Molecule Generation for Drug Design: a Graph Learning Perspective

Nianzu Yang[1#] , Huaijin Wu[1#] , Junchi Yan[1*] , Xiaoyong Pan[2] , Ye Yuan[2] , Le Song[3]

[1]Department of CSE, and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University
[2]Department of Automation, Shanghai Jiao Tong University
[3]BioMap and MBZUAI
{yangnianzu,whj1201,yanjunchi,2008xypan,yuanye_auto}@sjtu.edu.cn le.song@mbzuai.ac.ae

Abstract

Machine learning has revolutionized many fields, and graph learning is recently receiving increasing attention. From the application perspective, one of the emerging and attractive areas is aiding the design and discovery of molecules, especially in drug industry. In this survey, we provide an overview of the state-of-the-art molecule (and mostly for *de novo* drug) design and discovery aiding methods whose methodology involves (deep) graph learning. Specifically, we propose to categorize these methods into three groups: i) all at once, ii) fragment-based and iii) node-by-node. We further present some representative public datasets and summarize commonly utilized evaluation metrics for generation and optimization, respectively. Finally, we discuss challenges and directions for future research, from the drug design perspective.

1 Introduction

In recent years, artificial intelligence based drug discovery has been more and more conspicuous since it greatly reduces time, money and labor costs for developing novel drugs [Jiménez-Luna *et al.*, 2020; Zhu, 2020; Kim *et al.*, 2020]. Among the processes of drug development, generating chemical molecules with good quality and optimizing chemical molecules for desired properties are of particular importance. So the challenge lies in how to apply machine learning methods to generate "good" molecules with or without additional constraints. Different approaches and models have been designed till now, including numerous generative model [Jin *et al.*, 2018; Shi *et al.*, 2020; Jin *et al.*, 2020a], reinforcement learning (RL)-based models [You *et al.*, 2018; Khemchandani *et al.*, 2020; Jin *et al.*, 2020b; Yang *et al.*, 2021a; Chen *et al.*, 2020], sampling-based models [Xie *et al.*, 2021; Fu *et al.*, 2021; Seff *et al.*, 2019] and evolutionary methods [Brown *et al.*, 2004; Jensen, 2019; Nigam *et al.*, 2020]. To characterize molecules, several types of molecular representations range from simple sequences of molecule entities to manually predefined molecular features [Redkar *et al.*, 2020] have been widely mentioned, while string-based and graph-based representations are two main methods used in recent years. Since the graph-based representation can catch the inherent structure of molecules, this paper focuses on graph-based methods.

There also exist surveys on molecule generation and optimization. [Guo and Zhao, 2020; Faez *et al.*, 2021] both provide a comprehensive overview of the literature in the field of deep generative models for graph generation. But they do not focus on molecules only. Apart from molecules, they also present deep generative models designed for other domains. As for [Elton *et al.*, 2019; Alshehri *et al.*, 2020], they both put emphasis on the four architectures often utilized as backbone model for molecule design methods, namely, recursive neural networks, autoencoders, generative adversarial networks, and reinforcement learning. It is also worth mentioning that [Elton *et al.*, 2019; Alshehri *et al.*, 2020; Guo and Zhao, 2020; Faez *et al.*, 2021] all discuss molecule design methods based on different molecular representations, including Simplified Molecular Input Line Entry System (SMILES) [Anderson *et al.*, 1987], 2D representation referring to connectivity graph and 3D representation that contains coordinates of the atoms within a molecule. There have been a rich body of literature on SMILES-based generative models [Gómez-Bombarelli *et al.*, 2018; Kang and Cho, 2018; Putin *et al.*, 2018; Grisoni *et al.*, 2020; Griffiths and Hernández-Lobato, 2020]. But the number of works on 3D-generative methods for molecule is quite small in contrast. Existing representative 3D methods include G-SchNet [Gebauer *et al.*, 2019], E-NF [Garcia Satorras *et al.*, 2021], GEN3D [Roney *et al.*, 2022] and G-SphereNet [Luo and Ji, 2022].

Context-sensitive SMILES-based molecule generation approaches cannot ensure 100% chemical validity [Elton *et al.*, 2019] unless complicated constraints are added. However, 2D-based molecule generation approaches are easily able to ensure that the generated molecules are 100% chemically valid. Different from existing surveys mentioned above, our survey puts emphasis on the molecule generation's application for drug design and conducts a comprehensive overview of the state-of-the-art drug design methods based on 2D representation, i.e. only from a graph learning perspective.

In this survey, we provide a thorough review of different state-of-the-art graph-based molecule generation and optimization methods proposed in recent years and classify them

[*] The first two authors contribute equally to this paper. Correspondence author is Junchi Yan.

into three categories in terms of their generation strategy, namely all-at-once, fragments-based and node-by-node. Representative public datasets are also discussed along with the common evaluation metrics. Furthermore, we give a thorough analysis of existing challenges and suggest three potential directions for future research.

2 Preliminaries and Problem Formulation

In this section, we first introduce graph-based molecule representation. Then, we formally formulate the problem of molecule generation and optimization task, respectively.

2.1 Graph-based Molecule Representation

Graph-based molecule representation uses molecule graph $G = (V, E)$ to model molecule, where V is the graph's node set mapping to atoms constituting a molecule and E is denotes the graph's edge set mapping to chemical bonds, with $|V| = n$ and $|E| = m$. In molecule graph, nodes are sometimes representing atomic types from the periodic table, or representing certain kinds of molecule fragments. The node feature matrix $\mathbf{X}$ characterizes the property of each node while the adjacency matrix $\mathbf{A}$ characterizes the relationships between each node. Let the number of edge types be b and the number of node types be c, then we have $\mathbf{A} \in \{0, 1\}^{n \times n \times b}$ and $\mathbf{X} \in \{0, 1\}^{n \times c}$, where $\mathbf{A}_{ijk} = 1$ when there exists an edge with type k between the i^{th} and j^{th} nodes, otherwise 0.

2.2 Problem Formulation

Molecule generation and optimization are closely related, but not all existing works on molecule generation involve optimization due to it is a further problem. Existing works involving optimization should propose a novel generative model for molecule generation first and then consider optimizing generated molecules for desired properties.

Molecule Generation

Generation tasks aim to generate novel samples from a similar distribution as the training data [Faez *et al.*, 2021]. A molecule generation method intends to generate novel, diverse molecules which follow the unknown data distribution $p(G)$ provided by a set of graphs D_G. A machine learning method towards this problem usually proposes a model to learn form large scales of data which either obtains an implicit strategy or estimates the $p(G)$ directly and then samples from the distribution to generate new molecules.

Molecule Optimization

In the scenario of molecule optimization, we are usually given a score function f measuring several needed property and a set of seed molecules with high scores. The goal of molecule optimization is to discover molecules with a high property score. Specifically, the problem can be formulated as given a molecule space $\mathcal{G}$ and a set of seed molecules $\mathcal{G}_0 \subset \mathcal{G}$, we aim to learn a molecule generative model $p(g)$ such that the expected score of generated molecules is maximized, i.e.

$$\max_{p(\cdot)} \mathbb{E}_{g \sim p(\cdot)}[f(g)] = \int_{g \in \mathcal{G}} p(g) f(g) \mathrm{d}g. \tag{1}$$

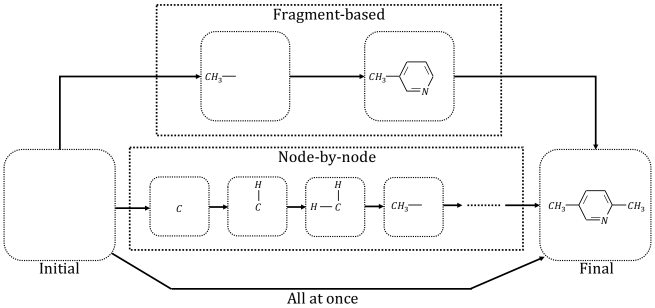

Figure 1: Three typical molecule generation strategies.

Notably, since molecule optimization problem is a kind of molecule generation problem in essence, by adding some constraints to the latter, we usually expect the distribution of generated molecules to be novel and diverse as well.

3 Generation and Optimization Strategies

For classifying existing methods of *de novo* molecule generation or molecule optimization, in this paper we propose to base on their generation or optimization granularity level as shown in Fig. 1: i) using an all-at-one generation or optimization strategy; ii) adopting rational substructures as editing blocks; or iii) building graphs in a node-by-node setting. We describe their main features as summarized in Table 1.

3.1 Generation Strategy I: All at once

There are a number of deep graph generators to generate the entire molecular in one shot, which we call "All at once".

VGAE [Kipf and Welling, 2016] is a framework for unsupervised learning on graph-based data built upon variational autoencoder (VAE) [Kingma and Welling, 2014].The graph generator makes use of latent variables and learns interpretable latent representations to generate new molecular graph. Unlike VGAE, which can only learn from a single input graph, GraphVAE [Simonovsky and Komodakis, 2018] proposes another VAE-based generative model that can learn from a set of graphs. The encoder of GraphVAE uses a Graph Convolutional Network (GCN) [Zhang *et al.*, 2019] to embed the input graph into continuous representation $\mathbf{z}$ while the decoder of GraphVAE outputs a probabilistic fully-connected graph constrained by a predefined maximum size. RVAE [Ma *et al.*, 2018] is another AE-based model for molecule generation, which proposes a novel regularization framework to guarantee semantic validity. To generate graphs of larger size, a model named MPGVAE [Flam-Shepherd *et al.*, 2020] applies a message passing neural network (MPNN) [Gilmer *et al.*, 2017] to the encoder and decoder of a VAE, avoiding complex graph matching operations.

MolGAN [De Cao and Kipf, 2018] proposes an implicit generative model for small molecular graph which utilizes generative adversarial network (GAN) [Goodfellow *et al.*, 2014]. It first samples a latent vector $\mathbf{z}$ from $\mathcal{N}(0, 1)$, then it generates a graph all at once using an MLP. GraphNVP is the first known molecule generation model based on invertible normalizing flow. It performs Dequantization technique [Dinh *et al.*, 2017; Kingma and Dhariwal, 2018] to transform discrete adjecency tensor and node label matrix into continuous node features and then uses coupling layers to obtain latent representations $\mathbf{z} = \mathrm{Concate}(\mathbf{z}_{\mathbf{A}}, \mathbf{z}_{\mathbf{X}})$. After sampling a latent vector $\mathbf{z}$ from a known prior distribution $p_{\mathbf{z}}$ and splitting $\mathbf{z}$ into $\mathbf{z}_{\mathbf{A}}$ and $\mathbf{z}_{\mathbf{X}}$, GraphNVP takes two steps to generate a molecule. The first is generating a graph structure $\mathbf{A}$ from $\mathbf{z}_{\mathbf{A}}$ and the second is generating node attributes $\mathbf{X}$ based on the structure $\mathbf{A}$ from $\mathbf{z}_{\mathbf{X}}$.

3.2 Generation Strategy II: Fragments-based

Many models adopting rational substructures, also known as fragments have been proposed, as building blocks to generate high quality molecules, which are categorized as "Fragment-based" here. Among them, some are based on autoencoder framework. An earlier work proposes a model named JT-VAE [Jin *et al.*, 2018] which first decomposes the molecular graph G into its junction tree $\mathcal{T}_G$, where each node in the tree represents a substructure of the molecule. JT-VAE then encodes both the tree and graph into their latent embeddings $\mathbf{z}_{\mathcal{T}}$ and $\mathbf{z}_G$. As for decoding phase, JT-VAE first reconstructs junction tree from $\mathbf{z}_{\mathcal{T}}$ then generates molecule graph from the predicted junction tree by a graph decoder which learns how to assemble subgraphs. Another AE-based model is HierVAE [Jin *et al.*, 2020a] which proposes a larger and more flexible graph motifs as building blocks and achieves a higher reconstruction accuracy when facing larger molecule size. Different from JT-VAE, a molecule in HierVAE is represented by a hierarchical graph with three distinct layers namely atom layer, attachment layer and motif layer. The encoder generates the embedding of each node in all three layers while the decoder reconstructs molecule coarse to fine.

To generate valid molecular graph, MHG-VAE [Kajino, 2019] proposes molecular hypergraph grammar (MHG) to encode chemical constraints and both the encoder as well as the decoder consist of three parts. Specifically,

$$\mathrm{Enc} = \mathrm{Enc}_H \circ \mathrm{Enc}_G \circ \mathrm{Enc}_N, \tag{2}$$

where $f \circ g$ means $f(g(\cdot))$, Enc_N encodes a molecule graph into a molecule hypergraph, Enc_G converts hypergraph into a parse tree and Enc_N encodes the parse tree into the latent vector. The decoder acts as an inversion to the encoder. MoleculeChef [Bradshaw *et al.*, 2019] uses a vocabulary of common reactant molecules as building blocks to generate synthetic molecules. Gated graph neural networks (GGNNs) [Li *et al.*, 2015] are adopted to embed each reactant separately and the results are summed to generate one vector. Thus, the encoder realizes the mapping of a multi-set of reactants to a distribution over latent space. The decoder uses an RNN to generate a multi-set of reactants from the latent space. Specifically, the latent vector $\mathbf{z}$ initializes the hidden layer and the RNN model outputs one reactant or terminate at each generation step. Then the authors propose a reaction model to predict how these generated reactants react together to generate new molecules. Instead of VAE, the objective function of WAE is adopted by MoleculeChef to learn the model parameters, which involves minimizing:

$$L = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[c(\mathbf{x}, p(\mathbf{x}|\mathbf{z}))] + \lambda D(\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[q(\mathbf{z}|\mathbf{x})], p(\mathbf{z})),$$

where $c(\cdot)$ is a cost function and $D(\cdot)$ is a divergence measure namely maximum mean discrepancy (MMD).

There also exist several works using reinforcement learning (RL) to optimize the properties of generated molecules. RationaleRL [Jin *et al.*, 2020b] uses rationales as building blocks for molecule generation. The first step of RationaleRL is extracting rationales that are likely accountable for each property from molecules by MCTS [Chaslot, 2010] and combining them for multiple properties. Specifically, during search process, each state in the search tree means a subgraph of the molecule and the property score of the subgraph indicates the reward. Then RationaleRL uses graph generative models to expand the rationales into full molecules. To generate realistic compounds, the graph generator is trained in two phases, namely pre-training phase and fine-tuning phase. After pre-training on a large set of real molecules, the graph generator is fine-tuned on property-specific rationales through multiple iterations using policy gradient.

Similarly, in MolEvol [Chen *et al.*, 2020], the authors adopt an expectation maximization (EM)-like process for molecule optimization. The framework also contains two stages: rationales search stage and molecule completion stage. The proposed EM-like evolution-by-explanation algorithm alternates between these two stages. Specifically, MolEvol first identifies rationales by using an explainable local search method, then it explores higher scoring molecules according to rationale samples by using a conditional generative model.

Another RL framework FREED [Yang *et al.*, 2021a], which couples a fragment-based generation method and a novel error-prioritized experience replay (PER) to find chemically realistic and pharmacochemically acceptable molecules. FREED adopts Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018a; 2018b] as basic reinforcement learning strategy and takes three actions to decide where to attach, which fragment to attach, and which bond to attach respectively in each iteration to generate molecules iteratively. Notably, the second action depends on the first and the third action depends on the first two. PER is applied to optimize docking scores by encouraging exploration when taking the third action.

Another emerging line considers molecule generation as a sampling procedure. MARS [Xie *et al.*, 2021] adopts the general markov chain monte carlo (MCMC) [Geyer, 1992] sampling framework, and generates multi-objective drug molecules. It starts from an initial molecule $x^{(0)}$ from the molecular space $\mathcal{X}$, then iteratively samples a candidate molecule $x' \in \mathcal{X}$ from the proposal distribution $q(x'|x^{(t-1)})$, x' is either accepted or rejected based on an acceptance rate $A(x^{t-1}, x') \in [0, 1]$. Thus, a sequence of molecules $\{x^{(t)}\}_{t=0}^{\infty}$ is generated. The proposal distribution is represented by molecular graph editing actions including "adding a fragment" and "deleting a bond", which are formulated as follows, respectively,

$$\begin{aligned} q_{\mathrm{add}}(x'|x) =& \frac{1}{2} p_{\mathrm{add}}(u) p_{\mathrm{frag}}(k), \\ q_{\mathrm{del}}(x'|x) =& \frac{1}{2} p_{\mathrm{del}}(b), \end{aligned} \tag{3}$$

where $u \in [n]$ is an indicator of atoms in x, $k \in [V]$ is an indicator of fragments in the vocabulary of size V and $b \in [2m]$ is an indicator of bonds in x. MPNNs are used to predict the probability distribution $(p_{\mathrm{add}}, p_{\mathrm{frag}}, p_{\mathrm{del}}) = \mathcal{M}_\theta(x)$ where

$\mathcal{M}_\theta$ is a MPNN model parameterized by θ. Furthermore, MARS proposes to train the editing model adaptively by collecting training data from the sampling paths.

MIMOSA [Fu *et al.*, 2021] is another molecule generation approach built on the MCMC sampling framework. It first pretrains two GNNs for substructure-type prediction and molecule topology prediction respectively. Then it alternates between two stages: molecule candidate generation and molecule candidate selection. Specifically, MIMOSA generates molecule candidates by editing (add, delete and replace) current molecules under the prediction of two pretrained GNNs. For candidate selection, it assigns weights which can encode multiple constraints of interest for molecule candidates, thus Gibbs sampling (a particular type of MCMC) [Geman and Geman, 1984] can be used to choose those plausible molecules for next iteration.

Besides AE-based, RL-based and sampling-based methods mentioned above, some other methods also using fragments to generate molecules are proposed recently. Mol-CycleGAN [Maziarka *et al.*, 2020] adopts a CycleGAN-based [Zhu *et al.*, 2017] method which generates a new molecule Y with desired property based on an initial one X. During the training phase, it first encodes X and Y to latent vectors based on the method proposed in JT-VAE, then it learns the transformation $\mathrm{F} : X \rightarrow Y$ in latent space. As for generation, Mol-CycleGAN takes X as input and obtains its embedding by adopting the encoder of JT-VAE. Then the optimized molecule Y, which is structurally similar to X, is generated by the decoder based on $\mathrm{F}(X)$.

GFlowNet [Bengio *et al.*, 2021] views the generative process as a flow network and aims to generate a diverse set of trajectories with high returns. The nodes represent states, edges represent actions and the weight of an edge (i.e. flow) represents the probability of taking an action. GFlowNet formulates a generative policy that samples with a probability proportional to the given return function and trains a generative model by conforming flow-matching conditions. GFlowNet can be applied to molecule generation problem where the "state" is the current molecule, and the "action" is adding fragment from predefined fragments vocabulary to the current moelcule (as well as a stop action).

Modof [Chen *et al.*, 2021] only encodes the difference between molecules before and after optimization. To modify a molecule, it first decodes a vector sampled from the learned latent difference space to generate a fragment, then original fragments are removed and new fragments at the predicted disconnection site are added. The authors also propose Modof-pipe to modify a given molecule at multiple sites. It is further enhanced into Modof-pipe^m, which can modify a given molecule to multiple optimized molecules.

DEG [Guo *et al.*, 2022] proposes a data-efficient generative model learned from much smaller datasets which only contains $\sim 10^2$ samples. Given a set of molecular structures and a set of evaluation metrics, DEG learns a graph grammar that samples molecules maximizing the metrics and generates molecules from a sequence of production rules. DEG views each molecule as a hypergraph and the grammar construction iteratively creates production rules from subgraphs by contracting hyperedges based on a parameterized function $\mathcal{F}_\theta$.

3.3 Generation Strategy III: Node-by-node

Besides generating entire molecules directly and using substructures as building blocks, there are some other methods proposed in recent years generating molecules in a manner entitled "node-by-node". We introduce them here one by one.

CGVAE [Liu *et al.*, 2018] is an autoencoder-based generative model which builds GGNNs [Li *et al.*, 2015] into the encoder and decoder. CGVAE uses GGNN to embed each node in an input graph G to a latent vector sampled from a diagonal normal distribution. The decoder of CGVAE initializes nodes with latent variables and generates edges between these nodes sequentially based on two decision functions: *focus* and *expand*. Specifically, the *focus* function determines which node to visit and the *expand* function decides which edges to add from the focus node in each step. The procedure will terminate when meeting the stop criteria. Notably, during the generation, all node representations should be updated once the generated subgraph changes. Furthermore, valency masking is applied to *expand* function to ensure chemical validity. [Lim *et al.*, 2020] proposes another AE-based method which can generate molecules with target properties while maintaining an arbitrary input scaffold as a substructure. Its encoder adopts a variant of interaction network to encode a whole-molecule graph G into a latent vector $\mathbf{z}$. The decoder is trained to generate molecules from $\mathbf{z}$ by taking a scaffold S as input and sequentially adding nodes and edges to S based on three loop stages namely "node addition", "edge addition", "node selection" and a extra final stage named "isomer selection". Furthermore, this model can generate molecules with constraints by concatenating the constraint vector with $\mathbf{z}$ sampled from latent space in decoding phase.

GCPN [You *et al.*, 2018] proposes a stepwise approach for molecule optimization based on RL. GCPN considers the generation as a markov decision process (MDP) and a molecule is sequentially constructed by either adding a bond to connect existing atoms or connecting a new subgraph with current molecular graph. Specifically, at each generation step t, GCPN first computes the state s_t based on the current graph G_t and the set of scaffolds $\mathcal{S}$, then GCPN takes s_t as input and predicts an action a_t . Graph Convolutional Networks (GCN) [Zhang *et al.*, 2019] and Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] are used to embed nodes and optimize policy networks respectively during each action prediction procedure. Furthermore, GAN is applied to guarantee the generated molecules resembling a given set of molecules. Recently, a model named DeepGraphMolGen [Khemchandani *et al.*, 2020] further improves GCPN by adding a molecular property prediction network to GCPN. The prediction network consists of a Graph Convolutional Network as a feature encoder together with a feed-forward Network and applies an adaptive robust loss function to avoid potentially gross errors. In this way, GCPN achieves extra rewards of additional properties (an example provided is the binding potency of small molecules to dopamine transporters), thus DeepGraphMolGen can generate multi-objective molecules with desirable properties.

GraphAF [Shi *et al.*, 2020] is a flow-based model which takes the advantage of autoregressive method. First, it adopts Dequantization [Dinh *et al.*, 2017; Kingma and Dhariwal,

Model	Generation Strategy	Methodology	Involve Optimize?	Venue
VGAE	All at once	Autoencoder-based	✗	NeurIPS workshop [Kipf and Welling, 2016]
RVAE	All at once	Autoencoder-based	✗	NeurIPS [Ma *et al.*, 2018]
GraphVAE	All at once	Autoencoder-based	✓	ICANN [Simonovsky and Komodakis, 2018]
MPGVAE	All at once	Autoencoder-based	✓	Arxiv [Flam-Shepherd *et al.*, 2020]
MolGAN	All at once	GAN-based	✗	ICML workshop [De Cao and Kipf, 2018]
GraphNVP	All at once	Flow-based	✗	Arxiv [Madhawa *et al.*, 2019]
JT-VAE	Fragment-based	Autoencoder-based	✗	ICML [Jin *et al.*, 2018]
HierVAE	Fragment-based	Autoencoder-based	✓	ICML [Jin *et al.*, 2020a]
MHGVAE	Fragment-based	Autoencoder-based	✗	ICML [Kajino, 2019]
MoleculeChef	Fragment-based	Autoencoder-based	✗	NeurIPS [Bradshaw *et al.*, 2019]
RationaleRL	Fragment-based	RL-based	✓	ICML [Jin *et al.*, 2020b]
MolEvol	Fragment-based	RL-based	✓	ICLR [Chen *et al.*, 2020]
MARS	Fragment-based	Sampling-based	✓	ICLR [Xie *et al.*, 2021]
MIMOSA	Fragment-based	Sampling-based	✓	AAAI [Fu *et al.*, 2021]
FREED	Fragment-based	RL-based	✓	NeurIPS [Yang *et al.*, 2021a]
Mol-CycleGAN	Fragment-based	CyclaGAN-based	✗	J. Cheminformatics [Maziarka *et al.*, 2020]
GFlowNet	Fragment-based	Flow Network-based	✗	NeurIPS [Bengio *et al.*, 2021]
Modof	Fragment-based	Autoencoder-based	✓	Nature MI [Chen *et al.*, 2021]
DEG	Fragment-based	Graph Grammar-based	✓	ICLR [Guo *et al.*, 2022]
CGVAE	Node-by-Node	Autoencoder-based	✗	NeurIPS [Liu *et al.*, 2018]
Lim et al.	Node-by-Node	Autoencoder-based	✓	Chemical Science [Lim *et al.*, 2020]
GCPN	Node-by-Node	RL-based	✓	NeurIPS [You *et al.*, 2018]
DeepGraphMolGen	Node-by-Node	RL-based	✗	J. Cheminformatics [Khemchandani *et al.*, 2020]
GraphAF	Node-by-Node	Flow-based	✗	ICLR [Shi *et al.*, 2020]
GraphDF	Node-by-Node	Flow-based	✓	ICML [Luo *et al.*, 2021b]
STGG	Node-by-Node	Tree-based	✓	ICLR [Ahn *et al.*, 2022]

Table 1: Recent representative works of molecule generation and optimization.

2018] technique to preprocess a discrete graph $G = (\mathbf{A}, \mathbf{X})$ into continuous data $\mathbf{z} = (\mathbf{z}^{\mathbf{A}}, \mathbf{z}^{\mathbf{X}})$ where $\mathbf{A}$ is edge feature matrix and $\mathbf{X}$ is node feature matrix as mentioned before. Then based on Autoregressive Flows (AF) [Papamakarios *et al.*, 2017], it computes conditional distributions:

$$\begin{aligned} p(z_i^{\mathbf{X}}|G_i) =& \mathcal{N}(\mu_i^{\mathbf{X}}, (\alpha_i^{\mathbf{X}})^2), \\ p(z_{ij}^{\mathbf{A}}|G_i, \mathbf{X}_i, \mathbf{A}_{i,1:j-1} =& \mathcal{N}(\mu_{ij}^{\mathbf{A}}, (\alpha_{ij}^{\mathbf{A}})^2), \end{aligned} \tag{4}$$

where i is the generation step, μ and α are mean and standard deviation of a Gaussian distribution using different neural networks. A variant of relational GCN is applied to learn the embeddings of node and the current graph, which can be utilized in the computing of parameters of Gaussian distributions. As for generating new graph, GraphAF just samples random variables ϵ_i and ϵ_{ij} from base Gaussian distribution and converts them into the molecule structures as follows:

$$\begin{aligned} \mathbf{z}_i^{\mathbf{X}} =& \epsilon_i \odot \alpha_i^{\mathbf{X}} + \mu_i^{\mathbf{X}}, \\ \mathbf{z}_{ij}^{\mathbf{A}} =& \epsilon_{ij} \odot \alpha_{ij}^{\mathbf{A}} + \mu_{ij}^{\mathbf{A}}, \end{aligned} \tag{5}$$

where $\odot$ is an element-wise multiple operator and $\epsilon = \{\epsilon_1, \epsilon_2, \dots, \epsilon_n\} \bigcup \{\epsilon_{21}, \epsilon_{31}, \dots, \epsilon_{n,n-1}\}$ can be computed according to an invertible mapping to molecule structures $\mathbf{z}$. Furthermore, it realizes parallel training and can be fine-tuned with reinforcement learning to do molecule optimization.

Unlike GraphAF using continuous latent variables, another flow-based model named GraphDF [Luo *et al.*, 2021b] uses discrete latent variables to map graph nodes and edges based on invertible modulo shift transforms. Specifically, latent variables $\epsilon = \{\epsilon_1, \epsilon_2, \dots, \epsilon_n\} \bigcup \{\epsilon_{21}, \epsilon_{31}, \dots, \epsilon_{n,n-1}\}$ are all discrete and sampled from multinomial distributions, and the discrete transforms for generating new nodes and edges are:

$$\begin{aligned} \mathbf{z}_i^{\mathbf{X}} =& q_i^N \circ \cdots \circ q_i^1(\epsilon^i), \\ \mathbf{z}_{ij}^{\mathbf{A}} =& q_{ij}^N \circ \cdots \circ q_{ij}^1(\epsilon_i j), \end{aligned} \tag{6}$$

where $f \circ g$ means $f(g(\cdot))$, and

$$\begin{aligned} q_i^d(\epsilon) &= (\epsilon + \mu_i^d) \mod c, d = 1, \dots, N, \\ q_{ij}^d(\epsilon) &= (\epsilon + \mu_{ij}^d) \mod (b+1), d = 1, \dots, N, \end{aligned} \tag{7}$$

where c and b is the total number of node types and edge types, respectively. N is the number of modulo shift modules, μ_i^d and μ_{ij}^d are different functions. The use of discrete latent variables and discrete transforms above makes GraphDF different from other flow-based methods while the main procedure of generating molecules is similar to GraphAF.

A recent work presents a spanning tree-based graph generation framework entitled STGG [Ahn *et al.*, 2022], which considers molecule generation as a composition of a spanning tree and residual edges. STGG represents the molecule

Dataset	Description	Number of molecules	Link
QM9	Stable small organic molecules made up of CHONF atoms	$133,885$	http://quantum-machine.org/datasets/
GDB-17	Enumeration of small organic molecules up to 17 atoms	$> 166,000,000,000$	http://gdb.unibe.ch/downloads/
ZINC15	Commercially available compounds	$> 750,000,000$	http://zinc15.docking.org/
ChEMBL	Bioactive molecules with drug-like properties	$> 2,000,000$	https://www.ebi.ac.uk/chembl/
PubChemQC	Compounds with quantum chemistry estimated property based on density functional theory	$3,981,230$	http://pubchemqc.riken.jp/
DrugBank	FDA-approved drugs and other drugs public available	$> 14,000$	https://www.drugbank.ca/

Table 2: Representative datasets for molecule generation.

as a sequence of decisions $d = \{d_1, d_2, \cdots, d_T\}$ based on a proposed spanning tree-based grammar. These decisions have seven forms: *attach_atom*, *attach_bond*, *branch_start*, *branch_end*, *res_atom*, *res_bond*, and *terminate*. As for generating the sequence of decisions, it utilizes a tree-based transformer neural network [Vaswani *et al.*, 2017] together with relative positional encoding for tree generation, and a attention-based predictor for residual edge prediction. Furthermore, invalid decisions are masked out during the generation process to guarantee the validity of generated molecules.

3.4 Discussion

The generation at different levels can have different advantages in specific applications. While in general, the fine-grained level manipulation at node level is flexible, while it may be less efficient for generation and has difficulty in modeling higher-level (sub)structure information. While the fragment-based pipeline allows edition of sub-structures of a molecule, which can be often meaningful to some specific functionality and reaction. Finally, the once-for-all scheme can be efficient while it may sometimes lack enough flexibility for incremental generation and optimization.

4 Datasets and Evaluation Metrics

Datasets. We list representative publicly available datasets that are commonly used in molecule generation and optimization tasks as in Table 2. Among them, ChEMBL and DrugBank are dynamic databases that is often updated over time.

Evaluation Metrics. Generation and optimization for molecules adopt two different sets of evaluation metrics. Molecule generation evaluates the overall quality of generated molecules from a statistical perspective in terms of these metrics, including *validity* (the percentage of generated molecules that are chemically valid), *novelty* (the fraction of molecules not appearing in the training data), *diversity* (the pairwise molecular distance among generated molecules), *uniqueness* (ratio of unique molecules) and *reconstruction* (the percentage of molecules which can be reconstructed from their latent variables). As for molecule optimization task, it adopts another set of metrics for evaluation on the basis of multi-property of generated molecules , such as *QED* (quantitative estimate of drug-likeness) [Ertl and Schuffenhauer, 2009], *SA* (synthetic accessibility) [Bickerton *et al.*, 2012], *logP* (octanol-water partition coefficients) and so on.

5 Challenges and Future Directions

Although graph-based deep learning has achieved great success in the automation of molecule design, challenges still exist due to the complexity of molecular structure. In this section, we suggest three future directions for further research.

Polymers. Existing works mainly focus on small-molecule design. Their performance degrades significantly when applied to designing larger molecules like polymers. The failure is likely due to many generation steps required to realize larger molecules and the associated challenges with gradients across the iterative steps [Jin *et al.*, 2020a]. Therefore, new methods should be developed to cope with larger molecules.

3D Drug Discovery. Generating 3D molecular geometries remains under-explored currently. Compared to SMILES-based and graph-based representation, the additional dimension significantly expands the molecular space to be explored, which increases difficulty [Yang *et al.*, 2021b]. However, generating 3D molecules is meaningful and necessary due to 3D coordinates are important for accurate prediction of quantum properties [Schütt *et al.*, 2017]. Regardless of its significance, limited works are related to it, thus requiring further research efforts.

Target Discovery. Here it refers to producing likely associated drugs for a given disease. Chemical space is vast, but the subset of molecules with certain desirable properties is much smaller by contrast, e.g. activity against a given target, that makes them well suited for the discovery of drug candidates [Zhavoronkov *et al.*, 2019]. The core of target discovery lies in sampling compounds from promising regions of chemical space and screening them for activity against the biological target [Masuda *et al.*, 2020]. Finding small molecules that bind to a given target protein is a concrete challenging task in this field, which is also known as structure-based drug discovery. However, there is only a few examples of drug design on target protein [Luo *et al.*, 2021a; Luo and Ji, 2022; Yang *et al.*, 2020], leaving a challenge in future work.

6 Conclusion

Generating molecules with desirable properties is of fundamental significance, especially in drug industry. We have introduced a wide range of graph-based deep models and classified them into three categories according to their generation strategy. Public datasets and commonly utilized evaluation metrics are summarized. Finally, we also discuss challenges and future promising directions in this exciting area.

References

[Ahn *et al.*, 2022] S. Ahn, B. Chen, T. Wang, and L. Song. Spanning tree-based graph generation for molecules. *ICLR*, 2022.

[Alshehri *et al.*, 2020] A. S. Alshehri, R. Gani, and F. You. Deep learning and knowledge-based methods for computer-aided molecular design—toward a unified approach: State-of-the-art and future directions. *Computers & Chemical Engineering*, 2020.

[Anderson *et al.*, 1987] E. Anderson, G. D. Veith, and D. Weininger. *SMILES, a line notation and computerized interpreter for chemical structures*. 1987.

[Bengio *et al.*, 2021] E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio. Flow network based generative models for non-iterative diverse candidate generation. *NeurIPS*, 2021.

[Bickerton *et al.*, 2012] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 2012.

[Bradshaw *et al.*, 2019] J. Bradshaw, B. Paige, M. J. Kusner, M. Segler, and J. M. Hernández-Lobato. A model to search for synthesizable molecules. *NeurIPS*, 2019.

[Brown *et al.*, 2004] N. Brown, B. McKay, F. Gilardoni, and J. Gasteiger. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of chemical information and computer sciences*, 44(3):1079–1087, 2004.

[Chaslot, 2010] G. M. J.-B. C. Chaslot. *Monte-carlo tree search*. Maastricht University, 2010.

[Chen *et al.*, 2020] B. Chen, T. Wang, C. Li, H. Dai, and L. Song. Molecule optimization by explainable evolution. *ICLR*, 2020.

[Chen *et al.*, 2021] Z. Chen, M. R. Min, S. Parthasarathy, and X. Ning. A deep generative model for molecule optimization via one fragment modification. *Nature MI*, 2021.

[De Cao and Kipf, 2018] N. De Cao and T. Kipf. MolGAN: An implicit generative model for small molecular graphs. *ICML workshop*, 2018.

[Dinh *et al.*, 2017] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. *ICLR*, 2017.

[Elton *et al.*, 2019] D. C. Elton, Z. Boukouvalas, M. D. Fuge, and P. W. Chung. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, 2019.

[Ertl and Schuffenhauer, 2009] P. Ertl and A. Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 2009.

[Faez *et al.*, 2021] F. Faez, Y. Ommi, M. S. Baghshah, and H. R. Rabiee. Deep graph generators: A survey. *IEEE Access*, 2021.

[Flam-Shepherd *et al.*, 2020] D. Flam-Shepherd, T. Wu, and A. Aspuru-Guzik. Graph deconvolutional generation. *arXiv preprint arXiv:2002.07087*, 2020.

[Fu *et al.*, 2021] T. Fu, C. Xiao, X. Li, L. M. Glass, and J. Sun. Mimosa: Multi-constraint molecule sampling for molecule optimization. *AAAI*, 2021.

[Garcia Satorras *et al.*, 2021] V. Garcia Satorras, E. Hoogeboom, F. Fuchs, I. Posner, and M. Welling. E (n) equivariant normalizing flows. *NeurIPS*, 2021.

[Gebauer *et al.*, 2019] N. Gebauer, M. Gastegger, and K. Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *NeurIPS*, 2019.

[Geman and Geman, 1984] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 1984.

[Geyer, 1992] C. J. Geyer. Practical markov chain monte carlo. *Statistical science*, 1992.

[Gilmer *et al.*, 2017] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *ICML*, 2017.

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

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[Griffiths and Hernández-Lobato, 2020] R.-R. Griffiths and J. M. Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 2020.

[Grisoni *et al.*, 2020] F. Grisoni, M. Moret, R. Lingwood, and G. Schneider. Bidirectional molecule generation with recurrent neural networks. *Journal of chemical information and modeling*, 2020.

[Guo and Zhao, 2020] X. Guo and L. Zhao. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686*, 2020.

[Guo *et al.*, 2022] M. Guo, V. Thost, B. Li, P. Das, J. Chen, and W. Matusik. Data-efficient graph grammar learning for molecular generation. *ICLR*, 2022.

[Haarnoja *et al.*, 2018a] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, 2018.

[Haarnoja *et al.*, 2018b] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[Jensen, 2019] J. H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 2019.

[Jiménez-Luna *et al.*, 2020] J. Jiménez-Luna, F. Grisoni, and G. Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2020.

[Jin *et al.*, 2018] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *ICML*, 2018.

[Jin *et al.*, 2020a] W. Jin, R. Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. *ICML*, 2020.

[Jin *et al.*, 2020b] W. Jin, R. Barzilay, and T. Jaakkola. Multi-objective molecule generation using interpretable substructures. *ICML*, 2020.

[Kajino, 2019] H. Kajino. Molecular hypergraph grammar with its application to molecular optimization. *ICML*, 2019.

[Kang and Cho, 2018] S. Kang and K. Cho. Conditional molecular design with deep generative models. *Journal of chemical information and modeling*, 2018.

[Khemchandani *et al.*, 2020] Y. Khemchandani, S. O'Hagan, S. Samanta, N. Swainston, T. J. Roberts, D. Bollegala, and D. B. Kell. Deepgraphmolgen, a multi-objective, computational strategy for generating molecules with desirable properties: a graph convolution and reinforcement learning approach. *Journal of Cheminformatics*, 2020.

[Kim *et al.*, 2020] H. Kim, E. Kim, I. Lee, B. Bae, M. Park, and H. Nam. Artificial intelligence in drug discovery: a comprehensive review of data-driven and machine learning approaches. *Biotechnology and Bioprocess Engineering*, 2020.

[Kingma and Dhariwal, 2018] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 2018.

[Kingma and Welling, 2014] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *stat*, 2014.

[Kipf and Welling, 2016] T. N. Kipf and M. Welling. Variational graph auto-encoders. *NeurIPS*, 2016.

[Li *et al.*, 2015] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[Lim *et al.*, 2020] J. Lim, S.-Y. Hwang, S. Moon, S. Kim, and W. Y. Kim. Scaffold-based molecular design with a graph generative model. *Chemical science*, 2020.

[Liu *et al.*, 2018] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt. Constrained graph variational autoencoders for molecule design. *NeurIPS*, 2018.

[Luo and Ji, 2022] Y. Luo and S. Ji. An autoregressive flow model for 3d molecular geometry generation from scratch. *ICLR*, 2022.

[Luo *et al.*, 2021a] S. Luo, J. Guan, J. Ma, and J. Peng. A 3d generative model for structure-based drug design. *NeuIPS*, 2021.

[Luo *et al.*, 2021b] Y. Luo, K. Yan, and S. Ji. Graphdf: A discrete flow model for molecular graph generation. *ICML*, 2021.

[Ma *et al.*, 2018] T. Ma, J. Chen, and C. Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. *NeurIPS*, 2018.

[Madhawa *et al.*, 2019] K. Madhawa, K. Ishiguro, K. Nakago, and M. Abe. Graphnvp: An invertible flow model for generating molecular graphs. *arXiv:1905.11600*, 2019.

[Masuda *et al.*, 2020] T. Masuda, M. Ragoza, and D. R. Koes. Generating 3d molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442*, 2020.

[Maziarka *et al.*, 2020] Ł. Maziarka, A. Pocha, J. Kaczmarczyk, K. Rataj, T. Danel, and M. Warchoł. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 2020.

[Nigam *et al.*, 2020] A. Nigam, P. Friederich, M. Krenn, and A. Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *ICLR*, 2020.

[Papamakarios *et al.*, 2017] G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation. *NeurIPS*, 2017.

[Putin *et al.*, 2018] E. Putin, A. Asadulaev, Q. Vanhaelen, Y. Ivanenkov, A. V. Aladinskaya, A. Aliper, and A. Zhavoronkov. Adversarial threshold neural computer for molecular de novo design. *Molecular pharmaceutics*, 2018.

[Redkar *et al.*, 2020] S. Redkar, S. Mondal, A. Joseph, and K. Hareesha. A machine learning approach for drug-target interaction prediction using wrapper feature selection and class balancing. *Molecular informatics*, 2020.

[Roney *et al.*, 2022] J. P. Roney, P. Maragakis, P. Skopp, and D. E. Shaw. Generating realistic 3d molecules with an equivariant conditional likelihood model, 2022.

[Schulman *et al.*, 2017] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[Schütt *et al.*, 2017] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *NeurIPS*, 2017.

[Seff *et al.*, 2019] A. Seff, W. Zhou, F. Damani, A. Doyle, and R. P. Adams. Discrete object generation with reversible inductive construction. *NeurIPS*, 2019.

[Shi *et al.*, 2020] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *ICLR*, 2020.

[Simonovsky and Komodakis, 2018] M. Simonovsky and N. Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *ICANN*, 2018.

[Vaswani *et al.*, 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[Xie *et al.*, 2021] Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li. MARS: markov molecular sampling for multi-objective drug discovery. *ICLR*, 2021.

[Yang *et al.*, 2020] Y. Yang, R. Zhang, Z. Li, L. Mei, S. Wan, H. Ding, Z. Chen, J. Xing, H. Feng, J. Han, et al. Discovery of highly potent, selective, and orally efficacious p300/cbp histone acetyltransferases inhibitors. *Journal of medicinal chemistry*, 63(3):1337–1360, 2020.

[Yang *et al.*, 2021a] S. Yang, D. Hwang, S. Lee, S. Ryu, and S. J. Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation. *NeurIPS*, 2021.

[Yang *et al.*, 2021b] Y. Yang, S. Ouyang, M. Dang, M. Zheng, L. Li, and H. Zhou. Knowledge guided geometric editing for unsupervised drug design. *openreview.net*, 2021.

[You *et al.*, 2018] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *NeurIPS*, 2018.

[Zhang *et al.*, 2019] S. Zhang, H. Tong, J. Xu, and R. Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 2019.

[Zhavoronkov *et al.*, 2019] A. Zhavoronkov, Y. A. Ivanenkov, A. Aliper, M. S. Veselov, V. A. Aladinskiy, A. V. Aladinskaya, V. A. Terentiev, D. A. Polykovskiy, M. D. Kuznetsov, A. Asadulaev, et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 2019.

[Zhu *et al.*, 2017] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.

[Zhu, 2020] H. Zhu. Big data and artificial intelligence modeling for drug discovery. *Annual review of pharmacology and toxicology*, 2020.