Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

MoVAE: A Variational AutoEncoder for Molecular Graph Generation

Zerun Lin*† Yuhan Zhang*‡ Lixin Duan‡ Le Ou-Yang†¶ Peilin Zhao§¶

Abstract

Molecule generation plays an important role in accelerating drug discovery. In recent years, many molecule generation methods have been proposed based on variational autoencoders (VAEs), due to its advantages in latent manifold representation learning and training stability. However, most of the existing VAE-based models require tedious graph matching operations during training, and tend to generate invalid molecules. To overcome these limitations, in this paper, we propose a novel molecular graph variational autoencoder (MoVAE). Firstly, to avoid complicated graph matching, the proposed MoVAE only encodes and decodes all the nodes and edges individually. Secondly, to improve the generation validity, it adversarially trains the model by treating the encoder and decoder as the discriminator and generator. In addition, to generate molecules with various target conditions, the MoVAE also introduces drug property constraints and valence histogram constraints. Experiment results on two real datasets show that our model outperforms almost all the state-of-the-art algorithms.

1 Introduction

Generation of new compounds with desirable properties is a challenging task in drug discovery [1]. In the past, molecule designs were mainly based on expert manual design and enumeration in traditional methods [2, 3], which not only rely on the domain knowledge and intuition of human experts, but are also time-consuming and labor-intensive. Recently, as machine learning-based methods can automatically generate desired molecules in a straightforward data-driven manner, which greatly reduce the time, money, and labor cost of developing new drugs, AI-based drug discovery has become increasingly popular [4, 5, 6]. Since the generation of high-quality molecules with desired properties is important for drug discovery, an increasing attention has been paid to the development of machine learning methods for generating "good" molecules [7].

Based on different molecular representation strategies, various computational approaches have been proposed for molecule generation [8, 9]. Simplified Molecular Input Line Entry System (SMILES) is a line notation that encodes molecular structures as sequences of symbols with a simple vocabulary and grammar rules via depth-first traversal. Recently, numerous SMILES-based molecule generation methods have been proposed [10, 11]. But molecule generation based on SMILES cannot ensure 100% chemical validity unless very complex constraints are added. Moreover, generating a valid SMILES string requires the model to learn molecular structure-irrelevant rules, such as the SMILES grammar, which adds extra burden on the model [12].

A molecule can be naturally modeled as an undirected molecular graph whose nodes and edges correspond to the atoms and chemical bonds of the molecule, respectively. Because the atoms and bonds that make up the nodes and edges are determined in advance to ensure that all outputs are valid molecules, graph-based molecule generation can be more efficient than generating syntactically valid SMILES strings [13]. Existing graph-based molecule generation methods can be roughly divided into sequential generation-based methods [14] and complete graph generation-based methods. For example, JT-VAE [15] uses the variational encoder to generate a molecular fragment tree, and then utilizes the composition of the fragments to synthesize the final molecular graph. CGVAE [16] selects the focused node from the generated initial atoms and finds out the nodes that may be connected to the focused node, then selects the type of bond and generates the final graph subsequently. However, sequential generation-based models may accumulate errors during the generation processes and ignore the overall correlation of molecular graph. Thus, complete graph generation-based methods have been developed for generating molecular graphs. For example, GraphVAE [17], a generative model based on VAE, and MolGAN [18], a reinforcement learning based model, are two representative methods. However, GraphVAE requires the addition of a graph matching module to solve the isomorphism problem, which is computationally expensive. Therefore, approximate graph matching, which approximates the distance between a graph and its reconstruction using the number of node

*These authors contributed equally. This work is done when Yuhan Zhang worked as an intern in Tencent AI Lab.

†Shenzhen University, Shenzhen, China. Email: 2060432004@email.szu.edu.cn, leouyang@szu.edu.cn

‡University of Electronic Science and Technology of China. Email: yuhanzhan9@gmail.com, lxduan@gmail.com

§Tencent AI Lab, China, Email: masonzhao@tencent.com

¶Corresponding Authors: Le Ou-Yang and Peilin Zhao

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

types, edge types, node-edge pair types, and node-edge-node pair types, was proposed [19]. But this approximate graph matching module does not fully guarantee isomorphism invariance.

To address the above problems, in this study, we propose a novel molecular graph variational autoencoder (MoVAE) for molecule generation, which combines the advantages of VAEs and generative adversarial networks (GANs) models. Our MoVAE model does not require an additional discriminator, as the inference model itself acts as a discriminator to distinguish generated samples from real samples, which overcomes the tendency of VAE based models to blur similar molecular graphs when generating molecular graphs [20]. At the same time, for the tedious problem of graph matching in molecule generation, we propose a framework that embeds nodes and edges respectively, and reconstructs nodes and edges in one-to-one correspondence. This generation method can guarantee the invariance of graph isomorphism. Finally, we also consider drug attribute constraints and valence histogram constraints, which can conditionally generate molecules that satisfy various target conditions. We evaluated the performance of our algorithm on two real datasets, namely ZINC and QM9. The experiment results showed that our algorithm could achieve the desired results on both datasets. Specifically, our contributions can be summarized as follows:

- To improve the effectiveness of generating molecules, in this algorithm, we adversarially train the model by treating the encoder and decoder as the discriminator and generator, respectively.

创新点

- The one-to-one correspondence generation method of node-to-node and edge-to-edge is adopted to ensure the isomorphic invariance of the generated molecular graphs and avoid complicated graph matching.
- Additional constraints on drug properties and valence histogram are added to generate drug molecular graphs with specific properties. Constraints on the valence histogram can guarantee the valence matching for atoms in the molecule and make the generated molecular graph more physically meaningful.

2 Related Work

2.1 VAE-Based Molecule Generation Variational autoencoders (VAEs) have been widely used for molecule generation. The VAE consists of two parts: a generative network (Decoder) $p_\theta(Y|Z)$, which samples the visible variable Y given a latent variable Z, and an approximate inference network (Encoder) $q_\phi(Z|Y)$, which maps the visible variable Y to the latent variable Z that approximates the prior $p(Z)$. The goal of the VAE is to maximize the evidence lower bound (ELBO) of $p_\theta(Y)$:

$$\begin{aligned} &\log p_\theta(Y) \\ &\geq E_{q_\phi(Z|Y)} \log p_\theta(Y|Z) - D_{KL}(q_\phi(Z|Y)|p(Z)) \\ &= ELBO(Y). \end{aligned} \tag{2.1}$$

The first term is the reconstruction loss, which minimizes the difference between the original graph and the graph generated using the embedding of the original graph. The second term utilizes the KL divergence to minimize the difference between the posterior probability distribution $q_\phi(Z|Y)$ and the prior distribution $p(Z)$.

2.2 Conditional Constrained Molecule Generation The ultimate goal of molecular generation is to directly generate molecules with desired properties, such as finding molecules with specific therapeutic effects in drug development, and finding suitable photosensitizer molecules in the field of organic materials development [21]. VAE is suitable for generating new data but does not have control over what is generated, which clearly does not satisfy our further requirements for molecular generation. Thus, a supervised version of VAE that can generate desirable data by specifying labels l, named CVAE [22], has been proposed. Similar to VAE, the loss function of CVAE is as follows:

$$\begin{aligned} &\log p_\theta(Y|l) \geq E_{q_\phi(Z|Y,l)} \log p_\theta(Y|Z,l) \\ &\quad - D_{KL}(q_\phi(Z|Y,l)|p(Z|Y)) = ELBO(Y|l), \end{aligned} \tag{2.2}$$

where, (Y, l) is the data pair in general supervised learning. For training a molecule generation model, we can add a label l to each molecule Y during the pre-processing of the dataset. After training, we can generate a molecule with desirable label l by feeding this label and a sample Z from prior $p(Z|Y)$ into the decoder.

3 Method

3.1 Problem Formulation A molecule is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes a set of n atoms (nodes) and $\mathcal{E}$ denotes a set of m chemical bonds (edges). It is worth noting that there is no directional distinction between edges. The node feature matrix X represents the attributes of each node, the adjacency matrix A represents the relationship between each node, l represents the drug property label of the molecule, and h represents the valence histogram corresponding to the molecule. Suppose that the number of node types is p and the number of edge types is q, then we can obtain $A \in \{0,1\}^{n\times n\times q}$ and $X \in \{0,1\}^{n\times p}$, where $A_{ijk} = 1$ means that there is an

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

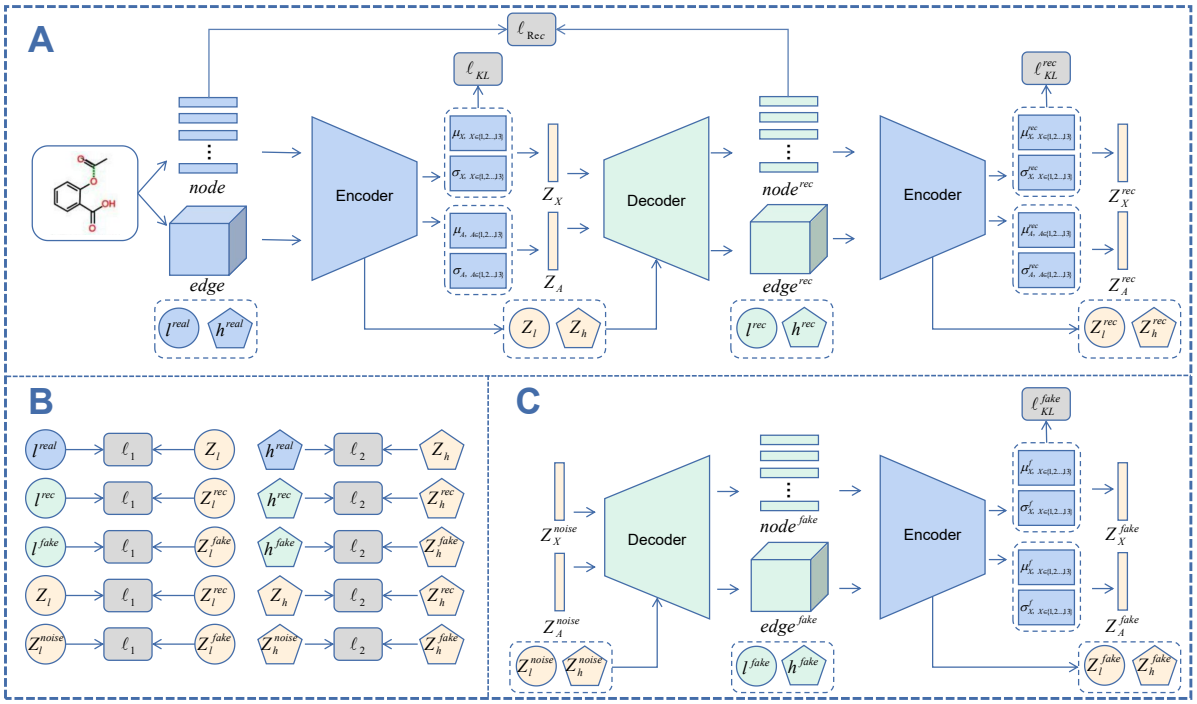

Figure 1: An overview of the MoVAE model. A: After decomposing the molecular graph into atoms and bonds, RDKit is used to match the corresponding drug property labels and valence histograms to the molecular graph. The encoder embeds atoms and bonds into continuous representations, respectively. Then, a sample is sampled from the distribution of nodes and edges, respectively, and sent to the decoder with drug attribute constraints and valence histogram constraints to generate the corresponding molecular graph. B: We use L1 loss and L2 loss to calculate the loss of drug property labels and valence histogram, respectively. To avoid complicated wiring, it is shown here as a module. C: Improve the effectiveness of molecule generation by using its own encoder and decoder as discriminator and generator.

edge of type k between the i^{th} and j^{th} nodes. Our goal is to learn an auto-generation framework based on Variational Auto-Encoders (VAEs) for data-driven molecular graph generation.

3.2 Overview Our proposed algorithm is termed as **Mo**lecular Graph **V**ariational **A**uto-**E**ncoders (**MoVAE**). The flow of our algorithm is shown in Fig. 1. Firstly, a chemical molecule is represented as a graph $\mathcal{G}$, and the node feature matrix X, adjacency matrix A, drug property label l and molecular valence histogram h of the molecular graph are generated accordingly. Different from traditional VAEs for graphs, our MoVAE only encodes and decodes all nodes and edges separately, which successfully avoids challenging graph matching between input and generated graphs. Then, based on the idea of introspective training (a special case of generative adversarial training), the encoder and decoder are used as discriminators and generators to adversarially train the model to improve the effectiveness of model generation. Meanwhile, during encoding and decoding, the molecular graph drug property labels and valence histogram constraints are input to generate molecules with various target conditions.

3.3 Graph Embedding based on Introspective Training It is well known that one of the limitations of VAEs is the low quality of generated samples, such as blurred images for image generation, which is attributed to the limited representation power of the model, injected noise, and imperfect generation criteria such as squared error [23, 24]. To improve the quality of generated molecular graphs, our MoVAE adopts the idea of introspective training, inspired by the Soft Introspective Variational Autoencoder (Soft-Intro VAE) algorithm [25].

Specifically, our MoVAE model is adversarially trained with the encoder and decoder as discriminator and generator, respectively. The specific implementation is to train the encoder such that the latent embedding distribution of the real molecules is close to the prior distribution, and the latent embedding distribution of the generated molecules deviates from the prior distribution. While the decoder is trained to make the latent embedding distribution of the generated molecules close to the prior distribution. Thus, by

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

combining with Equation 2.1, the objective functions of our MoVAE for the Encoder E_ϕ, and the Decoder D_θ are defined as:

(3.3)
$$\mathscr{L}_{E_\phi}(Y,Z) = ELBO(Y) - \frac{1}{\alpha}\exp(\alpha ELBO(D_\theta(Z))),$$
$$\mathscr{L}_{D_\theta}(Y,Z) = ELBO(Y) + \gamma ELBO(D_\theta(Z)),$$

where $ELBO(Y)$ is defined as in Equation 2.1, $ELBO(D_\theta(Z))$ is also defined as in Equation 2.1 with $D(Z) \backsim p_\theta(Y|Z)$ instead of sample Y, and α and γ are hyper-parameters. Equation 3.3 depicts the relationship between encoder and decoder: the encoder is optimized to distinguish between real samples (high ELBO) and generated samples (low ELBO) by the ELBO value, while the decoder is optimized to generate samples that can "fool" the encoder [25]. At the same time, a soft function is used for ELBO to solve the problem of unstable training due to the hard threshold function. It enables the model to self-estimate the difference between the generated samples and the training data, and then self-update the algorithm to produce more realistic samples.

In this way, our MoVAE preserves the advantages of VAEs, such as stable training and good latent manifold. Unlike other hybrid models of VAEs and GANs, MoVAE does not require an additional discriminator, as the inference model itself acts as a discriminator to distinguish generated samples from real ones. Such a framework narrows the gap between VAE and GAN in terms of sampling quality, while still retaining the advantages of the variational inference models.

3.4 Embedding Nodes and Edges Individually In VAE generation, the reconstruction loss $\mathbb{E}_{z\backsim q_\phi(z|\mathcal{G},l,h)}[-\log p_\theta(\mathcal{G}|z,l,h)]$ involves the original input graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ and its VAE reconstruction $\hat{\mathcal{G}} = (\hat{\mathcal{V}},\hat{\mathcal{E}})$. In traditional VAE algorithms for graphs, it is often necessary to introduce a graph matching procedure to seek the best possible match between two graphs to ensure isomorphic invariance of the generated molecular graph. However, solving the graph matching problem is non-trivial and inefficient. To avoid this, in this algorithm we try to reconstruct the types of nodes and edges with a one-to-one correspondence between them. Specifically, as shown in Fig 1.A, we encode nodes and edges individually and learn the distribution of nodes and edges in the latent space, respectively.

For the encoder and decoder, we adopt a message passing neural network (MPNN) [26]. We take X and A as the input to the encoder and learn the node embedding distribution, edge embedding distribution, drug properties and valences histograms in a low-dimensional space. Then, we try to reconstruct the node feature matrix and adjacency matrix with a decoder of similar structure:

(3.4)
$$E_\phi(X,A) = q_\phi(Z_X,Z_A,Z_l,Z_h|X,A),$$
$$D_\theta(Z_X,Z_A,Z_l,Z_h) = p_\theta(X,A|Z_X,Z_A,Z_l,Z_h).$$

The distributions $q_\phi(Z_X,Z_A,Z_l,Z_h|X,A)$ and $p_\theta(X,A|Z_X,Z_A,Z_l,Z_h)$ parameterized by ϕ and θ denote the encoder and decoder of the VAE, respectively. To train the encoder and decoder, our MoVAE also constructs molecules from noise embedding sampled from the prior distribution. Thus, for MoVAE, we have the following state variables:

(3.5)
$$Z_X, Z_A, Z_l, Z_h = E_\phi(X,A),$$
$$X^{rec}, A^{rec} = D_\theta(Z_X,Z_A,Z_l,Z_h),$$
$$Z_X^{rec}, Z_A^{rec}, Z_l^{rec}, Z_h^{rec} = E_\phi(X^{rec},A^{rec}),$$
$$X^{fake}, A^{fake} = D_\theta(Z_X^{noise},Z_A^{noise},Z_l^{noise},Z_h^{noise}).$$

In the VAE model, the output of the encoder includes two separate variables, i.e., the mean μ and the variance σ in the low-dimensional distribution. The input Z of the decoder is sampled from $\mathcal{N}(Z;\mu,\sigma^2)$ using a reparameterization trick: $Z = \mu + \sigma \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0,I)$. In this setting, the KL objective of our MoVAE, is defined as follows:

(3.6)
$$\mathscr{L}_{KL}^E = -\log(NKL^{real}) - \log(1-NKL^{rec}) - \log(1-NKL^{fake}),$$

(3.7)
$$\mathscr{L}_{KL}^D = -\log(NKL^{rec}) - \log(NKL^{fake}),$$

where

(3.8)
$$NKL^{real} = e^{-KL(concate(Z_X,Z_A)|\mathcal{N}(0,I))},$$
$$NKL^{rec} = e^{-KL(concate(Z_X^{rec},Z_A^{rec})|\mathcal{N}(0,I))},$$
$$NKL^{fake} = e^{-KL(concate(Z_X^{fake},Z_A^{fake})|\mathcal{N}(0,I))}.$$

Since we learn node and edge embeddings separately, our reconstruction loss function also consists of node and edge parts. By combining with the above KL objective function, we can simplify Equation 3.3 into the following form:

(3.9)
$$\mathscr{L}^E = \mathscr{L}_r(X) + \mathscr{L}_r(A) + \lambda_{KL}\mathscr{L}_{KL}^E,$$
$$\mathscr{L}^D = \mathscr{L}_r(X) + \mathscr{L}_r(A) + \beta_{KL}\mathscr{L}_{KL}^D,$$

where $\mathscr{L}_r(X) = E_{q_\phi(Z|X)}\log p_\theta(X|Z)$ represents the reconstruction loss error of the node, and $\mathscr{L}_r(A)$ represents the reconstruction loss error of the edge.

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

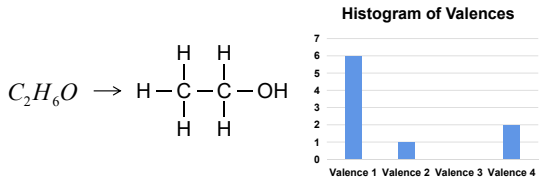

Figure 2: An example of a molecular valence histogram is shown here for an ethanol molecule, where Carbon has valence 4, Hydrogen has valence 1, and Oxygen has valence 2.

3.5 Auxiliary Drug Property Constraints To further improve the quality of generated samples, our MoVAE adds drug property constraints to the molecular generation model [27, 28]. Specifically, a drug property $l^{real}(l^{rec},\ l^{fake})$ can be computed by RDKit [29] for each molecule, including quantitative estimate of drug-likeness (QED), or Synthetic Accessibility score (SA score). The addition of conditional attribute constraints to MoVAE is beneficial in producing molecules that are physically similar to a given dataset under specific conditions [21].

To this end, when generating the latent space distribution Z_X and Z_A of nodes and edges, MoVAE takes the node feature matrix X and the adjacency matrix A as the input of the encoder, and learn the latent vector Z_l of the label vector l^{real} in the latent space. As shown in Fig 1.B, this introduces a property constraint loss for the encoder:

$$\mathscr{L}^E_{Property} = \|l^{real} - Z_l\|_1 + \|l^{rec} - Z_l^{rec}\|_1 + \|l^{fake} - Z_l^{fake}\|_1. \tag{3.10}$$

Then, MoVAE not only takes Z_X and Z_A as the input of the decoder, but also the corresponding properties Z_l. In this way, MoVAE can generate new molecules that are physically similar to the training set by sampling from a conditional generative distribution for the decoder. This can be formulated as the following loss for the decoder:

$$\mathscr{L}^D_{Property} = \|Z_l - Z_l^{rec}\|_1 + \|Z_l^{noise} - Z_l^{fake}\|_1. \tag{3.11}$$

3.6 Molecular Valence Histogram Constraints Most existing node-to-node methods assume that the atoms in the molecule are independent of each other, and sample from the latent space independently for each atom [30, 31]. However, they clearly do not conform to physical laws, since atoms in the same molecule must depend on each other. Thus, our MoVAE adds an valence histogram constraint on the sampled atoms: given a molecule with m atoms, the maximum atom valence $v \in \mathbb{N}$ [32]. Take Fig 2 for example, the valence histogram h is generated considering the valences of all m atoms in the molecule, where $h[i]$ represents the number of atoms with valence i in the molecule. The valence histogram constraint takes into account the valence matching of atoms, which can further improve the generation validity.

Specifically, MoVAE treats the valence histogram as a prior knowledge, i.e., a label of the molecule. Firstly, the valence histogram is generated for each molecule as a prior knowledge. Then, MoVAE encodes atoms and edges to generate a latent representation of the valence histogram, and decodes the latent variable to restore the histogram. Accordingly, similar to the constraints of drug property, this results in two loss functions for the encoder and decoder, respectively:

$$\mathscr{L}^E_{hist} = \|h^{real} - Z_h\|_2 + \|h^{rec} - Z_h^{rec}\|_2 + \|h^{fake} - Z_h^{fake}\|_2, \tag{3.12}$$

$$\mathscr{L}^D_{hist} = \|Z_h - Z_h^{rec}\|_2 + \|Z_h^{noise} - Z_h^{fake}\|_2. \tag{3.13}$$

After training, we can sample from $\mathcal{N}(0, I)$, or can sample an existing histogram from the training set for specified generation. This is different from existing methods, such as CCGVAE [32], which uses the valence histogram to guide atom-by-atom selection.

3.7 Final Objective Function The final objective function of MoVAE can be divided into the following two parts: encoder objective function $\mathscr{L}^E$ and decoder objective function $\mathscr{L}^D$:

$$\mathscr{L}^E(\phi) = \mathscr{L}^E + \lambda_1 \mathscr{L}^E_{Property} + \lambda_2 \mathscr{L}^E_{hist}, \tag{3.14}$$

损失函数

$$\mathscr{L}^D(\theta) = \mathscr{L}^D + \beta_1 \mathscr{L}^D_{Property} + \beta_2 \mathscr{L}^D_{hist}, \tag{3.15}$$

where λ_1, λ_2, β_1 and β_2 are hyper-parameters that control the balance between different objective functions.

Given the empirical data distribution $p_{data}(\mathcal{G}, l, h)$, we train the MoVAE model on the entire dataset to optimize $\mathscr{L}^E(\phi)$ and $\mathscr{L}^D(\theta)$ simultaneously. During each iteration, a batch of Y is sampled from the data distribution. Then, the encoder's parameter ϕ, and the decoder's parameter θ are updated by gradient descent using the loss function $\mathscr{L}^E(\phi)$, and the loss function $\mathscr{L}^D(\theta)$, respectively. Finally, the learning procedure is summarized in Algorithm 1.

Once the model is trained, we can use the decoder $p_\theta(\mathcal{G}|Z, l, h)$ to generate molecular graphs. For unconditional generation, l is sampled from the prior distribution $p(l)$. To conditionally generate molecular graph, l can be sampled from a conditional distribution.

4 Evaluation

In this section, we will compare our algorithm with several state-of-the-art baselines on real-world datasets. We will first introduce the data sets and evaluation metrics, and then present the experiment results.

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Algorithm 1 Learning Procedure

Input: Empirical data distribution $p_{data}(\mathcal{G}, l, h)$, batch size T

Parameter: Optional list of parameters

Output: Learned parameters ϕ, θ

procedure TRAIN
  Initialize ϕ, θ
  repeat
    Sample $Y = \{(\mathcal{G}_T, l_T, h_T)\}_{t=1}^{T}$ from p_{data}
    Update ϕ by gradient descent of $\mathscr{L}^E(\phi)$
    Update θ by gradient descent of $\mathscr{L}^D(\theta)$
  until termination condition

数据集

4.1 Datasets and Metric We considered two commonly used molecular datasets:

- QM9 has about 134,000 organic molecules. Each molecule consists of up to 9 atoms of types $\{C, N, O, F, None\}$, and 4 different bond types $\{single, double, triple, none\}$ [33], [34].
- ZINC has 250,000 drug-like molecules. Each molecule has a maximum of 38 atoms of types $\{C, N, O, F, P, S, Cl, Br, I, None\}$, also 4 different bond types $\{single, double, triple, none\}$ [6].

For each molecule, RDKit package can generate drug property labels, including quantitative estimate of drug-likeness (QED), the molecular weight (MolWt) and Wildman-Crippen partition coefficient (LogP). In the following experiments, QED is chosen as the label for the property constraints of MoVAE, while the other labels are left as future work.

The following commonly used metrics were chosen to evaluate the performance of various algorithms:

- **Validity**: Given a set of generated molecules, the percentage of molecules that are valid.
- **Uniqueness**: The ability of the model to generate different molecules. Specifically, the percentage of unique molecules in the resulting pool.
- **Novelty**: The ratio of generated molecules that do not belong to the training dataset and satisfy the chemical validity.
- **G-mean**: The geometric mean of three indicators: Validity, Uniqueness, Novelty, for overall comparison between models.
- **Diversity**: Measure the richness of the generated molecular graph.
- **Synthetic Accessibility Score (SAS)**: The Synthetic Accessibility Score, which represents how easy (0) or difficult (1) it is to synthesize a molecule.
- **Quantitative Estimation Drug-likeness (QED)**: The drug-like properties of the resulting molecule, indicating the likelihood of the molecule being a candidate drug.

评价指标

4.2 Experiment Results In this subsection, we will introduce the experimental settings, and show the empirical results.

4.2.1 Experimental Settings Two types of algorithms are adopted as baselines. One is SMILES sequence-based models, including character VAE [10], Grammar VAE [35], Syntax Directed VAE [36]. The other is graph-based models, including Graph VAE [37], Regularized GCAE [37], Junction Tree VAE [15], MolGAN [18], CGVAE [16] and CCGVAE [32]. The empirical results of these models are reported in [38].

Since the number of atoms varies in different molecular graphs, the largest atom number of different molecules will be used as the criterion. Other molecules with fewer atoms will be filled with "0", so that the number of atoms is the same for different molecules.

For MoVAE, the structure of the encoder is MPNN. The dimension of the embeddings of nodes and edges are set to 1. Then a MPNN layer with a Tanh activation function is used to generate the mean and variance of nodes and edges in low-dimensional distributions, respectively. In addition, two MPNN layers with Tanh activation functions are used to generate low-dimensional representations of drug property labels and valence histograms, respectively. The decoder also uses MPNN to reconstruct the molecular graph from the sampled embeddings. The hyper-parameters β_1, β_2, λ_1, λ_2 are all set to 1 in the experiments. For every dataset, the MoVAE model is trained for 80 epoches using the AdamW optimizer. The learning rate of both encoder and decoder is set to $1e-4$. The batch size is set to 128.

We also perform ablation studies on our model to investigate the effectiveness of drug properties constraints and valence histogram constraints. We evaluated two ablation models, denoted as A1 and A2, respectively. For model $A1$, λ_2 and β_2 are set to 0, so only the drug attribute constraints are added. For model $A2$, λ_1 and β_1 are set to 0, so only valence histogram constraints are added.

4.2.2 Generating molecules Table 1 shows the experimental results of different molecule generation models. The direction of each arrow indicates the measurement direction of the indicator. The ↑ indicates that the higher the value, the better the performance, while the ↓ indicates that the lower the value, the better the performance.

Among them, G-mean is the geometric mean of the three indicators of validity, uniqueness and novelty, which represents the comprehensive evaluation of the three indicators. Bold numbers indicate the best per-

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Table 1: Experiments results on QM9 and ZINC datasets. Here, ↑ indicates that the metric should be maximized, while ↓ indicates that the metric should be minimized. Bold indicates the best value for that metric.

Dataset	Baseline	Method	↑Validity	↑Unique	↑Nov	↑G-mean	↑Div	↓SAS	↑QED
QM9	SMILES-based	Character VAE	0.064	0.923	0.994	0.389	0.980	0.435	0.303
		Grammar VAE	0.045	0.832	0.942	0.328	**0.989**	0.338	0.319
		Syntax Directed VAE	0.150	**1.000**	**1.000**	0.531	0.977	0.229	0.350
	graph-based	Regularized GVAE	0.877	0.831	0.413	0.670	0.630	0.289	**0.488**
		GraphVAE	0.891	0.857	0.428	0.670	0.669	0.321	0.483
		JT-VAE	0.999	0.903	0.911	0.937	0.576	0.189	0.462
		MolGAN	0.767	0.200	0.562	0.442	0.611	0.311	0.484
		CGVAE	**1.000**	0.989	0.928	0.971	0.791	**0.138**	0.468
		CCGVAE	**1.000**	0.932	0.885	0.938	0.792	0.171	0.466
		A1	0.954	0.955	**1.000**	0.969	0.922	0.489	0.464
		A2	0.977	0.943	**1.000**	0.973	0.932	0.551	0.446
		MoVAE	0.969	0.976	**1.000**	**0.982**	0.938	0.521	0.453
ZINC	SMILES-based	Character VAE	0.009	0.914	**1.000**	0.203	0.982	0.311	0.387
		Grammar VAE	0.051	0.946	**1.000**	0.364	0.992	0.268	0.254
		Syntax Directed VAE	0.190	**1.000**	**1.000**	0.575	0.936	0.147	0.395
	graph-based	Regularized GVAE	0.865	0.903	**1.000**	0.921	0.978	0.446	0.344
		GraphVAE	0.140	0.314	**1.000**	0.353	0.715	0.281	0.459
		JT-VAE	0.996	0.998	0.999	0.998	0.330	0.447	**0.751**
		CGVAE	**1.000**	0.999	**1.000**	0.999	0.660	0.163	0.651
		CCGVAE	**1.000**	0.928	**1.000**	0.975	0.800	0.200	0.524
		A1	**1.000**	0.953	**1.000**	0.984	0.983	0.106	0.432
		A2	**1.000**	0.933	**1.000**	0.977	0.963	0.143	0.354
		MoVAE	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.032**	0.349

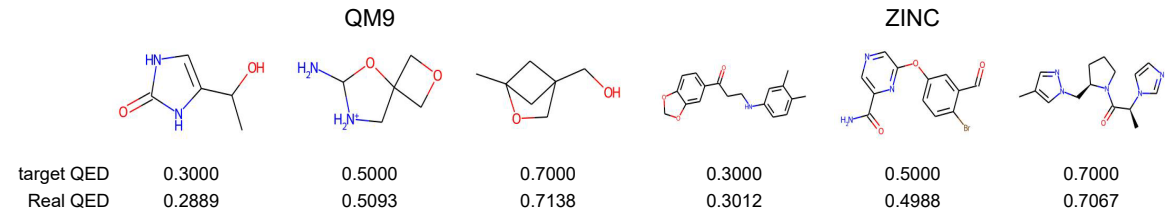

Figure 3: Specific QED attribute values guide the generation of molecules.

forming algorithms. From Table 1, we can observe that the G-mean of our model is the best on both datasets, compared with sequence based and graph based models. This indicates our MoVAE achieves the best trade-off on all the tree metrics.

Among all the algorithms, the sequence based models, such as Syntax Directed VAE, generally enjoys high uniqueness and novelty. However, their validities are too low, so that they are impractical for real applications. The reason for their low validity is that they only learn on SMILES sequence which can not incorporate the information of the topological structure of each molecule. This indicates the importance to study graph based molecule generation models.

Now, lets compare MoVAE only with graph based models. Firtly, our MoVAE reaches the highest validity, uniqueness and novelty for the ZINC dataset, and also performs very well on the QM9 dataset. This indicates the effectiveness of our VAE model. Secondly, MoVAE achieves the best diversity on both datasets, which indicates our model is more possible to generate new molecules. Thirdly, MoVAE obtains the best SAS on ZINC, common SAS on QM9, and fine QED on both datasets. By the way, JT-VAE achieves much higher QED on the ZINC dataset, since it employs substructures to generate the molecules. However, its diversity is very low.

Compared with the ablation model, the uniqueness of the A1 model is higher than that of the A2 model, which indicates that the drug property constraint contributes to the generation of diverse molecules. However, the validity of the A2 model is higher than that of the A1 model, indicating that the valence histogram constraint contributes to the generation of valid molecules. Overall, the performance of our MoVAE on both datasets is optimal, thanks to the one-to-one cor-

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

respondence between atoms and bonds.

4.2.3 Directed molecule generation In our model, drug property constraints are added so that our algorithm can be used to guide the generation of specific molecules. We label each molecule with drug properties during training (only QED values were used in the experiments), and retrain the model. This allows us to generate a particular molecule with a particular QED value by setting a fixed QED value when generating the molecule. Fig 3 shows the performance of the algorithm in generating specific molecules, including the expected QED values and the QED values calculated by RDKit for the generated molecules. From the results, it can be seen that the proposed model successfully generates molecular graphs with properties close to the target value when a target conditional QED is set. This again demonstrates the effectiveness of our MoVAE model.

5 Conclusion

In this study, we propose a VAE-based chemical molecule generation algorithm, named MoVAE, which combines the advantages of the VAE model and the GAN model. MoVAE does not require an additional discriminator because the inference model itself acts as a discriminator to distinguish generated samples from real samples, which overcomes the tendency of VAE models to blur similar molecular graphs when generating molecular graphs. To deal with the tedious problem of graph matching in molecule generation, we propose a framework that embeds nodes and edges separately and reconstructs nodes and edges in one-to-one correspondence, which guarantees the invariance of graph isomorphism. Moreover, to generate molecules with various target conditions, we also incorporate drug property constraints and valence histogram constraints into our model. Experiment results on two real datasets verify the effectiveness of our model in molecule generation.

References

[1] Swen Hoelder, Paul A Clarke, and Paul Workman. Discovery of small molecule cancer drugs: successes, challenges and opportunities. *Molecular Oncology*, 6(2):155–176, 2012.

[2] Ivan Huc and Jean-Marie Lehn. Virtual combinatorial libraries: dynamic generation of molecular and supramolecular diversity by self-assembly. *Proceedings of the National Academy of Sciences*, 94(6):2106–2110, 1997.

[3] Radislav Potyrailo, Krishna Rajan, Klaus Stoewe, Ichiro Takeuchi, Bret Chisholm, and Hubert Lam. Combinatorial and high-throughput screening of materials libraries: review of state of the art. *ACS Combinatorial Science*, 13(6):579–633, 2011.

[4] José Jiménez-Luna, Francesca Grisoni, and Gisbert Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.

[5] Hyunho Kim, Eunyoung Kim, Ingoo Lee, Bongsung Bae, Minsu Park, and Hojung Nam. Artificial intelligence in drug discovery: a comprehensive review of data-driven and machine learning approaches. *Biotechnology and Bioprocess Engineering*, 25(6):895–930, 2020.

[6] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015.

[7] Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of Computer-aided Molecular Design*, 27(8):675–679, 2013.

[8] Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, 4(4):828–849, 2019.

[9] Faezeh Faez, Yassaman Ommi, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Deep graph generators: A survey. *IEEE Access*, 9:106675–106702, 2021.

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

[11] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020.

[12] Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of Cheminformatics*, 10(1):1–24, 2018.

[13] Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, 2021.

[14] Daniel D Johnson. Learning graphical state transitions. *International Conference on Learning Representations (ICLR)*, pages 1–19, 2017.

[15] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited

Downloaded 12/03/24 to 57.181.173.163 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.

[16] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in Neural Information Processing Systems*, 31, 2018.

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

[18] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[19] Youngchun Kwon, Jiho Yoo, Youn-Suk Choi, Won-Joon Son, Dongseon Lee, and Seokho Kang. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics*, 11(1):1–10, 2019.

[20] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. *Advances in Neural Information Processing Systems*, 31, 2018.

[21] Myeonghun Lee and Kyoungmin Min. Mgcvae: Multi-objective inverse design via molecular graph conditional variational autoencoder. *Journal of Chemical Information and Modeling*, 2022.

[22] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28, 2015.

[23] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566. PMLR, 2016.

[24] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[25] Tal Daniel and Aviv Tamar. Soft-introvae: Analyzing and improving the introspective variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4391–4400, 2021.

[26] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[27] Seokho Kang and Kyunghyun Cho. Conditional molecular design with deep generative models. *Journal of Chemical Information and Modeling*, 59(1):43–52, 2018.

[28] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pages 2642–2651. PMLR, 2017.

[29] Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 2013.

[30] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

[31] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in Neural Information Processing Systems*, 31, 2018.

[32] Davide Rigoni, Nicolò Navarin, and Alessandro Sperduti. Conditional constrained graph variational autoencoders for molecule design. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 729–736. IEEE, 2020.

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

[34] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):1–7, 2014.

[35] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

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

[37] Tengfei Ma, Jie Chen, and Cao Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. *Advances in Neural Information Processing Systems*, 31, 2018.

[38] Davide Rigoni, Nicolò Navarin, and Alessandro Sperduti. A systematic assessment of deep learning models for molecule generation. *arXiv preprint arXiv:2008.09168*, 2020.

Copyright © 2023 by SIAM
Unauthorized reproduction of this article is prohibited