Deep Latent-Variable Models for Controllable Molecule Generation

Yuanqi Du
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
ydu6@gmu.edu

Yinkai Wang
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
ywang88@gmu.edu

Fardina Alam
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
falam5@gmu.edu

Yuanjie Lu
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
ylu22@gmu.edu

Xiaojie Guo
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
xguo7@gmu.edu

Liang Zhao
Dept. of Computer Science
Emory University
Atlanta, GA, USA
liang.zhao@emory.edu

Amarda Shehu
Dept. of Computer Science
George Mason University
Fairfax, VA, USA
amarda@gmu.edu

***Abstract*—In-silico molecular design is a challenging but important task to advance cheminformatics, drug discovery, biotechnology, and material science. Representation learning by increasingly-sophisticated deep generative models is opening a new avenue for small molecule generation in silico. Much of the research has focused on equipping such models with the ability to generate novel yet valid molecules, and graph-based variational autoencoders have yielded some of the best performance in this regard. Latest efforts have investigated disentangled representation learning as a way of additionally providing some interpretability. However, how to link chemical and biological space remains a key challenge that unsupervised representation learning cannot address. In this paper, we debut a graph-based variational autoencoder framework to address this challenge under the umbrella of disentangled representation learning. Specifically, the framework learns a disentangled representation and additionally permits several inductive biases that allow connecting the learned latent factors to molecular properties, such as drug-likeness, synthesizability, and many others. Extensive and rigorous comparative analysis on diverse benchmark datasets shows that the resulting models are powerful and open up an exciting line of research on controllable molecule generation in support of cheminformatics, drug discovery, and other application settings.**

***Index Terms*—Deep latent variable models, variational autoencoder, disentangled representation, molecule generation, controllable generation, supervised representation learning**

I. Introduction

In-silico molecular design is an important task to advance cheminformatics, drug discovery, biotechnology, and material science [**?**]. About 10^{60} drug-like molecules are estimated to be synthetically-accessible [**?**]. This is a vast chemical space that has traditionally presented challenges in silico [**?**]. For several decades, researchers have had to rely on incomplete domain-specific knowledge to construct molecular representations [**?**]. Advances in deep learning have lately renewed focus on representation learning directly from data [**?**].

In particular, the SMILES representation permitted addressing molecule generation as a string generation problem, but SMILE-based models fell short on generating valid molecules. The linear SMILES representation could not capture inherent chemical relationships in small molecules [**?**], [**?**], [**?**]. By leveraging a more expressive representation of a small molecule as a graph, graph-generative models proved more powerful [**?**]. In particular, graph-based variational autoencoder (VAE) models were shown to generate more valid molecules than the SMILES-based models [**?**].

Much of the research on deep latent variable models for small molecule generation has focused on equipping such models with the ability to generate novel yet valid molecules, and graph-based variational autoencoders have yielded some of the best performance in this regard [**?**], [**?**].

Latest efforts have investigated disentangled representation learning for understanding what aspects of molecular structure are controlled by the learned latent factors [**?**]. While disentangled representations enhance interpretability, work in [**?**] additionally shows that the disentanglement preserves the ability of these models to generate novel and valid molecules, often outperforming models that do not enforce disentanglement.

This work is supported in part by NSF Grant No. 1907805, 1900061, and 1763233.

However, currently, deep latent variable models for small molecule generation typically operate under the umbrella of unsupervised learning and so cannot link the chemical and biological space; that is, such models cannot control the generation towards molecules with specific biological properties.

This paper addresses the challenge of linking chemical and biological space for small molecule generation under the umbrella of supervised representation learning. We propose a graph-based VAE framework that implements inductive bias and allows us to obtain and compare various models for how they connect the learned latent factors to molecular properties, such as drug-likeness, synthesizability, surface accessibility, and more properties of interest. We evaluate the various models on three different benchmark datasets. The experiments demonstrate that the proposed framework is powerful and opens up an exciting line of research on controllable molecule generation in support of cheminformatics, drug discovery, and other application settings.

The rest of this paper proceeds as follows. Section I-A provides a concise review of related work on deep models for small molecule generation. Section II describes the proposed model in detail. The evaluation of the model is presented in Section III. The paper concludes with a summary of contributions and future directions of research in Section IV.

A. Related Work and Preliminaries

Machine learning expedited progress in in-silico small molecule generation, but shallow models proved ineffective at generating novel and valid molecules [**?**], [**?**], [**?**]. The SMILES representation [**?**], permitted the application of deep learning for molecule generation as string generation. SMILES is a linear representation of a molecule that stands for "molecular-input line-entry system." It is a formal grammar that denotes atom types and bond types by designated characters and symbols. Recurrent neural network (RNN)-based models were shown more powerful than shallow models [**?**], [**?**], [**?**], but they could generate few valid molecules. The SMILES representation could not fully capture the non-local constraints in the chemical structure of a small molecule.

To address the validity issue, researchers added explicit syntactic and semantic constraints [**?**], [**?**]. Others guided models through active learning, reinforcement learning, and additional training signals [**?**], [**?**]. This yielded some improvements, but generating valid molecules remained challenging.

Seminal work in [**?**] demonstrated the power of graph-based VAEs which represent a molecule as a graph with atoms as vertices and bonds as edges. The GraphVAE model opened the door to many other graph-based VAE models that were shown to significantly improve our ability to generate novel yet valid small molecules in silico.

State-of-the-art (SOTA) models for molecule generation leverage the VAE framework to: (1) encode: learn a low-dimensional, latent representation of a molecular graph; and (2) decode: learn to map the latent representation back into a molecular graph. GraphVAE [**?**] generates molecular graphs by predicting their adjacency matrices. Work in [**?**] proposes a constrained graph generative model that enforces validity by generating one atom at a time in a molecule. Other works encode the vertices into vertex-level embeddings and predict the edges between each pair of vertices to generate a graph [**?**], [**?**].

There is also another line of works leveraging flow-based generative models for molecule generation [**?**], [**?**]. However, the design of dimension partition and required same input and latent dimensions greatly limit the power of the latent space and prevent advances in discovering the learnt latent space for controls.

Most recently, disentangled representation learning has been debuted in small molecule generation. Inspired by the need for model transparency and interpretability, researchers in [**?**] evaluate a disentangled graph VAE for small molecule generation. They show that the disentangled factors result in similar or higher validity and novelty over generated molecules than graphVAE and other related SOTA models. In particular, they show that the disentangled factors control interesting aspects of molecular structure but are not powerful enough to control desirable molecular properties due to the unsupervised learning setting. In essence that is what we address in this paper. We demonstrate how one can leverage a disentangled graph VAE framework but additionally incorporate inductive bias to carry out supervised representation learning for *controllable molecule generation.* Before we describe the proposed framework and the models resulting from it, we summarize here the VAE framework in the interest of completeness.

B. Preliminaries: The VAE Framework

The encoder (inference model) and the decoder (generative model) are two connected but independently parameterized models in the VAE framework. These two models complement one another. The generative model receives an approximation to its posterior over latent random variables from the inference model, which it uses to update its parameters during an iteration of "expectation maximization" learning. The generative model acts as a form of scaffolding for the inference model to learn meaningful representations of the data.

1) Inference Model: The input to the encoder is a datapoint $x \in X$, and the encoder produces a hidden representation $z \in Z$ as an output; the encoder has weights and biases. It transforms the data into a latent (hidden) representation space Z by outputting two vectors, a mean vector Z_μ and a standard deviation vector Z_σ, which are of significantly fewer dimensions than the input dimensions. Because the encoder must learn an efficient compression of the data into this lower-dimensional region, commonly referred to as a "bottleneck," the VAE can sample z throughout a continuous space based on what it has learned from the input data by using the mean and standard deviation vectors.

2) Generative Model: The representation z is then fed into the decoder, which outputs the parameters of the probability distribution of the data; the decoder is also a neural network with its own weights and biases. VAEs assume that the input distribution inherently follows a distribution similar to the

normal/Gaussian distribution, so that the latent space regularization can be expressed quite naturally. To achieved this, two loss functions reconstruction loss and Kullback-Leibler (KL) divergence loss are optimized.

The reconstruction loss ensures that the output generated by the decoder is similar to the input. The KL divergence measures the divergence between two probability distributions. The latent vector z is sampled from Z_μ and Z_σ and unit normal distribution $\mathcal{N}(0,1)$. KL-divergence ensures that the latent-variables are close to the standard normal distribution. Specifically, the loss function $L = (reconstruction\_loss) + (regularization\_term) = \frac{1}{N}\sum_{n=1}^{N}|x-y|^2 + KL[G(Z_\mu, Z_\sigma), N(0,I)]$.

II. Methods

First, we represent a molecule as a graph $G = (\mathcal{V}, \mathcal{E}, E, F)$, where V is the set of N nodes (that is, the atoms) and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ is the set of edges (that is, the bonds connecting the atoms bonds) between pairs of nodes $\mathcal{V}$. $E \in \mathbb{R}^{N\times N\times K_1}$ refers to the edge features (bond type), where K_1 is the total number of bond types. $F \in \mathbb{R}^{N\times N\times K_2}$ refers to the node features (atom types), where K_2 is the total number of atom types. We additionally consider L common molecular descriptors $Y = \{y_1, y_2, ...y_L\}$, such as drug-likeness, molecular weight, synthesis accessibility, etc., as labels. These properties are described in some detail later in this section. Let us now summarize the deep latent-variable framework that works with the graph representation of a molecule before we describe in detail the various models that we build over this framework.

A. Framework Architecture

Following the molecular graph generation literature, the sequential generation process is essential for generating valid molecules. We adopt a similar model architecture as in [**?**]. During the generation process, we first initialize each node in a set of unconnected nodes as an initial step for the generative process. In each step, we start with a focus node and determine whether there is another node connecting to it. We first select an edge, then label edge, and then we update the nodes via message passing. We repeat the process for edge selection, edge labeling, and node update until a special stop node is selected. Then, we move to the next focus node in the loop, and the process repeats. We terminate when there is no candidate focus node in the connected sub-graph.

B. Deep Latent-Variable Framework

The deep latent-variable framework parameterizes VAEs to learn a joint distribution over a molecular graph G and desired properties Y, given a group of learned disentangled latent variables Z. The generative process is formulated as $p(G|Y,Z)$. The objective for a VAE model here is to learn to (i) encode a molecular graph into a continuous latent space with $p(Z,Y|G)$, as well as (ii) decode a molecule from the learned latent space with $p(G|Z,Y)$. In our exposition of this framework we will regularly refer to the illustration in Fig. 1.

We first illustrate this with a model that we treat as a baseline in this paper, β-VAE. The model has been published before [**?**], so we only summarize its most salient characteristics. The rest of the section then focuses on the novel VAE variants with inductive biases that allow us to control for desired properties. We refer to these models as *Conditional VAE*, abbreviated from now on as CondVAE, *Conditional Space VAE*, abbreviated as CSVAE, *Property-Controllable* VAE, abbreviated as PCVAE, and PCVAE-nsp, a variant of PCVAE without spectral normalization. In our exposition of these four models, we will regularly refer to Figure 1, which summarizes each of the models.

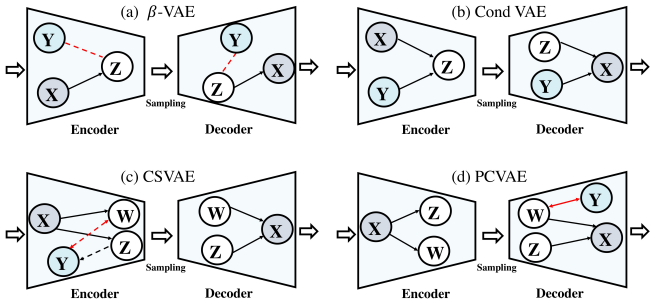

Fig. 1. Each sub-figure depicts the generative model (right) and its model inference (left). The enforcement of independence is shown by dotted red arrows, whereas the invertible dependence between two variables is represented by double arrows. Data is denoted by X and Z. W are two subsets of latent variables, and Y denotes the molecular properties.

C. β-VAE

The objective here is to learn a marginal probability distribution $p_\theta(G|Z)$ of a molecule G given a latent variable Z. However, learning $p_\theta(G|Z)$ requires the inference of its posterior distribution $p_\theta(Z|G)$, which, unfortunately, is intractable. Thus, one estimates the approximated posterior $q_\phi(Z|G)$ and minimizes the Kullback-leibler Divergence (KLD) between the true posterior and the approximated posterior $D_{KL}(q_\phi(Z|G)||p_\theta(Z|G))$. This is known as maximization of the evidence lower-bound (ELBO) [**?**]:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(Z|G)}[\log p_\theta(G|Z)] - \beta D_{KL}(q_\phi(Z|G)||p(Z)) \quad (1)$$

In Equation 1, θ and ϕ are learned; q_ϕ and the prior distribution $p(Z)$ are each assumed to be an isotropic Gaussian distribution, where the latent variables Z are independent of one another. The hyperparameter β controls the disentanglement ability of the VAE model [**?**]; a larger β value enforces more disentanglement. The β-VAE model does not guarantee that any latent variable $z \in Z$ exposes any molecular property $y \in Y$; this is indicated schematically (by the red dotted arrow between Z and Y) in the top left panel in Fig. 1. That is the reason we use this popular model as a baseline in our evaluation in this paper.

D. Conditional VAE (CondVAE)

One can easily modify β-VAE to explicitly control for desired molecular properties. We do so in what we refer to

as CondVAE by incorporating ground truth properties in the training process and selecting one-to-one pairs between latent variables $z \in Z$ and molecular properties $y \in Y$. To achieve this objective, we add an L_2 norm/loss which enforces the value of one latent variable z to be the same as one specific molecular property y of a molecule. Specifically, the objective is updated as follows:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(Z|G,Y)}[\log p_\theta(G,Y|Z)] - \beta D_{KL}(q_\phi(Z|G,Y)||p(Z)) - \sum_1^L ||z_l - y_l||^2 \quad (2)$$

As Equation 2 shows, the last term allows to specify controls over the latent space. Note that in this model, the number of latent variables is set to the number of desired molecular properties. Also note the dependence now between Z and Y in the top right panel of Fig. 1. As our results show later in this paper, this simple conditional VAE, while adding supervised control over the latent space, is not as effective as the other two models we propose next.

E. Conditional Subspace VAE (CSVAE)

CSAVE incorporates a new inductive bias. As the bottom panel of Fig. 1 shows, a new "latent subspace" W is introduced between the semantic space (the molecular properties) and the latent space. This model is inspired by work in [**?**]. We leverage this inductive bias here for controllable molecule generation, where we design a latent subspace W, a molecular property space Y, and a regular latent space Z. We aim to keep the information separate; the model tries to separate the latent variable corresponding to a desired molecular property from latent variables corresponding to unspecified molecular properties in the two latent spaces W and Z. In the original CSVAE model in [**?**], the semantic property can only be binary. Here, we add a linear mapping from a molecular property $y \in Y$ to the latent subspace W. To enforce this, we minimize the mutual information between the latent space Z and the molecular properties Y. The objective becomes:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(Z,W|G,Y)}[\log p_\theta(G,Y|Z,W)] - \beta D_{KL}(q_\phi(Z,W|G,Y)||p(Z)) - I(Y;Z) \quad (3)$$

In Equation 3, $I(Y;Z)$ denotes the mutual information between molecular properties Y and latent space Z. In practice, we treat mutual information minimization as an adversarial component in our model.

F. Property-controllable VAE (PCVAE)

An alternative approach to linking the latent space to a semantic space is proposed in [**?**]; specifically, a mutual dependency is enforced between the latent space W and the semantic properties Y. We adopt this idea here by designing an invertible ResNet [**?**] that enforces mutual dependency between W and Y. Note the double arrow between these two in the bottom right panel in Fig. 1. To learn an invertible function $f(w;y)$, we decompose the function $f(w;y)$ as $f(w;y) = \overline{f}(w;y) + w$. As proved by work in [**?**], the sufficient condition for function f to be invertible is $Lip(\overline{f}) < 1$, where $Lip(\overline{f})$ is the Lipschitz-constant of $\overline{f}(w;y)$. So, the objective becomes:

$$\max_{\theta,\phi} \mathbb{E}_{q_\phi(Z,W|G,Y)}[\log p_\theta(G,Y|Z,W)] - \beta D_{KL}(q_\phi(Z,W|G,Y)||p(Z)) \quad (4)$$

In practice, we utilize the Multi-layer Perceptrons (MLP) to model $\overline{f}$. As the function is the composition of a linear layer and nonlinear activation functions (e.g. ReLU), we have $Lip(\overline{f}) < 1$ if $||H||_2 < 1$, where H is the weight in all the MLP layers, and $||\cdot||_2$ denotes spectral normalization, as introduced in [**?**].

G. PCVAE without Spectral Normalization (PCVAE-nsp)

Following the architecture in [**?**], PCVAE-nsp serves as a baseline model for PCVAE, without spectral normalization.

III. Results

A. Experimental Setup

We train each of five models, β-VAE, CondVAE, CSVAE, and PCVAE, separately on three benchmark datasets that we describe below. The evaluation utilizes benchmark metrics, which we described next.

1) Benchmark Datasets: We adopt three standard datasets: QM9, ZINC, and MOSES. QM9 [**?**], [**?**] contains $\sim$134k stable small organic molecules with up to 9 heavy atoms (e.g. Carbon (C), Oxygen (O), Nitrogen (N), and Fluorine (F)). ZINC [**?**] contains about $250,000$ purchasable compounds, each with 23 heavy atoms on average. MOSES [**?**] contains about 1.9 million molecules, each with up to 30 heavy atoms. We use 120K/13K as training/validation set for QM9, 60K/10K as training/validation set for ZINC, and 30K/5K as training/validation set for MOSES.

2) Molecular Properties: We cast a wide net over cheminformatics literature and compile a list of 6 molecular properties: cLogP, cLogS, PSA, SA, Weight, and Drug-likeness [1] [2]. While more detailed information can be found in the above resources, we summarize here each of these properties. The 'c' in front of properties, such as cLogP and cLogS stands for "computationally-predicted/computed." These properties are computed over a molecule. cLogP, for instance, computes lipophilicity and is the ratio at the equilibrium of the concentration of a compound between two phases, an oil and a liquid phase. Lipophilicity is a critical physicochemical parameter when developing new drugs, because it influences various pharmacokinetic properties, such as the absorption, distribution, permeability, and routes of clearance of a candidate drug. cLogS stands for computed logS and it measures the water solubility of a drug. PSA stands for polar surface area and is an important evaluator of a drug's ability to permeate cells. SA stands for synthetic ability and estimates our ability to synthesize a molecule in the wet laboratory. Weight measures molecular weight; smaller molecules are desirable. Finally, Drug-likeness, computed with RDKIT via QED, stands for

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

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

quantitative estimation of drug-likeness. It 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.

3) Evaluation Metrics: Each of the trained models (on each of the three datasets) is used to generate 30K molecules. (1) The quality a generated dataset is evaluated in section III-B via 3 benchmark metrics: *Novelty*, *Uniqueness*, and *Validity*. Validity measures the fraction of generated molecules that are chemically valid. 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. (2) The quality of controllability is evaluated in various ways. First, we compute the mutual information between the learned latent variables and molecular properties and visualize it via a heatmap. Second, we evaluate the property prediction accuracy which exposes how the latent variables control the molecular properties. Third, we visualize how a model can control a specific molecular property. Fourth, we evaluate how the latent variable controls the molecular properties in a more general setting, where we sample 100 molecules and attempt to control the molecular properties of the generated molecules in a list of specified values (i.e. in practice, we use the highest density region in the molecular property distribution). We now relate these experiments in greater detail.

B. Validity, Novelty, Uniqueness of Generated Molecules

In Fig. 2 we draw some molecules sampled at random over the $30K$ molecules generated by each model on each of the three datasets. The quality of generated molecules is evaluated via the three metrics described above, as shown in Table I.

Table I allows making several observations. First, all five models are powerful and generate 100% chemically-valid molecules (on each of the datasets). Performance on novelty and uniqueness varies. CSVAE is the worst-performing model on uniqueness on all datasets. On the QM9 dataset, β-VAE outperforms all models on uniqueness. On the ZINC dataset, all five models (CSVAE excluded) perform similarly on uniqueness. On the MOSES dataset, only CondVAE drops from the list of similar-performing models on uniqueness. On novelty, all models are very close in performance to one another, above 99.9% on both the ZINC and MOSES dataset. Slightly higher variation is observed on novelty on the QM9 dataset, but β-VAE, PCVAE-nsp, and PCVAE are all very close to one another in performance. The main takeaway from these results is that the inductive bias does not hurt the performance of the models with the exception of CondVAE; in this model, the naive linking of the chemical and biological space restricts the diversity of generated molecules.

C. MI between Latent Variables and Molecular Properties

We visualizing the MI between latent variables and molecular properties. We do so on the QM9-trained models in the interest of space (other settings show similar results). Fig. 3 shows that the baseline $\beta - VAE$ model rarely learns

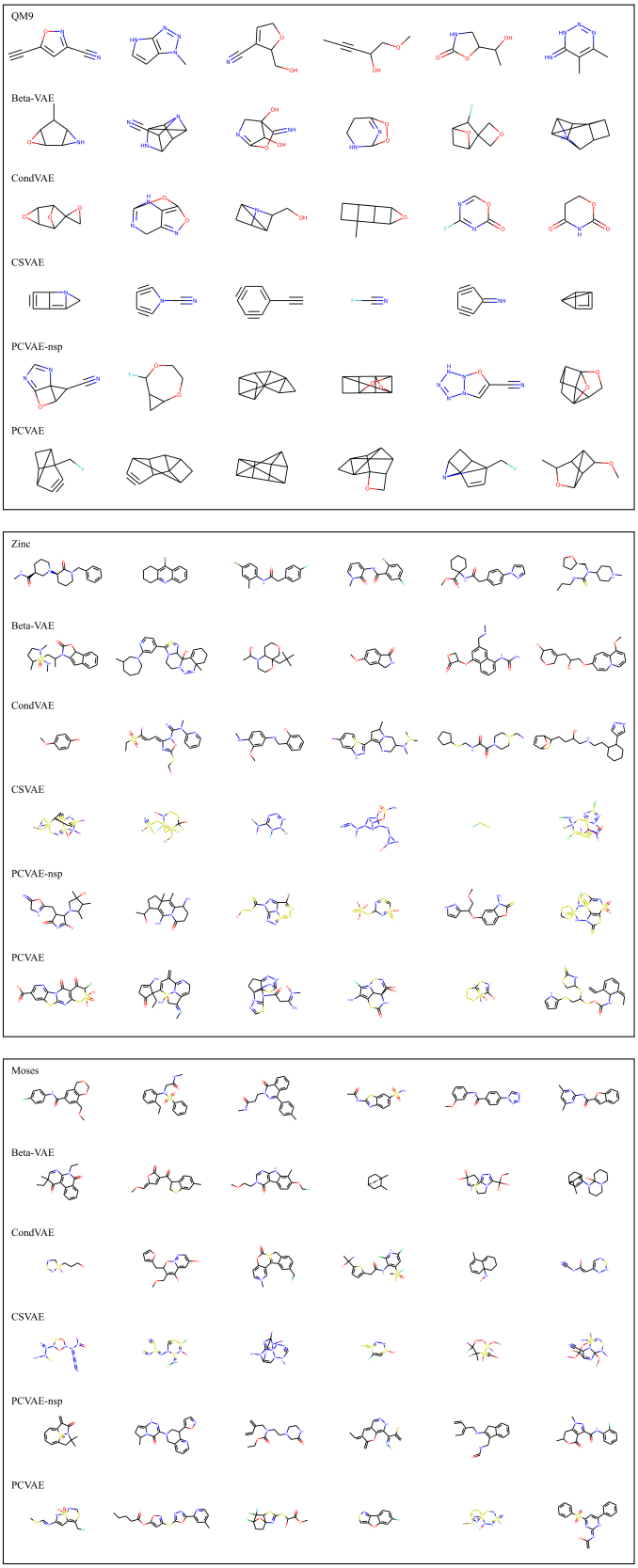

Fig. 2. We draw here molecules randomly selected from the 30K molecules generated from each of the models, trained on each of the three datasets – (top) QM9, (middle) ZINC, and (bottom) Moses – are drawn here.

TABLE I

NOVELTY, UNIQUENESS, AND VALIDITY, SHOWN IN %, ARE MEASURED ON A GENERATED DATASET. THE HIGHEST VALUE ACHIEVED ON A METRIC IS HIGHLIGHTED IN BOLDFACE.

	QM9			ZINC			MOSES		
Model	Validity	Novelty	Uniqueness	Validity	Novelty	Uniqueness	Validity	Novelty	Uniqueness
β-VAE	**100.00%**	98.23%	**99.28%**	**100.00%**	**100.00%**	**99.78%**	**100.00%**	99.92%	**99.88%**
CondVAE	**100.00%**	92.60%	90.00%	**100.00%**	99.98%	98.02%	**100.00%**	99.98%	93.30%
CSVAE	**100.00%**	97.01%	27.41%	**100.00%**	**100.00%**	42.72%	**100.00%**	**100.00%**	54.28%
PCVAE-nsp	**100.00%**	**98.57%**	86.94%	**100.00%**	**100.00%**	99.74%	**100.00%**	99.90%	99.80%
PCVAE	**100.00%**	97.43%	88.24%	**100.00%**	**100.00%**	99.48%	**100.00%**	99.96%	98.62%

correlations between the latent variables z and the molecular properties p. CondVAE performs similarly, suggesting that its control mechanism is not effective. Higher MI values are observed for CSVAE, PCVAE, and PCVAE-nsp. For CSVAE, more latent variables participate, but with weak control.

D. Evaluation on Property Prediction

As we introduce labels for desired properties in a supervised setting, our models offer another benefit, learning a property predictor. We now compare the models on property prediction accuracy, which implies learned controllability because we expect the predicted property value to be the same as the property value during the model training. We add a layer to minimize an MSE loss between one latent variable and a molecular property. We do so for each of the models and each of the properties. Table II relates the results. It shows that several of the properties can be predicted well by all the proposed models, with the exception of Weight and PSA. The two models that are able to consistently do well across all 6 properties and on all three datasets are PCVAE-nsp and PCVAE.

TABLE II

THE MSE BETWEEN ONE LATENT VARIABLE AND EACH MOLECULAR PROPERTY IN A SUPERVISED SETTING. THE BEST VALUE PER ROW IS HIGHLIGHTED IN BOLDFACE.

Dataset	Method	MSE					
		cLogP	cLogS	Drug	Weight	PSA	SA
QM9	β-VAE	2.54	2.34	15.59	15113.88	1691.80	19.03
	CondVAE	1.02	2.21	15.68	13865.48	1599.08	17.26
	CSVAE	**0.89**	**0.61**	13.88	53.72	407.72	**0.87**
	PCVAE-nsp	1.46	1.60	**3.40**	**35.80**	**7.58**	1.96
	PCVAE	1.39	1.55	8.93	36.25	13.36	1.87
ZINC	β-VAE	5.85	13.31	30.37	114717.69	6228.77	10.13
	CondVAE	5.72	12.88	28.88	110894.66	6045.01	9.89
	CSVAE	**2.04**	**1.94**	**28.76**	112158.86	5562.85	**0.69**
	PCVAE-nsp	3.03	2.55	29.65	**349.74**	**0.15**	1.72
	PCVAE	2.97	2.61	29.66	422.27	92.34	1.70
MOSES	β-VAE	7.93	14.38	28.89	94736.52	7168.32	5.61
	CondVAE	6.02	13.36	21.12	85160.43	5633.47	4.95
	CSVAE	7.71	14.21	**14.89**	94526.97	7157.01	4.82
	PCVAE-nsp	1.65	**2.04**	15.44	**8.73**	13.13	**1.17**
	PCVAE	**1.63**	2.04	15.46	10.24	**8.68**	1.19

E. Evaluation on Property Control

In Fig 4, we visualize how a model allows controlling the molecular properties of generated molecules. We illustrate this for the cLogS property with the increasing value of a latent variable z. Specifically, we set out to control the cLogS property of generated molecules to be $[-2, -1.5, -1, -0.5, 0, 0.5]$, respectively. The premise is that the four models will show different levels of controllability. We visualize this in Fig. 4.

The top panel in Fig. 4 shows results from β-VAE, which serves as a baseline. It is evident that the cLogS properties of generated molecules appear randomly drawn from the specified range, suggesting that control is rarely observed. The second panel shows results for CondVAE, which demonstrates a partial monotonic relationship in the first three generated molecules but not so for the last three molecules. This indicates that, while CondVAE may provide some improvement, the control is not strong enough. Similar variability is observed in the results from CSVAE, with partial control (on the last three molecules). Fig. 4 shows that PCVAE achieves superior performance in this qualitative evaluation. The model learns a monotonic relationship; an increasing value of z relates very closely here with an increase in the value of cLogS.

In Table III, we evaluate quantitatively how effective the control is overall 6 molecular properties. In the interest of space, we focus on the QM9 dataset. We repeat a similar experiment as in Fig. 4 100 times for each property and carry out statistical analysis on it. Specifically, for each property, we generate 100 molecules with properties specified in the range with the highest density in the molecular property distribution (as observed over the training dataset). We report the discrepancy via MSE between the properties of molecules generated in this manner and the expected properties predefined within the highest property density region. Again, $\beta - VAE$ serves as a baseline, since it is unsupervised.

The results shown in Table III support several observations. Even though CondVAE does not work as well as other proposed models, it outperforms $\beta - VAE$. This is not surprising, as there is more control specified in the model. However, it is evident that simply incorporating an MSE-based based constraint does not work well. Overall, the top two performing models are CSVAE and PCVAE. $PCVAE_nsp$, which serves as a baseline for PCVAE, performs slightly worse than CSVAE for most properties. PCVAE achieves the best overall performance in this task. In summary, Table III confirms many of our observations; namely, that weight and PSA are difficult properties, perhaps not quite captured by the molecular graph representation. The results show, however, that the lowest MSE in each of the molecular properties is obtained by one of our proposed models for property control.

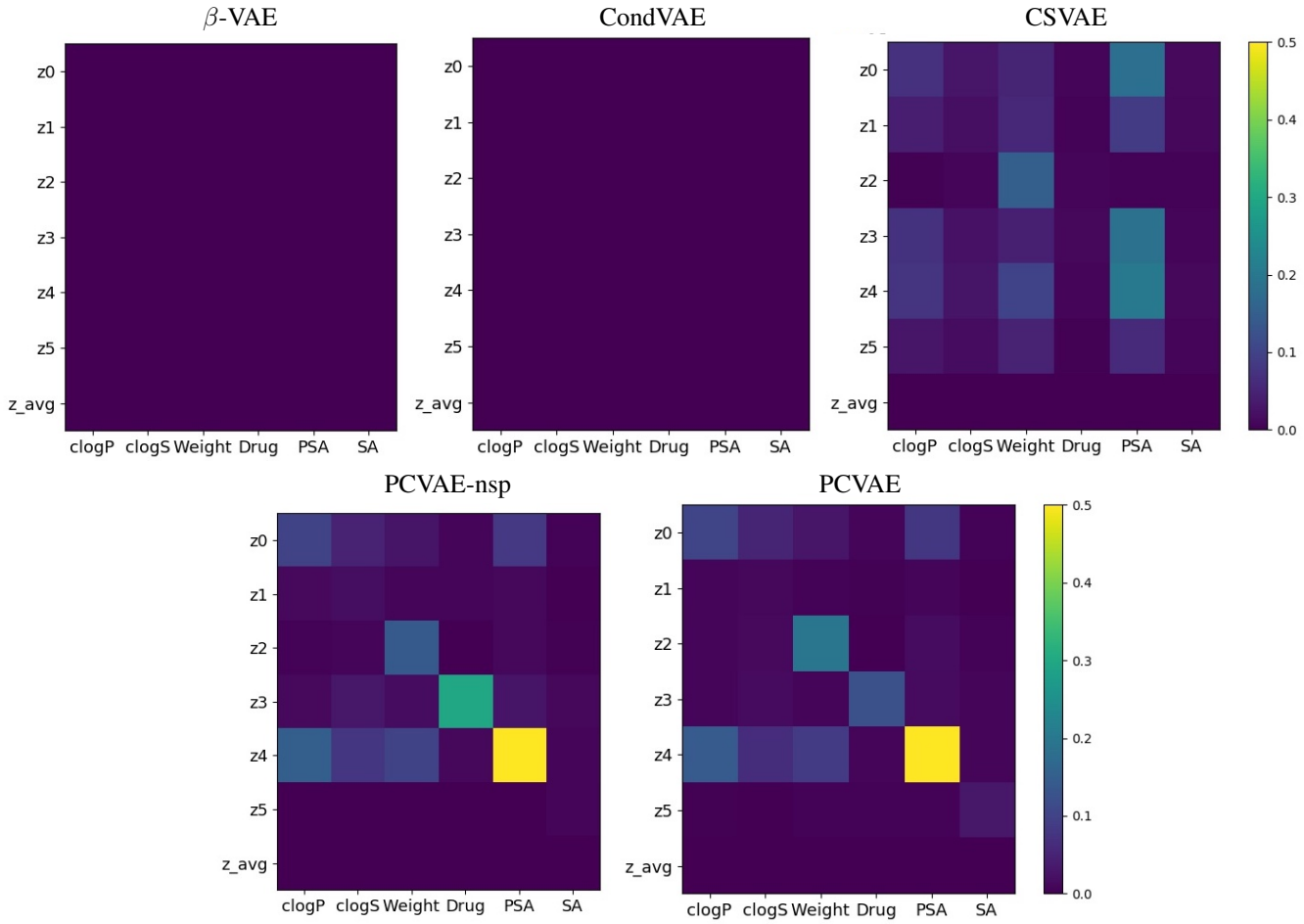

Fig. 3. MI is calculated between each of the disentangled factors learned by a (QM9-trained) model and the molecular properties computed on the molecules generated by the model.

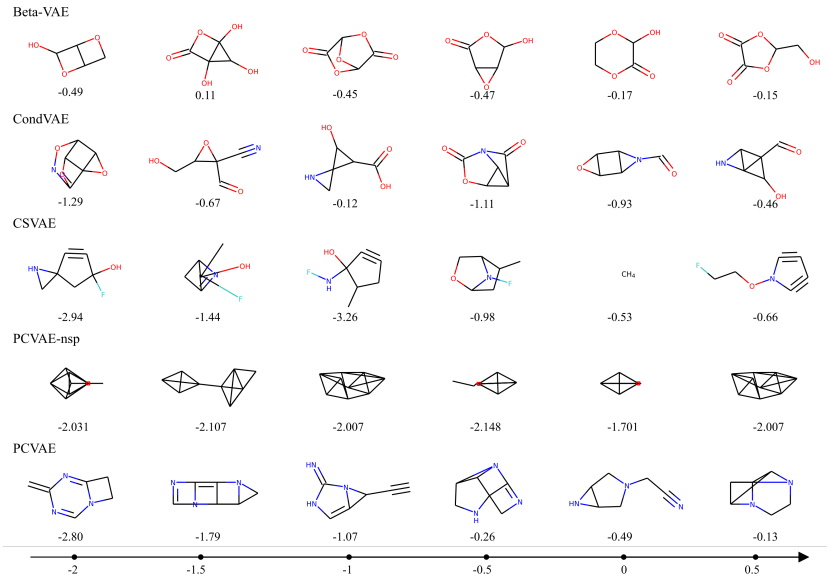

Fig. 4. The molecules in the following rows sampled at random over those generated by each model are shown, respectively, along with their cLogS values. The bottom axis indicates increasing the value of z from -2 to 0.5.

TABLE III

THE MSE BETWEEN THE EXPECTED VALUE OF EACH PROPERTY AND THE VALUE OBTAINED FROM MOLECULES GENERATED BY THE A MODEL AS DESCRIBED ABOVE. RESULTS ARE SHOWN FOR MODELS TRAINED OVER THE QM9 DATASET.THE BEST VALUE PER COLUMN IS HIGHLIGHTED IN BOLDFACE. THE SECOND ROW SHOWS THE RANGE OF HIGHEST DENSITY PER PROPERTY.

Method	cLogP	cLogS	Drug	Weight	PSA	SA
	$[-2, 2]$	$[-2, 2]$	$[-5, 5]$	$[120, 130]$	$[20, 60]$	$[2, 5]$
β-VAE	2.45	1.01	43.83	264.59	249.72	7.03
CondVAE	2.20	0.99	22.27	42.03	**183.43**	4.87
CSVAE	**0.67**	0.96	9.24	39.73	810.45	1.86
PCVAE-nsp	2.15	3.18	8.99	**38.45**	765.44	**1.84**
PCVAE	1.13	**0.62**	**5.41**	38.59	1554.00	1.87

IV. CONCLUSION

In this paper, we propose several deep latent-variable models to generate small molecules with desired molecular properties. The models operate under supervised, disentangled representation learning and leverage both graph representation learning to learn inherent constraints in the chemical space and inductive bias to connect chemical and biological space. The evaluations show that the models are a promising step in controllable molecule generation in support of cheminformatics, drug discovery, and other application settings. In practice, we observe that CSVAE is hard to train due to the adversarial scheme. The results also altogether point towards PCVAE-nsp and PCVAE as more effective models for property control. Much work remains. All current models for small molecule generation, including those proposed in this paper, are concerned with global properties. Preserving local properties may be additionally desirable, as it may provide chemical biologists with a better understanding of the contribution of local elements onto global properties, as well as guide them on how to further modify molecules in the wet laboratory.

ACKNOWLEDGMENT

Computations were run on ARGO, a research computing cluster provided by the Office of Research Computing at George Mason University, VA (URL: http://orc.gmu.edu). 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.