.078**	**0.253**
CHEMBL	CGVAE	82.0	61.3	0.181	0.500
	Mol-V	80.0	62.0	0.202	0.499
	Mol-β	82.6	62.3	**0.219**	0.491
	Mol-DI	84.0	62.0	0.209	0.481
	Mol-DII	80.0	64.0	0.213	0.456
	Mol-VIB	**85.3**	**64.6**	0.183	**0.504**

3.3.1 Quantitative Evaluation of Disentanglement Learning

Table 7 relates the evaluation of our models' disentanglement scores via β-M, F-M, MOD, and DCI, which are four popular metrics to evaluate disentanglement. Briefly, β-M (Higgins *et al.*, 2017a) measures disentanglement by examining the accuracy of a linear classifier that predicts the index of a fixed factor of variation. F-M (Kim and Mnih, 2018) addresses several issues by using a majority voting classifier on a different feature vector that represents a corner case in the β-M. The β-M and F-M metrics are formulated as follows:

$$x_{1,1}, x_{2,1}, \cdots, x_{1,L}, x_{2,L} \sim (f_k \cup \mathcal{N}(0,1)) \tag{9}$$

$$z_{1,1}, z_{2,1}, \cdots, z_{1,L}, z_{2,L} = p_{(z|x)}(x_{1,1}), p_{(z|x)}(x_{2,1}), \cdots, \tag{10}$$

$$p_{(z|x)}(x_{1,L}), p_{(z|x)}(x_{2,L}), \tag{11}$$

$$z_{diff} = \frac{1}{L}\sum_{l=1}^{L} |z_{1,l} - z_{2,l}|, \tag{12}$$

$$\beta - M = p(k|z_{diff}), \tag{13}$$

$$F - M = p(k| \operatorname*{argmin}_{d} \; \underset{l}{Var} \; z_l^d/\sigma_d), \tag{14}$$

MOD (Ridgeway and Mozer, 2018) measures whether each latent variable depends on at most a factor describing the maximum variation using their mutual information. We first calculate the mutual information between the latent representations and the values of the factors of variation in a matrix m. Then, we compute a vector t_i for each dimension of representation i. Finally, we average over the dimensions of the representation with N factors, as follows:

$$t_{i,f} = \begin{cases} \theta_i & \text{if } f = \text{argmax}_g \; m_{i,g} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$MOD = \frac{1}{I}\sum_i 1 - \frac{\sum_f (m_{i,f} - t_{i,f})^2}{\theta_i^2 (N-1)} \tag{16}$$

DCI (Eastwood and Williams, 2018) 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. For DCI, we first take the importance weights for each factor by fitting gradient boosted trees and form an importance matrix R. We then compute the relative importance of each dimension ρ_i and disentanglement score

DCI as follows:

$$\rho_i = \frac{\sum_j R_{ij}}{\sum_{ij} R_{ij}} \tag{17}$$

$$DCI = \sum_i \rho_i (1 - H(R_i)) \tag{18}$$

All implementation details are as in (Locatello *et al.*, 2018).

Table 7 shows that our models achieve the best overall disentanglement scores over CGVAE. Specifically, on the QM9 dataset with smaller molecules, D-MolVAE-DIP-I, D-MolVAE-DIP-II, and D-MolVAE-VIB achieve F-M scores of 61.2%, 62.0%, 72.0%, respectively, whereas CGVAE achieves only 57.0%. All models achieve comparable MOD scores, with D-MolVAE-DIP-I achieving the the highest. All models achieve a $\beta - M$ of 100%. D-MolVAE-VIB outperforms all others on the DCI score, and this observation holds across all four datasets. Interestingly, all models perform worse on the ZINC dataset, which contains larger molecules than the QM9 dataset. Similarly, on the MOSES dataset, all the models perform worse than on QM9 but better than on ZINC. Specifically, D-MolVAE-DIP-I and D-MolVAE-VIB rank as the top two on the $F - M$ metrics, and D-Mol-VAE achieves the best performance on the DCI and Mod metrics, with an up to 16% improvement over the second best model, D-MolVAE-DIP-II. On the CHEMBL dataset, D-MolVAE-DIV performs the best across the $\beta - M$, $F - M$, and Mod metrics. D-Mol-DIP-I achieves the second in $\beta - M$ (84.0%), while CGVAE performs only 82.0%. Nevertheless, D-Mol-β performs slightly better over D-Mol-DII on the DCI metric which achieves the best performance. Altogether, these results show that the proposed disentanglement-enhanced models improve the ability of a model for disentanglement learning, especially for D-MolVAE-VIB.

3.3.2 Relating Disentangled Factors to Molecular Properties

In Figure 1 we show how the learned disentangled factors relate to the biological properties computed on each generated molecule. The mutual information is calculated between each of the disentangled factors learned by CGVAE and the D-MolVAE models and the molecular properties computed on generated molecules. We focus the comparison here to the MOSES-trained CGVAE and D-MolVAE-VIB models but show all models on all datasets in the Supplementary Material.

Figure 1 clearly show that the factors learned by CGVAE relate weakly with the molecular properties. Such relationship is stronger on the disentangled factors learned by our D-MolVAE models, even though all models are unsupervised. Moreover, different disentangled factors from D-MolVAE-VIB tend to more clearly correlate to different properties than CGVAE, thanks to the disentanglement enhancement.

Figure 2 allows digging deeper into the impact of a property of interest by visualizing the change in the property over molecules generated when a particular latent factor is varied in a range, and others are kept fixed. We focus on one of our top models, D-MolVAE-VIB, and on PSA, which is a crucial consideration when generating molecules, as it directly relates to our ability to actually synthesize them in wet laboratories. We can clearly see that basically only one factor is majorly related to PSA, thanks to our disentanglement enhancement that strengthens the independence among different factors and hence minimizes the number of different factors correlated to a property (e.g., PSA). Figure 2 shows that one of the latent factors impacts PSA, and this is more clearly visible on the QM9 and MOSES datasets.

4 Conclusion

The evaluation presented in this paper suggests that the proposed disentanglement framework D-MolVAE is effective at generating valid, novel, and unique small molecules and outperforms several state-of-the-art generative models. This performance is due to the sequence decoding process and, specifically, valence checking and the stop-checking mechanism. Other graph-based generative models that lack this process (for instance, GraphVAE) suffer in this respect and generate invalid molecules. The variational inference in D-MolVAE also allows better capturing the distribution of the input dataset and so sampling novel and unique molecules from the learned distribution.

It is important to note that the loss functions in the models we propose here effectively implement a trade-off between the disentanglement enhancement and the reconstruction. The distributions of specific properties (for instance, synthesis accessibility) show the exploration-exploitation trade-off in the various disentangled models. Our analysis shows that explicit disentanglement enforcement does not hurt the proposed models; indeed, like CGVAE, the proposed models generate novel and unique molecules and even surpass CGVAE on some of the datasets; the disentangled factors provide an advantage. Moreover, the proposed D-MolVAE models better capture the underlying graph statistics and distributions of various biological properties. Our evaluation also reveals that different types of disentangled models have different abilities. In particular, the experiments suggest that D-MolVAE-VIB is a promising model for exploring disentangled representations.

We consider the proposed work to be a first step to address remaining challenges in small molecule generation. Beyond interpreting the generation process, it is important to precisely control the properties of generated molecules. The disentangled representation learning is this paper falls under the umbrella of unsupervised learning. Therefore, specific control and correspondence of latent factors to molecular properties of interest is not expected to be strong. Our analysis shows that, in principle, one can build over the models proposed here for such precise control. Ideally, given specific, target values for several properties of interest, one could decode back the latent variables into a molecule that achieves the target property values. Our future work will address such models.

We also note that current models, including those proposed and evaluated this paper, 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 so far. Doing both can be helpful in designing novel molecules while improving our understanding of the contribution of each element in the overall molecular properties of interest. We caution, however, that supervised representation learning, while useful in many specific applications, may also bias towards a known, target set of molecular properties and miss possibly interesting new discoveries. In our future work we hope to advance both unsupervised and supervised representation learning in small molecule generation.

Acknowledgements

This work is supported in part by the National Science Foundation Grant No. 1942594, No. 1755850, and No. 1907805. 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

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. (2017). Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net.

Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., and Chen, H. (2018). Application of generative autoencoder in de novo molecular design. *Molecular informatics*, **37**(1-2), 1700123.

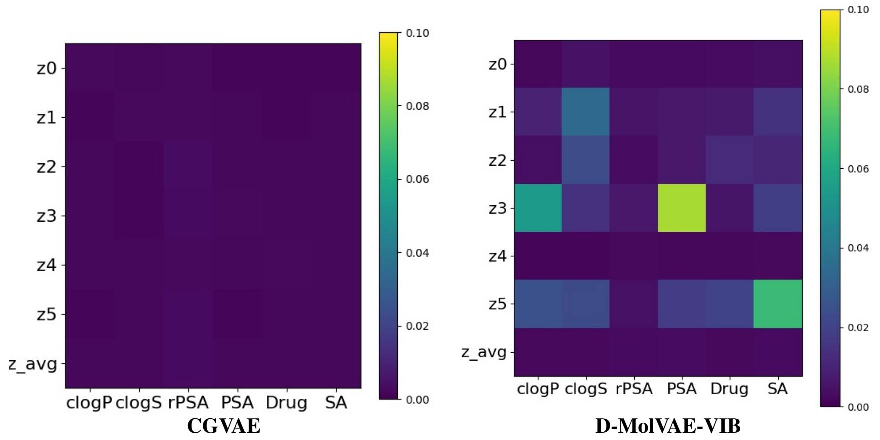


Fig. 1. The mutual information is calculated between each of the disentangled factors and the molecular properties computed on generated molecules.

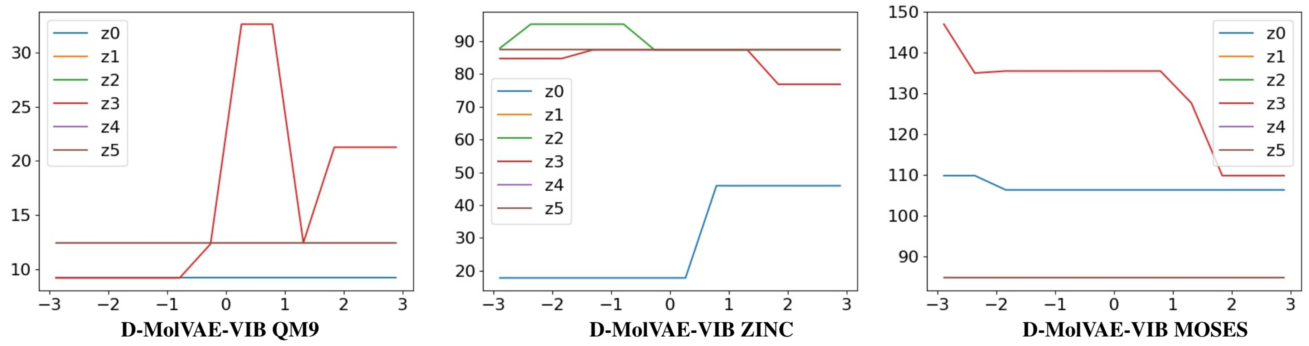


Fig. 2. Change in PSA is tracked as a latent factor is varied in a range while keeping all others fixed. Focus here is on the latent factors learned by D-MolVAE-VIB.

Bojchevski, A., Shchur, O., Zügner, D., and Günnemann, S. (2018). Netgan: Generating graphs via random walks. In *International Conference on Machine Learning*, pages 609–618.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620.

Dai, H., Tian, Y., Dai, B., Skiena, S., and Song, L. (2018). Syntax-directed variational autoencoder for structured data. In *International Conference on Learning Representations*.

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

Du, Y., Guo, X., Shehu, A., and Zhao, L. (2020). Interpretable molecule generation via disentanglement learning. In *11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics Workshops: Comput Struct Biol Workshop (CSBW)*, pages 1–8.

Du, Y., Wang, Y., Alam, F., Lu, Y., Guo, X., Zhao, L., and Shehu, A. (2021a). Deep latent-variable models for controllable molecule generation. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 372–375. IEEE.

Du, Y., Wang, S., Guo, X., Cao, H., Hu, S., Jiang, J., Varala, A., Angirekula, A., and Zhao, L. (2021b). Graphgt: Machine learning datasets for graph generation and transformation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Eastwood, C. and Williams, C. K. (2018). A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations ICLR*. OpenReview.net.

Ellman, J. A. (1996). Design, synthesis, and evaluation of small-molecule libraries. *Accounts of chemical research*, **29**(3), 132–143.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and van de Meent, J.-W. (2019). Structured disentangled representations. *Proceedings of Machine Learning Research*, **89**.

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

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, **4**(2), 268–276.

Grover, A., Zweig, A., and Ermon, S. (2019). Graphite: Iterative generative modeling of graphs. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2434–2444.

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

Guo, X., Wu, L., and Zhao, L. (2018). Deep graph translation. *arXiv preprint arXiv:1805.09980*.

Guo, X., Zhao, L., Nowzari, C., Rafatirad, S., Homayoun, H., and Dinakarrao, S. M. P. (2019). Deep multi-attributed graph translation with node-edge co-evolution. In *the 19th International Conference on Data Mining (ICDM 2019)*.

Guo, X., Du, Y., and Zhao, L. (2020). Property controllable variational autoencoder via invertible mutual dependence. In *International Conference on Learning Representations*.

Guo, X., Du, Y., and Zhao, L. (2021). Deep generative model for spatial networks. *27th ACM SIGKDD international conference on knowledge discovery & data mining*.

Hassoun, M. H. *et al.* (1995). *Fundamentals of artificial neural networks*. MIT press.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017a). beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017b). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net.

Honda, S., Akita, H., Ishiguro, K., Nakanishi, T., and Oono, K. (2019). Graph residual flow for molecular graph generation. *arXiv preprint arXiv:1909.13521*.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. (2012). Zinc: a free tool to discover chemistry for biology. *Journal of chemical information* *and modeling*, **52**(7), 1757–1768.

Janz, D., van der Westhuizen, J., and Hernández-Lobato, J. M. (2017). Actively learning what makes a discrete sequence valid. *arXiv preprint arXiv:1708.04465*.

Jin, H., Song, Q., and Hu, X. (2018). Discriminative graph autoencoder. In *Intl Conf on Big Knowledge (ICBK)*. IEEE.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. (2017). Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1945–1954. JMLR. org.

Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P. W. (2018). Learning deep generative models of graphs. *CoRR*, **abs/1803.03324**.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. (2018a). Constrained graph variational autoencoders for molecule design. In *Advances in neural information processing systems*, pages 7795–7804.

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. (2018b). Constrained graph variational autoencoders for molecule design. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7795–7804. Curran Associates, Inc.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2018). Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*.

Lopez, R., Regier, J., Jordan, M. I., and Yosef, N. (2018). Information constraints on auto-encoding variational bayes. In *Advances in Neural Information Processing Systems*, pages 6114–6125.

Madhawa, K., Ishiguro, K., Nakago, K., and Abe, M. (2019). Graphnvp: An invertible flow model for generating molecular graphs. *arXiv preprint arXiv:1905.11600*.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. (2020). Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. (2014). Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, **1**, 140022.

Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S., and Klambauer, G. (2020). On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*.

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

Ridgeway, K. and Mozer, M. C. (2018). Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, pages 185–194.

Ruddigkeit, L., Van Deursen, R., Blum, L. C., and Reymond, J.-L. (2012). Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, **52**(11), 2864–2875.

Samanta, B., De, A., Ganguly, N., and Gomez-Rodriguez, M. (2018). Designing random graph models using variational autoencoders with applications to chemical design. *arXiv preprint arXiv:1802.05283*.

Schneider, P. and Schneider, G. (2016). De novo design at the edge of chaos. *J Medicinal Chem*, **59**(9), 4077–4086.

Segler, M. H., Kogej, T., Tyrchan, C., and Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, **4**(1), 120–131.

Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., and Tang, J. (2019). Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*.

Simonovsky, M. and Komodakis, N. (2018). Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pages 412–422. Springer.

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

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

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

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

Xue, D., Gong, Y., Yang, Z., Chuai, G., Qu, S., Shen, A., Yu, J., and Liu, Q. (2019). Advances and challenges in deep generative models for de novo molecule generation. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, **9**(3), e1395.

Yoshikawa, N., Terayama, K., Sumita, M., Homma, T., Oono, K., and Tsuda, K. (2018). Population-based de novo molecule generation, using grammatical evolution. *Chemistry Letters*, **47**(11), 1431–1434.

You, J., Ying, R., Ren, X., Hamilton, W. L., and Leskovec, J. (2018). Graphrnn: Generating realistic graphs with deep auto-regressive models. *arXiv preprint arXiv:1802.08773*.

Zhao, S., Song, J., and Ermon, S. (2019). Infovae: Information maximizing variational autoencoders. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.

Supplementary Data

Architecture of D-MolVAE

A preliminary version of our graph-based VAE framework D-MolVAE has appeared in our recent workshop paper (Du *et al.*, 2020). Here, in addition to implementing the trade-off between disentanglement and reconstruction loss in various ways, thus obtaining the different models described above, we also extend the framework to handle variable-size graphs (thus, variable-size molecules). For the sake of completeness, we summarize the entire framework for the interested reader.

D-MolVAE is its essence a VAE, with an encoder and decoder. As in a VAE, the encoder learns the mean and standard deviation of the latent representation of the input; the decoder decodes the sampled latent representation to reconstruct the input. In D-MolVAE, the *graph* encoder models the prior distributions $q_\phi(Z|G)$ by generating the mean μ and standard variation σ of the learned distribution. The graph decoder models $p_\theta(G|Z)$. Graphs are generated by sampling the inferred mean μ and standard derivation σ of the learned distribution. Each component is described in detail below.

Molecule Encoder

To model the prior distributions $q_\phi(Z|G)$, the D-MolVAE encoder is constructed based on a graph neural network (GNN) (Guo *et al.*, 2019). The GNN embeds each vertex in an input graph G into the L-dimension latent space following the distribution $q_\phi(Z|G)$ parameterized by mean μ_i and standard deviation vectors σ_i for each vertex v_i, which is the output of the GNN. As a result, by sampling from the modeled distribution, $Z = \{Z_1, ..., Z_N\}$ are the variables containing the representation vectors for all the vertices.

Molecule Decoder

The molecule decoder models the distribution $p_\theta(G|Z)$ by generating the molecule graph G conditioned on the latent representation variables Z sampled from the distribution learned by the encoder. The process proceeds in an auto-regressive style. In each step, a focus vertex is selected to be visited, and related edges are then generated. The vertices are ordered via breadth-first traversal.

Specifically, the molecule decoder contains three steps, *vertex initialization*, *vertex update*, and *edge selection and labeling*.

Vertex Initialization Here is where we can handle variable-size graphs. We set N as the upper bound on the number of vertices in a generated graph. Briefly, an initial state $h_i^{(t=0)}$ is assigned to each vertex v_i in a set of initially-unconnected vertices. Specifically, $h_i^{(t=0)}$ is the concatenation $[Z_i, \tau_i]$, where τ_i is a one-hot vector indicating 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 multi-layer perception (MLP) (Hassoun *et al.*, 1995). From these vertex-level states, one can then calculate global representations $H(t)$, which is the average representation of vertices in the connected component at generation step t. In addition to N working vertices, a special "stop vertex" is initialized to a learned representation h_{end} for the purpose of termination, detailed as below.

Edge Selection and Labeling At each step t, a focus vertex v_i is picked from the queue of vertices. An edges $e_{i,j}$ is selected from vertex v_i to vertex v_j with label $E_{i,j}$. Specifically, for each non-focus vertex v_j, one constructs 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 (the path) between two vertices v_i, v_j. These representations are utilized to produce a distribution over candidate edges, as follows:

$$p(e_{i,j}, E_{i,j}|\eta_{i,j}^{(t)}) = p(E_{i,j}|\eta_{i,j}^{(t)}, e_{i,j}) \cdot p(e_{i,j}|\eta_{i,j}^{(t)}) \tag{19}$$

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

$$p(e_{i,j}|\eta_{i,j}^{(t)}) = \frac{M_{i,j}^{(t)} \exp(f_{\text{vertex}}(\eta_{i,j}^{(t)}))}{\sum_k^N M_{i,k}^{(t)} \exp(f_{\text{vertex}}(\eta_{i,k}^{(t)}))}, \tag{20}$$

$$p(E_{i,j} = l|\eta_{i,j}^{(t)}) = \frac{m_{i,j,l}^{(t)} \exp([f_{\text{bond}}(\eta_{i,j}^{(t)})]_l)}{\sum_u^L m_{i,j,u}^{(t)} \exp([f_{\text{bond}}(\eta_{i,j}^{(t)})]_u)}, \tag{21}$$

In the above, 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. New edges are sampled one by one from the above learned distributions. Any vertices that are connected to the graph for the first time during this edge selection are added to the vertex queue.

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

$$r_i^{(s+1)} = GRU[r_i^{(s)}, \sum_{j \leftrightarrow i} \text{MLP}(r_j^{(s)})] \tag{22}$$

$$h_i^{(t+1)} = r_i^{(S)}, \tag{23}$$

In the above, $r_i^{(0)} = h_i^{(0)}$, and the sum runs over all edges in the current graph. 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 vertex, the edges to a vertex v_i are kept added until an edge to the stop vertex is selected. When that happens, the "focus" then moves to vertex v_i, v_i is regarded as "closed" vertex. The next focus vertex is then selected from the focus queue. In this way, a single connected component is grown in a breadth-first manner. The vertex and edge generations continue until the vertex queue is empty. There may be unconnected vertices left at the end; these are 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 physically-realistic molecule. In the molecule graph, each atom type has a fixed valency. For example, vertex type "H" (a hydrogen atom) has a valency of 1, and vertex type "O" (an oxygen atom) has a valency of 2. Throughout the generation process, two types of masks $M^{(t)}_{i,j}$ and $m^{(t)}_{i,j,l}$ are used to guarantee that the bonds b_i of each atom never exceed the atom 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, $M^{(t)}_{i,j}$ also handles avoidance of edge duplication and self loops, and is defined as:

$$\begin{aligned} M^{(t)}_{i,j} = \mathbb{I}(b_i < b^*_i) \times \mathbb{I}(b_j < b^*_j) \times \mathbb{I}(e_{i,j} \text{ not exist}) \times\ \mathbb{I}(i \neq j) \\ \times \mathbb{I}(v_i \text{ is focus}) \end{aligned} \tag{24}$$

In the above, $\mathbb{I}(\cdot)$ is an indicator function; as a special case, connections to the stop vertex are always unmasked. Further, when selecting the label for a chosen edge, we again avoid violating the valency constraint by defining $m^{(t)}_{i,j,l} = M^{(t)}_{i,j} \times \mathbb{I}(b^*_j - b_j < l)$, where l refer to the bond type, and $l = 1, 2, 3$ indicates single, double, and triple bond types, respectively.

Comparative Analysis

The 5 proposed D-MolVAE models are pitched against 9 state-of-the-art deep generative models for molecule generation: *ChemVAE* (Gómez-Bombarelli *et al.*, 2018), *GrammarVAE* (Kusner *et al.*, 2017), *GraphVAE* (Simonovsky and Komodakis, 2018), *GraphGMG* (Li *et al.*, 2018), *SMILES-LSTM* (Sundermeyer *et al.*, 2012), *GraphNVP* (Madhawa *et al.*, 2019), *GRF* (Honda *et al.*, 2019), *GraphAF* (Shi *et al.*, 2019), and *CGVAE* (Liu *et al.*, 2018b).

ChemVAE (Gómez-Bombarelli *et al.*, 2018) is a generative model that converts discrete representations of molecules to and from a multidimensional continuous representation. *GrammarVAE* (Kusner *et al.*, 2017) enforces syntactic and semantic constraints over SMILES strings via context free and attribute grammars. *GraphVAE* (Simonovsky and Komodakis, 2018) is a generic deep generative model for graph generation. *GraphGMG* (Li *et al.*, 2018) is a deep auto-regressive graph model that generates the vertices of a graph sequentially. *SMILES-LSTM* (Sundermeyer *et al.*, 2012) is an LSTM model that utilizes the SMILES representation. *GraphNVP* (Madhawa *et al.*, 2019) introduces self-normalizing flow in a molecule generative model. *GRF* (Honda *et al.*, 2019) and *GraphAF* (Shi *et al.*, 2019) employ an auto-regressive generation process. In *CGVAE* (Liu *et al.*, 2018b), both the encoder and decoder are graph-structured and enforce a validity constraint. We note that CGVAE shares a similar architecture with the proposed D-MolVAE models and also utilizes $\beta - VAE$. We utilize published default settings for each of these models.

Distribution Distance Metrics and Molecular Properties

A distribution of a variable of interest is computed from the training and the generated dataset to compare these datasets in terms of distances of distributions. Specifically, distributions are compared via MMD or KLD. When utilizing MMD, we focus on variables that are routinely used to summarize distributions of graphs (You *et al.*, 2018; Liu *et al.*, 2018a), such as *node degree*, *clustering coefficient*, or *average orbit count*. The latter counts the number of 4-orbits in a graph. When utilizing KLD, we focus on benchmark molecular properties in cheminformatics, such as cLogP, clogS, PSA, rPSA, Drug-likeness, and SA Score [2] [3]. Briefly, cLogP stands for computationally-predicted lipophilicity, most commonly referred to as logP. This represents the ratio at equilibrium of the concentration of a compound between two phases, an oil and a liquid phase. Lipophilicity is an important physicochemical parameter when developing new drugs, because it influences various pharmacokinetic properties, such as absorption, distribution, permeability, and routes of drugs clearance. cLogS, which stands for computationally-predicted logS, is directly related to the water solubility of a drug and is defined as a common solubility unit corresponding to the 10-based logarithm of the solubility of a molecule measured in mol/L. The polar surface area (PSA) and relative PSA (rPSA) are important evaluators in medicinal chemistry of a drug's ability to permeate cells. Drug-likeness, computed with RDKIT via QED, which stands for quantitative estimation of drug-likeness, is calculated as a geometric mean over individual descriptors that combine the desirability of a new drug over the underlying distribution of molecular properties in known drugs. SA Score stands for synthetic accessibility score and estimates the ease of synthesis of a drug.

[2] RDKit: Open-source cheminformatics; http://www.rdkit.org

[3] DataWarrior: Open-source molecules; https://openmolecules.org/datawarrior/

Correlation of Molecular Properties

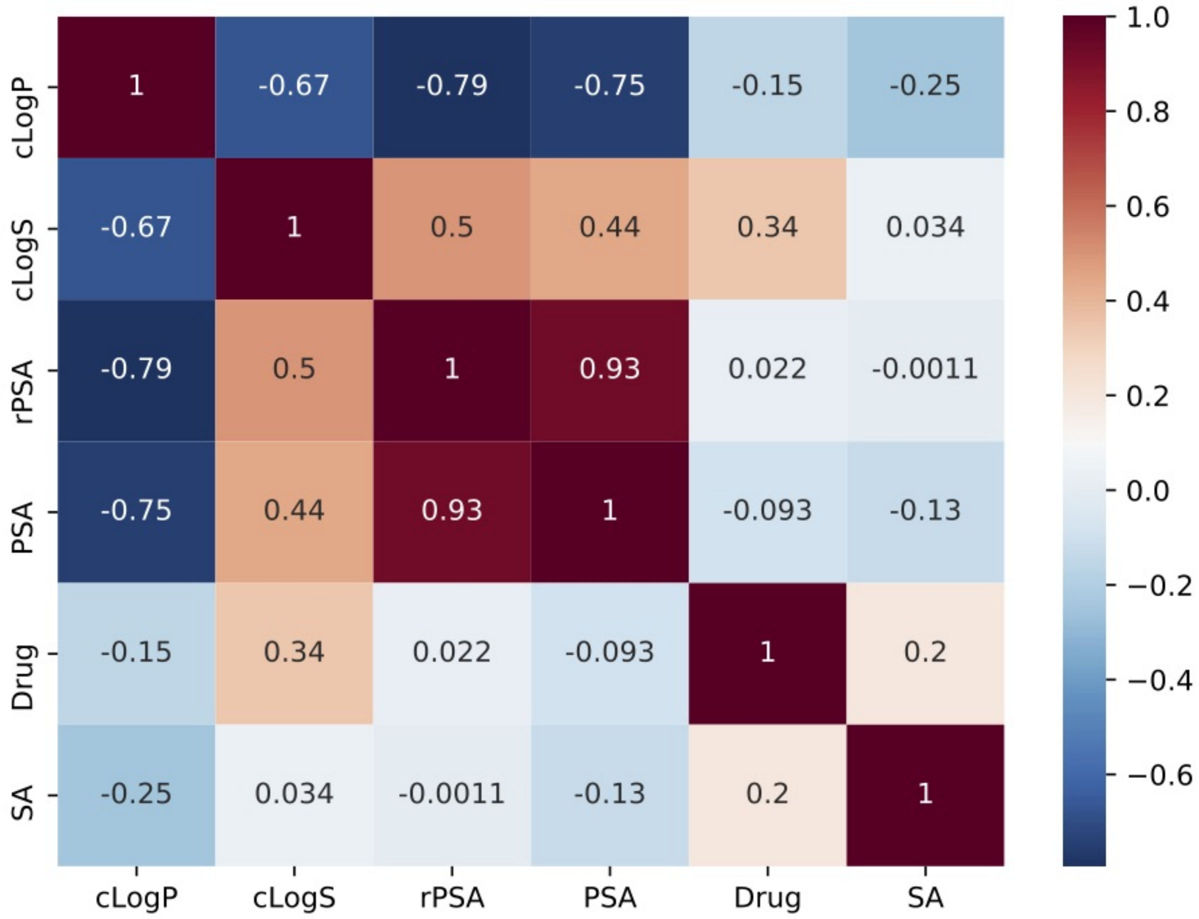


Fig. 1. Correlation heatmap between molecular properties is calculated over the QM9 training dataset. Pearson’s correlation is used.

Visual Comparison of Learned to Input Distributions

CGVAE

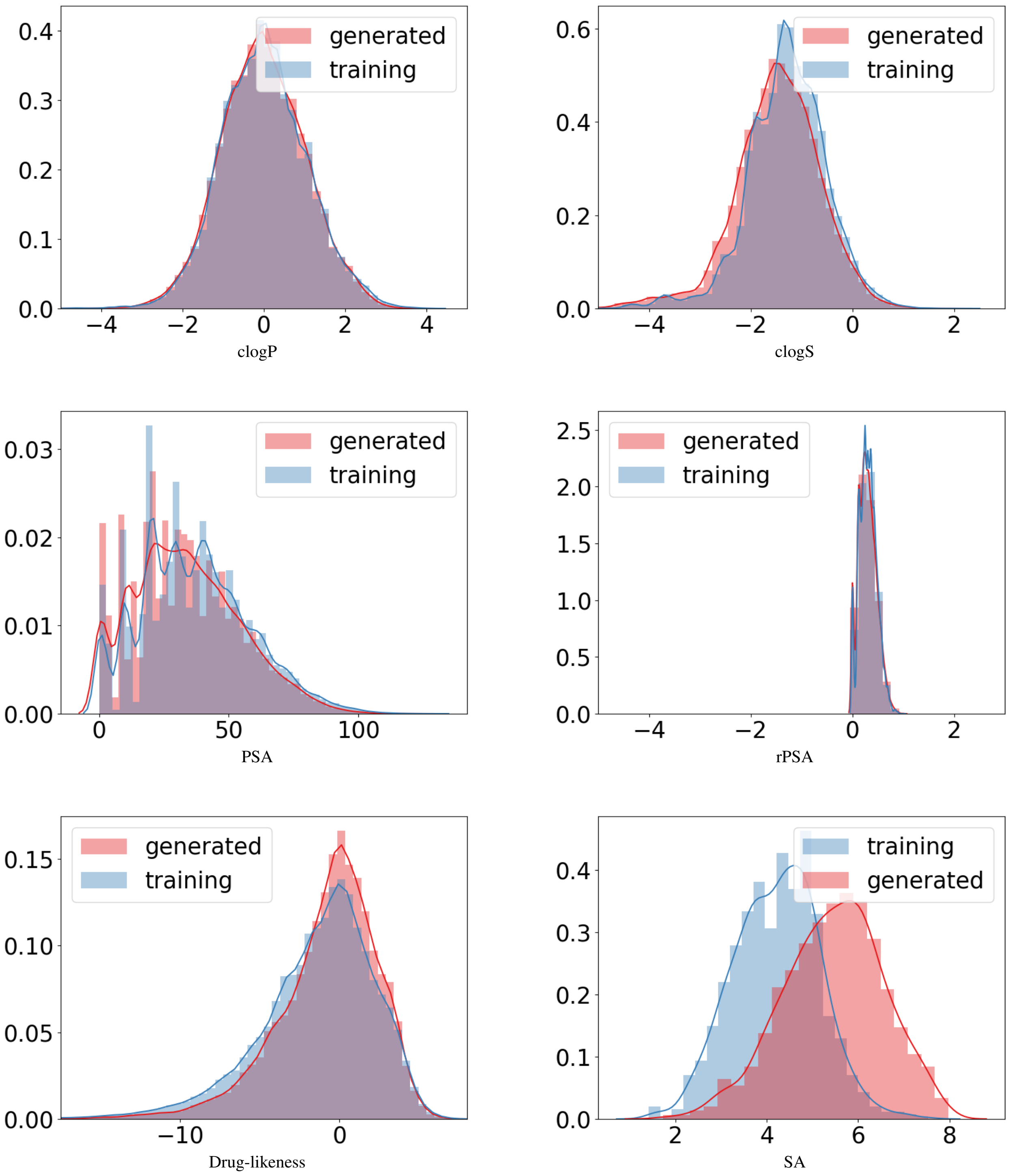


Fig. 2. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for CGVAE. Results are better seen in color.

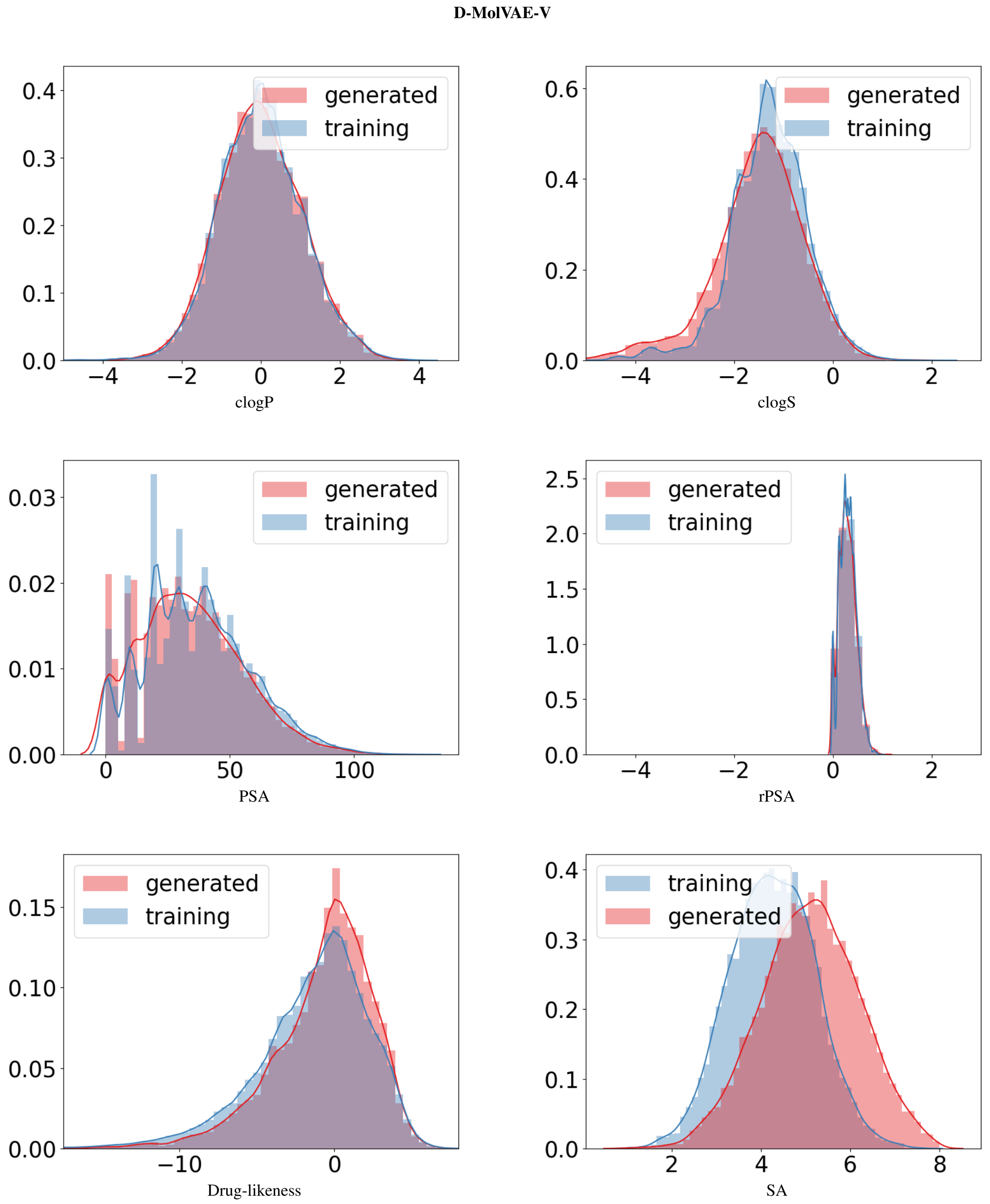


Fig. 3. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for D-MolVAE-V. Results are better seen in color.

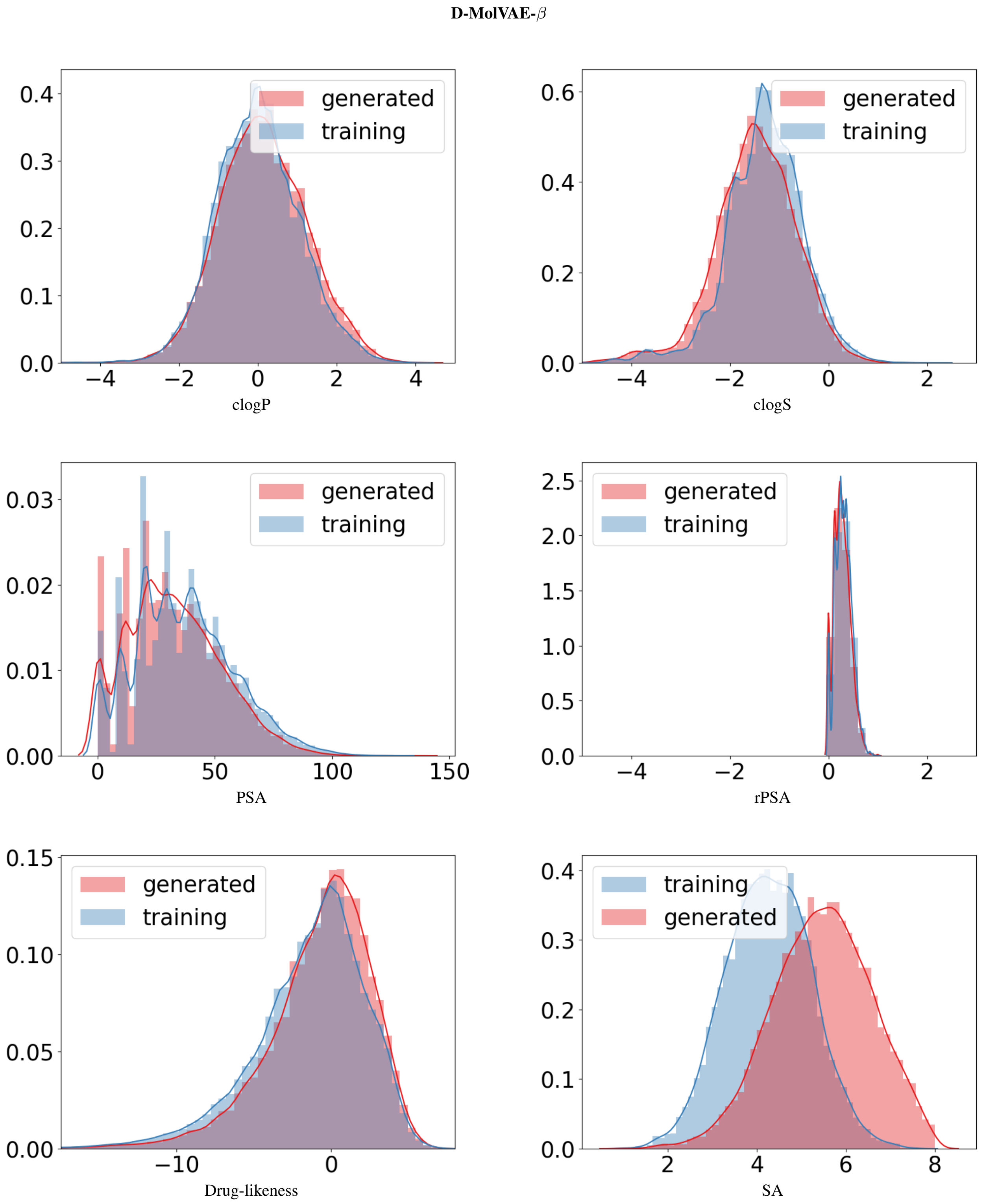


Fig. 4. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for D-MolVAE-β. Results are better seen in color.

D-MolVAE-DIP-I

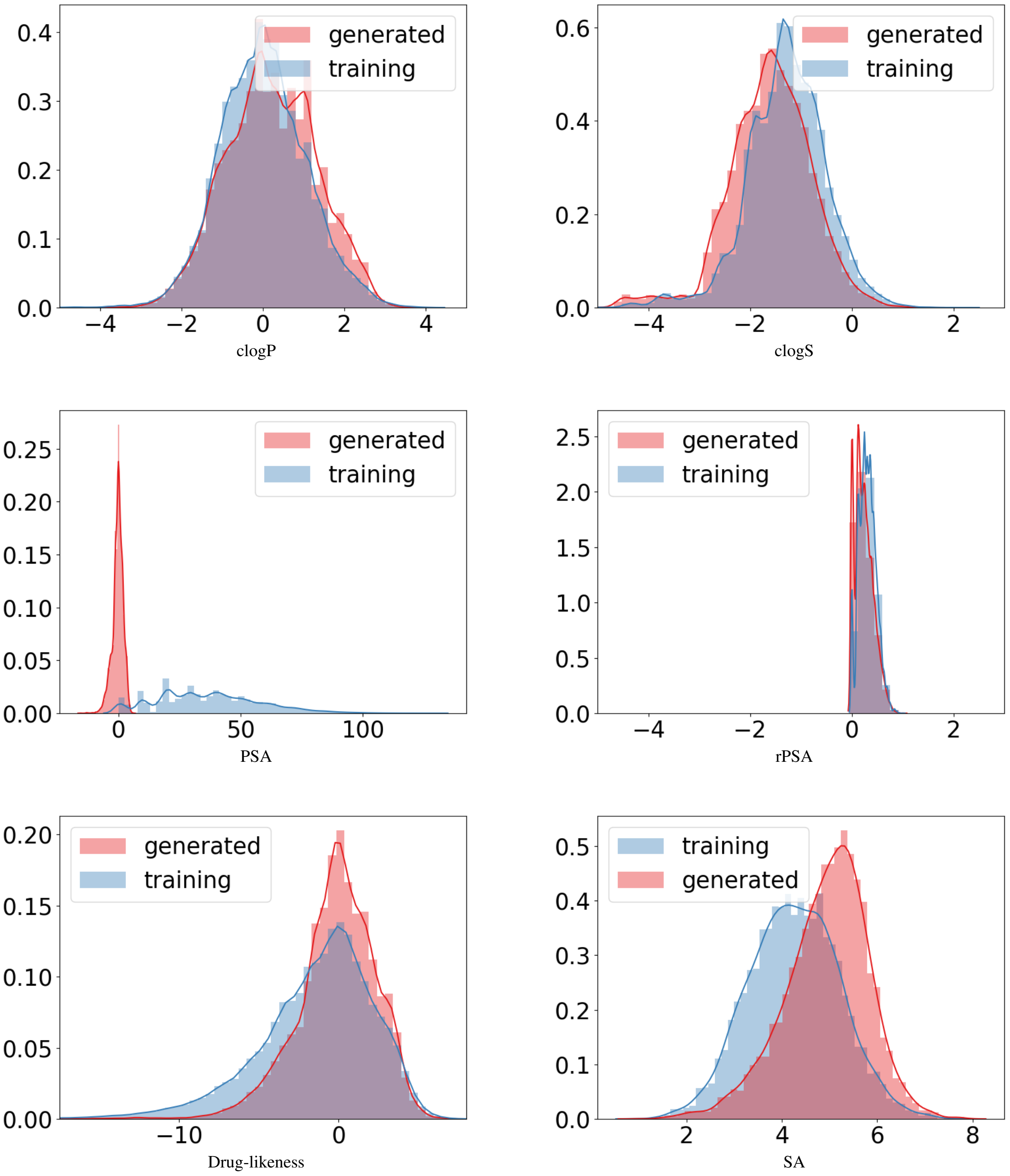


Fig. 5. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for D-MolVAE-DIP-I. Results are better seen in color.

D-MolVAE-DIP-II

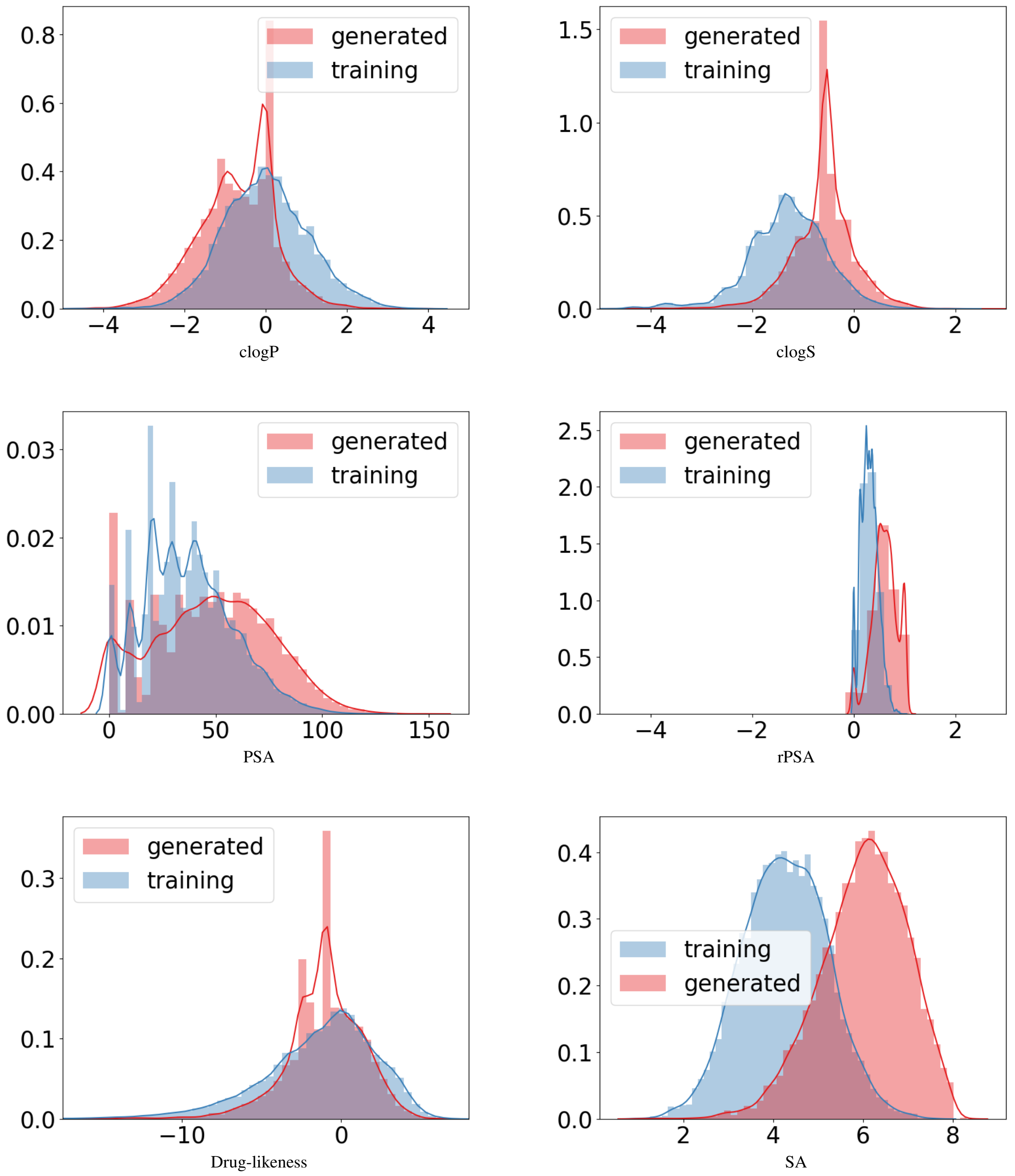


Fig. 6. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for D-MolVAE-DIP-II. Results are better seen in color.

D-MolVAE-VIB

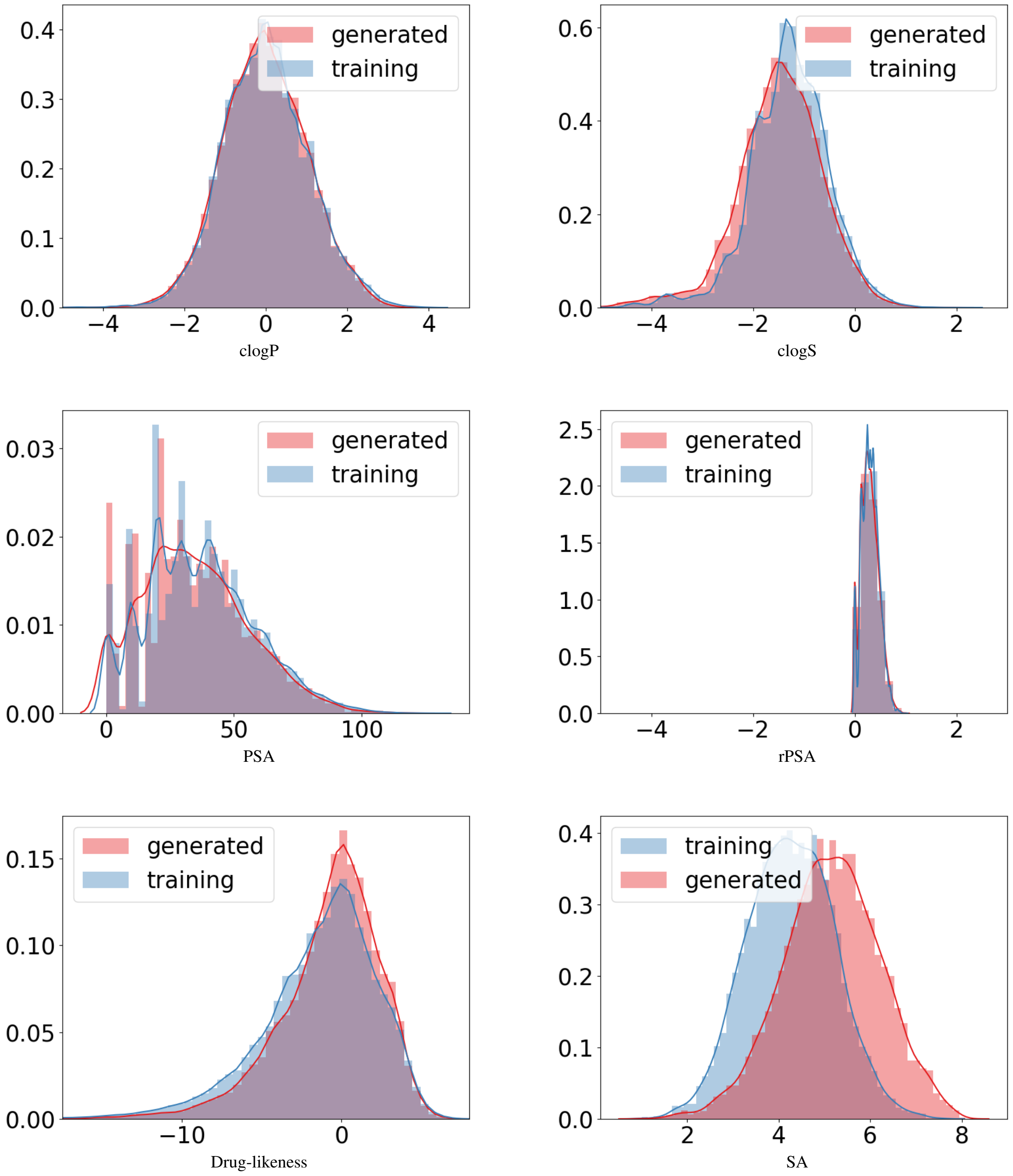


Fig. 7. Comparison of the distribution of cLogP, cLogS, PSA, rPSA, and drug-likeness in the generated versus the training dataset for D-MolVAE-VIB. Results are better seen in color.

Visualization of Selected Generated Molecules

We show some molecules generated from eac of the models trained on the QM9 dataset.

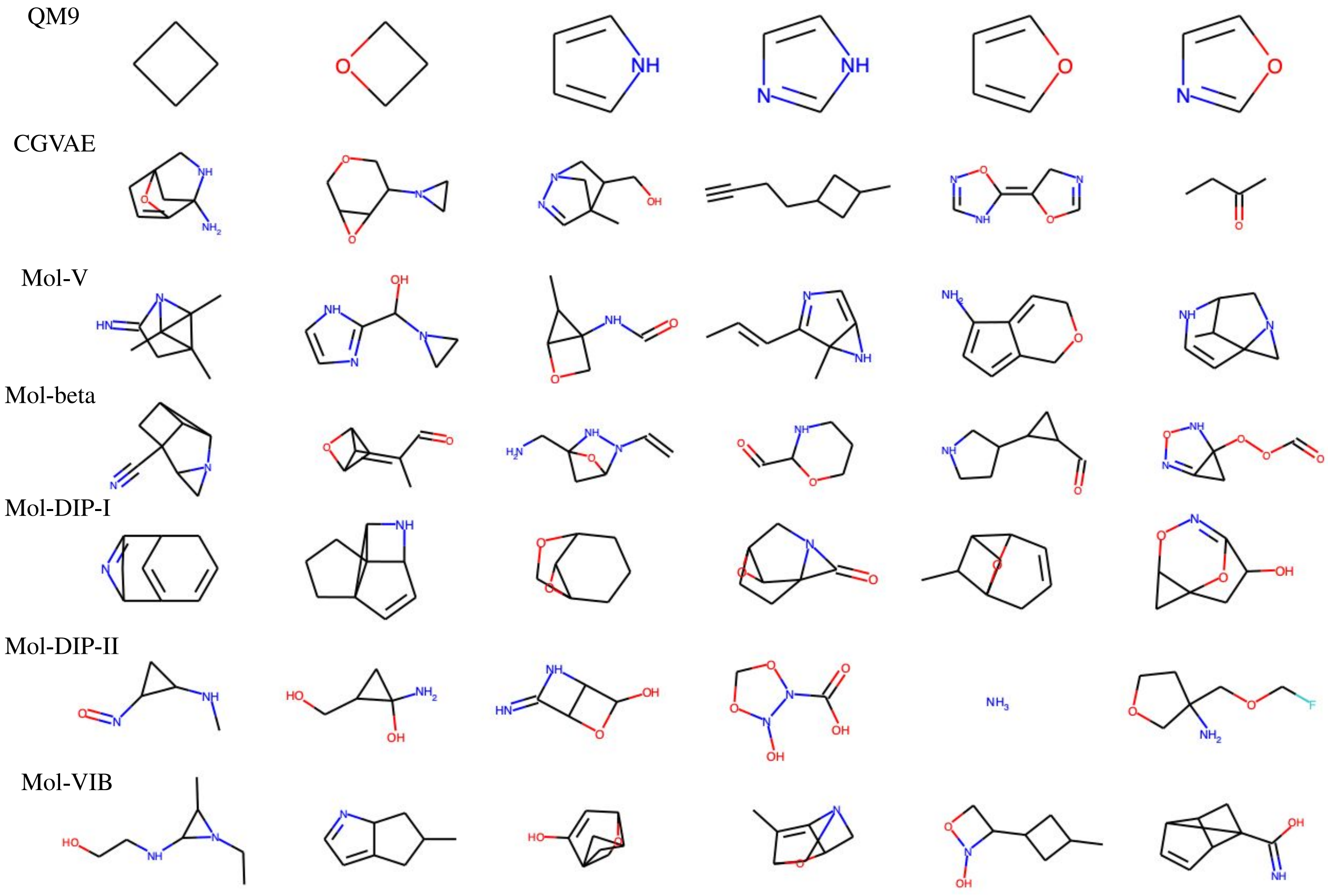
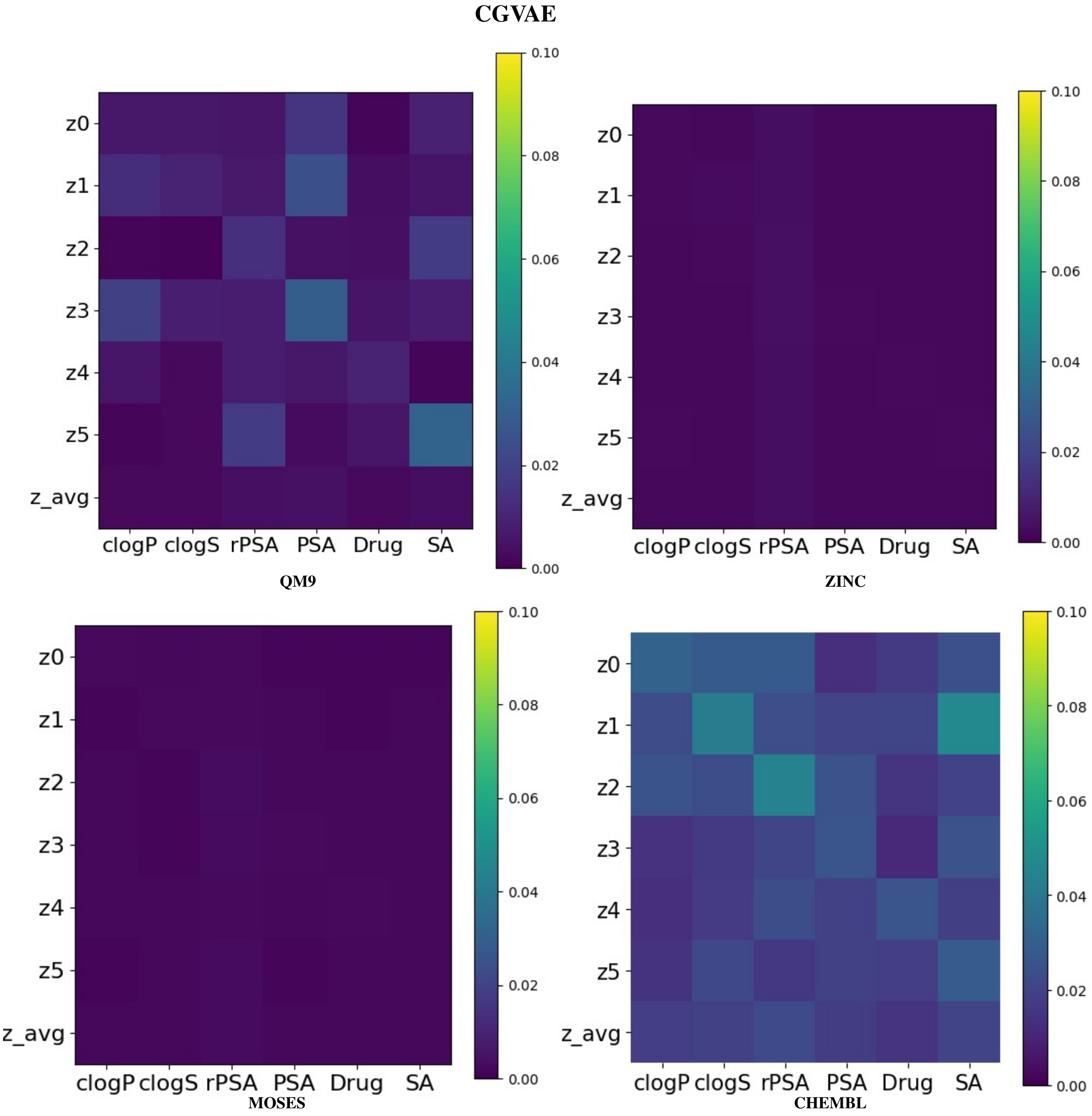


Fig. 8. Molecules randomly selected from the QM9 dataset are shown in the top row. The next row shows molecules sampled at random over those generated by CGVAE, MolVAE-V, MolVAE-β, MolVAE-DIP-I, MolVAE-DIP-II and MolVAE-VIB, repectively.

Correlation of Disentangled Factors with Molecular Properties

The mutual information is calculated between each of the disentangled factors learned by a model and the molecular properties computed on the molecules generated by the model.


CGVAE

QM9

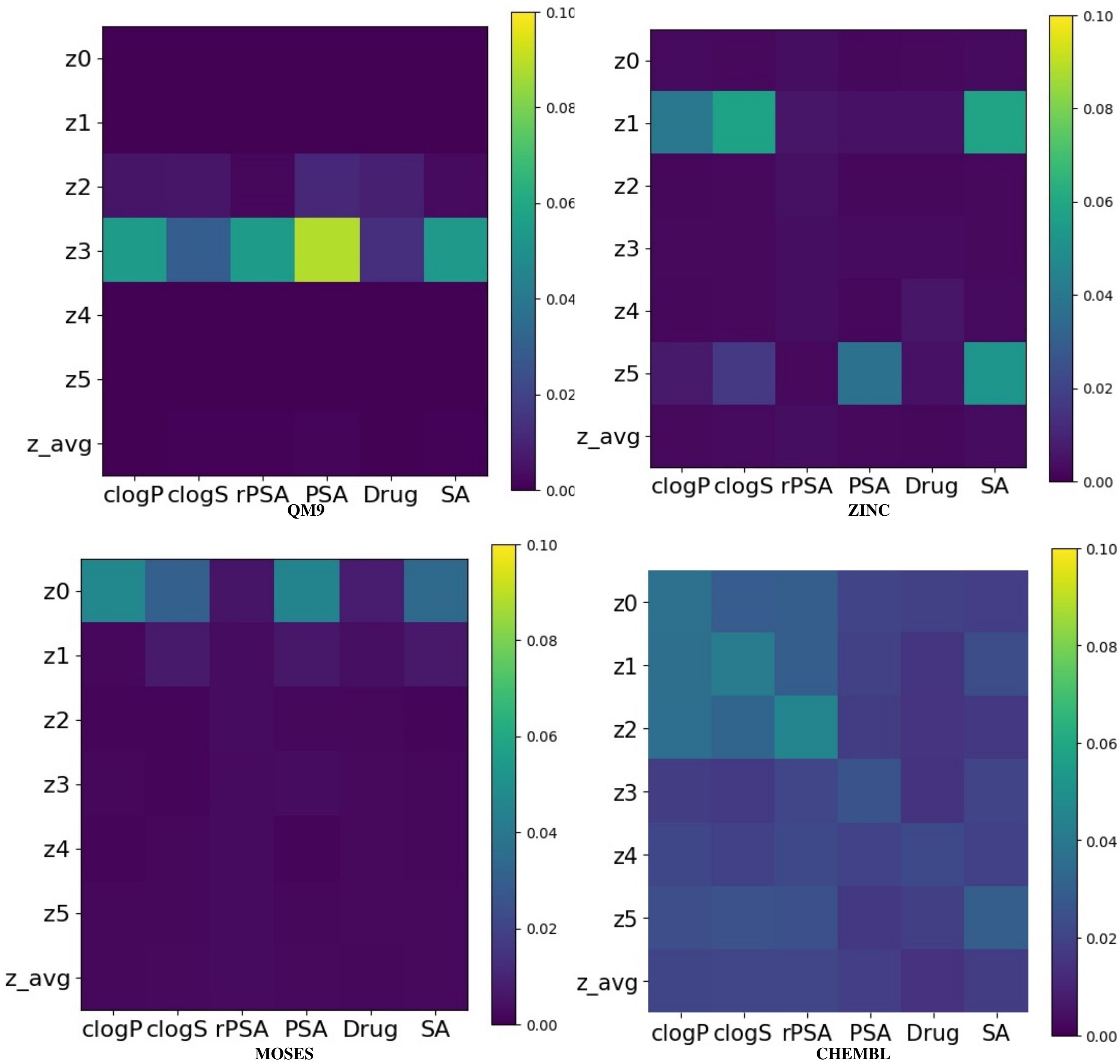
ZINC

MOSES

CHEMBL

Fig. 9. Heatmaps visualize the mutual information between each of the latent factors learned by CGVAE and the molecular properties computed on molecules generated by the model.

D-MolVAE-V



QM9

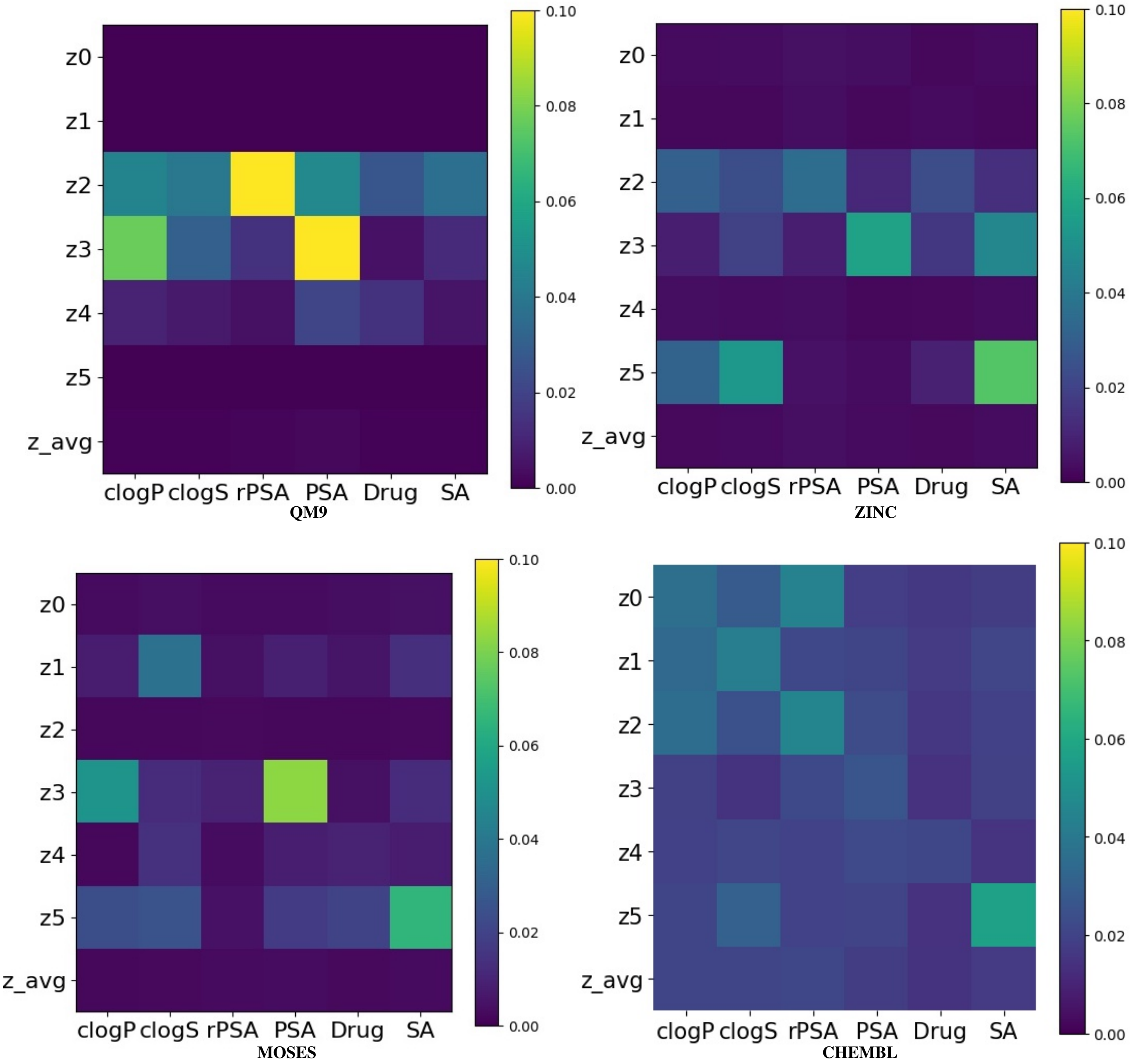
ZINC

MOSES

CHEMBL

Fig. 10. Heatmaps visualize the mutual information between each of the latent factors learned by D-MolVAE-V and the molecular properties computed on molecules generated by the model.

D-MolVAE-β



QM9

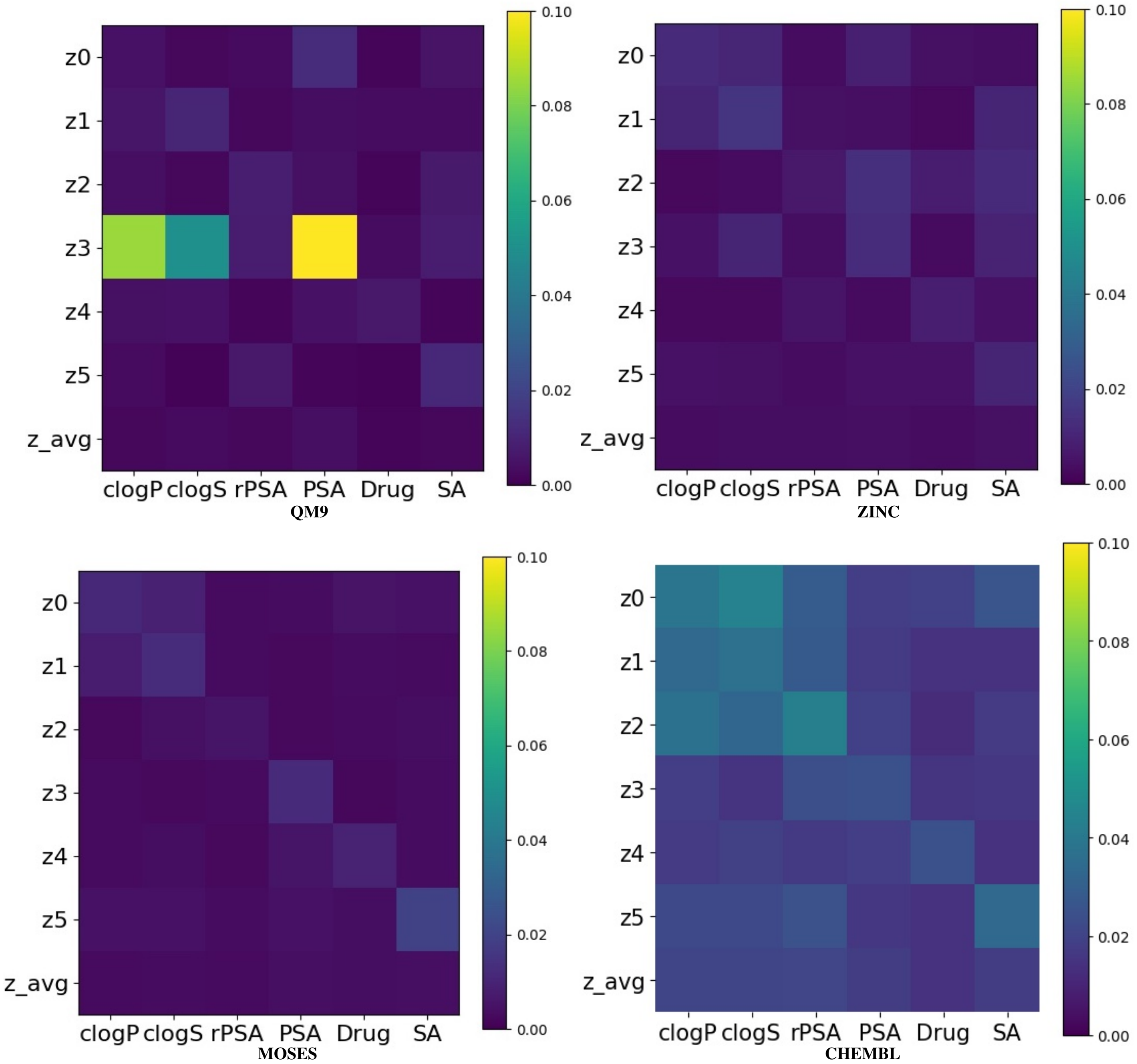
ZINC

MOSES

CHEMBL

Fig. 11. Heatmaps visualize the mutual information between each of the latent factors learned by D-MolVAE-β and the molecular properties computed on molecules generated by the model.

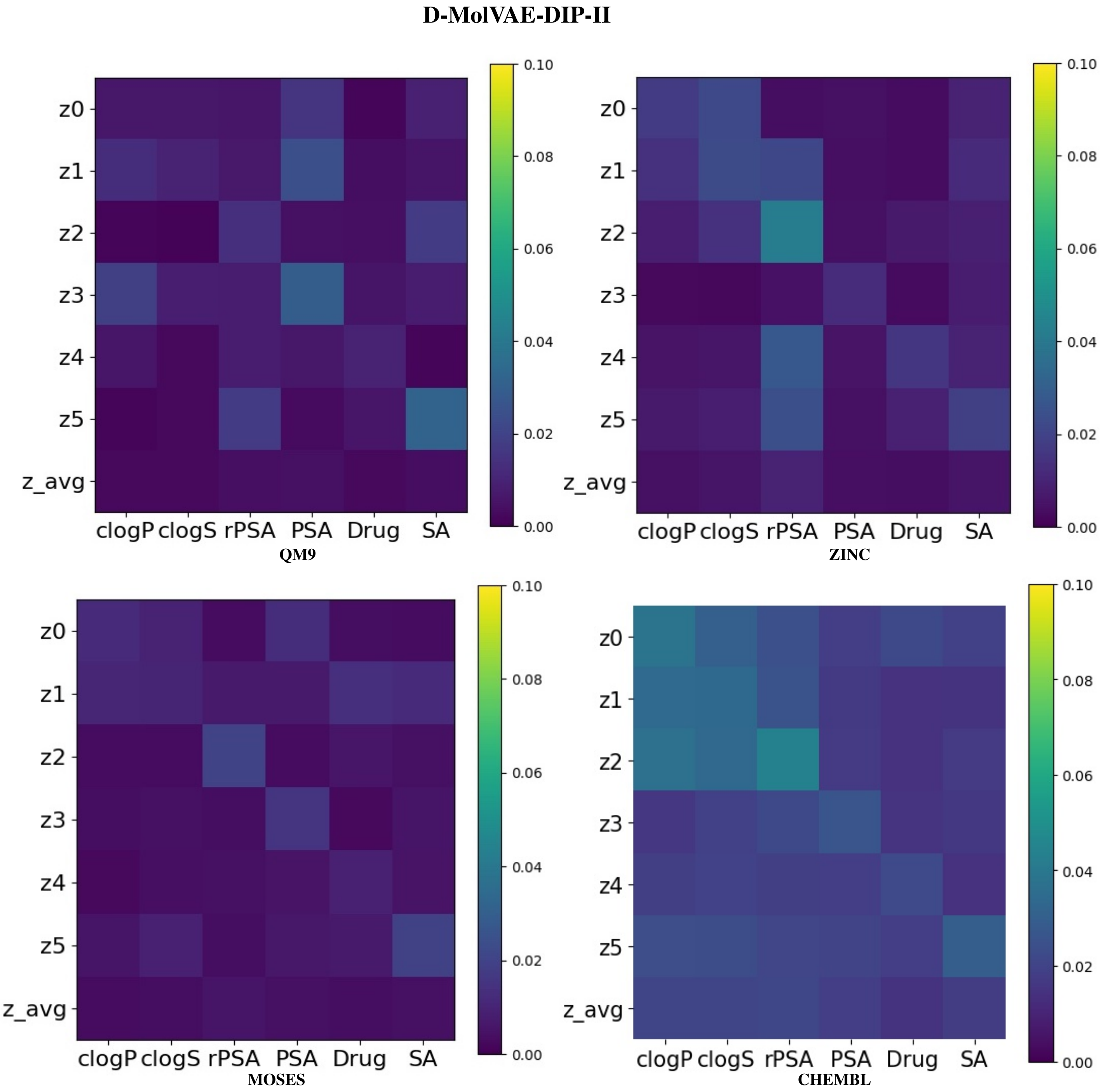
D-MolVAE-DIP-I



QM9 ZINC

MOSES CHEMBL

Fig. 12. Heatmaps visualize the mutual information between each of the latent factors learned by D-MolVAE-DIP-I and the molecular properties computed on molecules generated by the model.



QM9

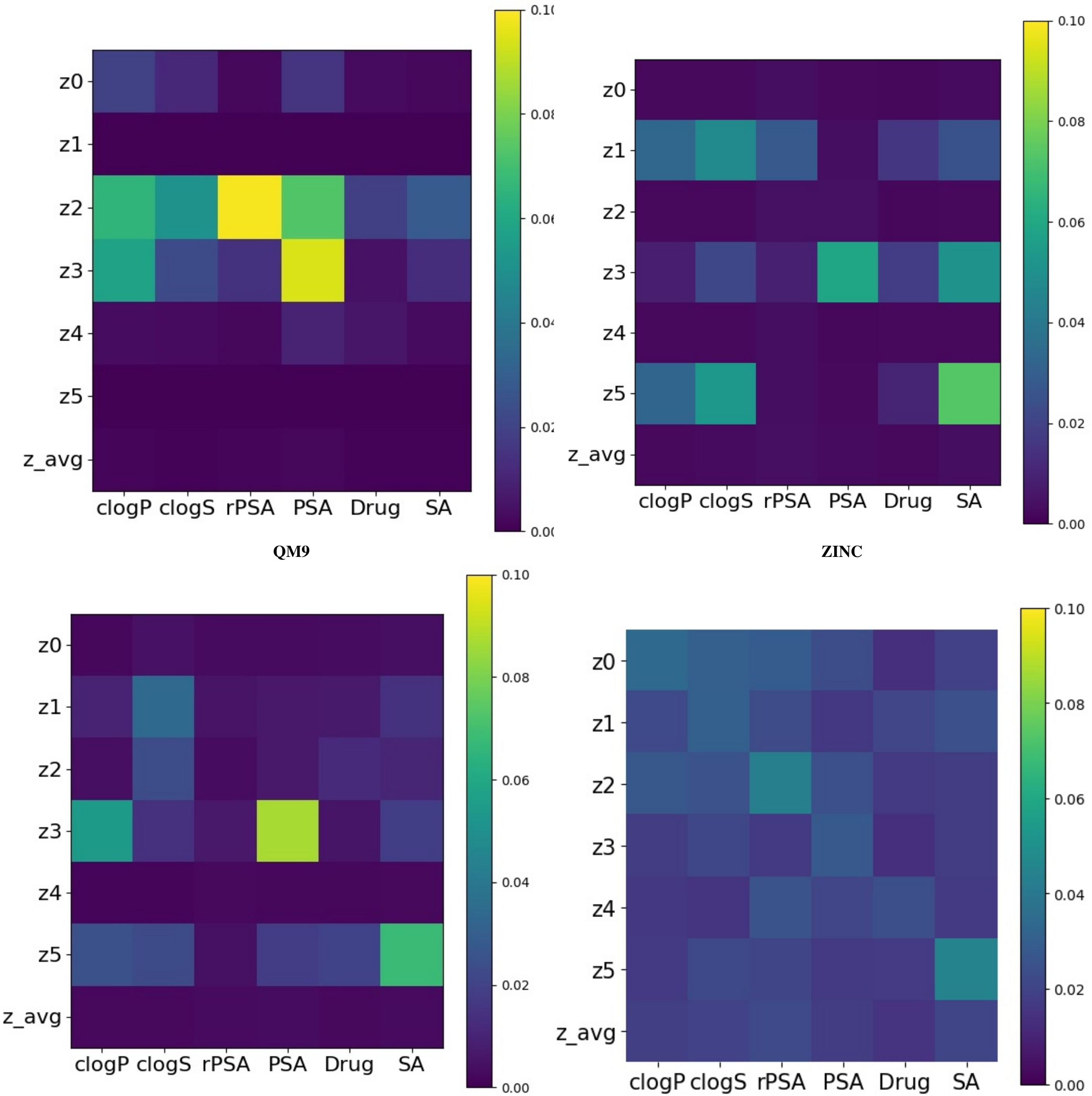
ZINC

MOSES

CHEMBL

Fig. 13. Heatmaps visualize the mutual information between each of the latent factors learned by D-MolVAE-DIP-II and the molecular properties computed on molecules generated by the model.

D-MolVAE-VIB



QM9

ZINC

MOSES

CHEMBL

Fig. 14. Heatmaps visualize the mutual information between each of the latent factors learned by D-MolVAE-VIB and the molecular properties computed on molecules generated by the model.