ELSEVIER

DRUG DISCOVERY TODAY TECHNOLOGIES

Drug Discovery Today: Technologies

Vol. 32, No. 2019

Graph-based generative models for *de Novo* drug design

Xiaolin Xia, Jianxing Hu, Yanxing Wang, Liangren Zhang, Zhenming Liu*

State Key Laboratory of Natural and Biomimetic Drugs, School of Pharmaceutical Sciences, Peking University, Xueyuan Road 38, Haidian District, 100191 Beijing, China

The discovery of new chemical entities is a crucial part of drug discovery, which requires the lead compounds to have desired properties to be pharmaceutically active. *De novo* drug design aims to generate and optimize novel ligands for macromolecular targets from scratch. The development of graph-based deep generative neural networks has provided a new method. In this review, we gave a brief introduction to graph representation and graph-based generative models for *de novo* drug design, summarized them as four architectures, and concluded each's characteristics. We also discussed generative models for scaffold- and fragment-based design and graph-based generative models' future directions.

Section editors: Johannes Kirchmair – University of Vienna, Department of Pharmaceutical Chemistry, Althanstrasse 14, 1090 Vienna, Austria.

Introduction

The development of new technologies is always having a profound impact on the evolution of drug discovery [1]. Classical pharmacology [2], aka forward pharmacology, relies on screening *in vitro* or *in vivo* to identify substances with desirable therapeutic effects and to identify and validate targets. With the development of bioinformatics, especially after the sequencing of the human genome, reverse pharmacology [2], which usually identifies protein target first and performs the *in vivo* efficacy the last, has become popular.

As a reverse pharmacology method, *de novo* drug design is the design of bioactive compounds by incremental construction of a ligand model within a model of the receptor or enzyme active site, the structure of which is known from X-ray or nuclear magnetic resonance data (receptor-based design) or known ligands (ligand-based design) [3]. It has been estimated that the synthesizable chemical space might be as large as 10^{60}–10^{100} molecules, wherein 10^{23}–10^{60} [4] could be possible potential drug-like compounds, but only 10^{8}–10^{10} have been synthesized. High-throughput screening [5] and high-throughput virtual screening [6] can only search for the database part of the chemical space, while *de novo* drug design has the potentiality to discover new bioactive compounds. Generative modeling, which learns from the chemical databases and generates hypotheses for searching under the iceberg, can be viewed as a *de novo* drug design variation.

Recent success has proved deep learning to be applicable for reducing the time and cost of drug discovery [7]. Based on molecular graph representation that bridges between real molecules and the data format in computers for deep learning

*Corresponding author.: L. Zhang (liangren@bjmu.edu.cn), Z. Liu (zmliu@bjmu.edu.cn)

 https://doi.org/10.1016/j.ddtec.2020.11.004

models, graph-based generative models have shown great potentialities in *de novo* drug design.

This review is focused on graph-based generative models for *de novo* drug design. We first discuss representations of molecular structure. Secondly, we comprehensively review recent graph-based generative models sorted by architecture and make comparisons between them. Then we discuss models for scaffold- and fragment-based design. Finally, we summarize the graph-based generative models for *de novo* drug design and give a future perspective.

Molecular graph representation

There are many kinds of molecular representations in chemoinformatics that describe structural information, such as fingerprints, string representations, tensor representations (graph representation definition used in this review), 2D image [8], 3D grid of voxels, and 3D feature vectors.

Molecules are graph-structured data with atoms connected by chemical bonds. A common way of representing a mathematical graph is to use multidimensional arrays (tensors) to store the vertex type, edge type, and connectivity information. In tensor representation of graphs, vertexes (also called nodes) denote atoms, and edges denote bonds (Fig. 1c). Graph-based generative models we mention in this article take tensor representations and generate molecular graphs by recreating atom tensors and bond tensors. Depending upon the specific model, tensor representation of graphs can also be combined with other information such as physicochemical properties, scaffold conditions, or 3D restrictions. Nevertheless, discrete graph structures were once hard to generate [9]. It was not until 2018 that the first graph model for molecular generation was proposed and graph-based molecular generative models started developing rapidly (Fig. 2).

String representations are methods to represent a 2D molecule graph as a 1D text string, *e.g.*, the widely-used Simplified Molecular Input Line Entry Specification (SMILES) [10] and subsequently International Chemical Identifier (InChI) [11]. Most of the molecular datasets store molecular graphs in the form of SMILES, which are compact and contain complete structural information. Enlightened by natural language processing, SMILES have already been applied to molecular generation and is still a popular representation. However, chemoinformatics approaches usually start by recreating adjacency and bond order matrix from the SMILES, while bypassing this step in deep learning can lead to the most common drawback of SMILES-based models: grammar mistakes. It is hard to judge whether a molecular structure that the SMILES string representing would be chemical valid during SMILES generation (Fig.1b). Unclosed parentheses, unpaired ring numbering, and broken aromaticity, researchers have made great effort to solve these problems, designing modules to ensure precise chemical grammar directly or learn chemical grammar rules from molecule datasets indirectly [12,13].

Fingerprints such as the Molecular ACCess System (MACCS) keys [14] and extend connectivity fingerprint (ECFP) [15] have been developed for decades. These 1D bit strings perform well on structure-activity modeling and molecular similarity analysis but are not invertible (original molecular graph could not be reconstructed from its fingerprint), hence not quite suitable for molecular generation.

Molecules are 3D objects. However, to represent 3D coordinates, 3D voxels also go with much higher dimensionality, leading to massive training datasets in generative modeling. Another problem that 3D coordinates face is the invariance to molecular translation and rotation, let alone conformational change. Thus, 3D representations have not been widely applied in molecular generation. 3D feature vectors [16,17] deal with the invariance but having information loss, among which tensor field networks [18] can be inverted and are expected to have the potentiality for generative modeling.

Current graph-based molecular generative models

The fundamental goal of generative drug design is to propose optimized molecules that meet predefined criteria such as drug-likeness and activity. It could be hypothesized that each molecule from the training set is sampled from an unknown hidden distribution, and the generative problem then turns into a probability distribution fitting problem. Molecules with similar or optimized properties can be generated from the fitted distribution (Fig. 3a, b).

Different generative models take various strategies to fit the prior distribution. They can be categorized into four categories: (1) recurrent neural network (RNN) based models, (2) variational autoencoder (VAE) based models, (3) generative adversarial network (GAN) based models, and (4) newly emerged generative flow-based models (Fig. 4). Datasets, generative performance, and optimization tasks of each model are gathered in the table in SI.

It should be noted that each architecture has its pros and cons, and it is hard to say which one is the best. Different architectures could also be built-up together to cover each other's deficiency for better performance or broader application. However, the model architecture should be carefully designed since simply applying techniques such as reinforcement learning (RL) [41] might not improve the performance of generation or optimization [39].

RNN-based generative models

Recurrent neural networks (RNNs) [42] (Fig. 4a), which predict the next symbol by the current sequence and usually generate a molecular graph sequentially (Fig. 1b, c), have been widely used in natural language processing. This sequential molecular generative process is a Markov decision

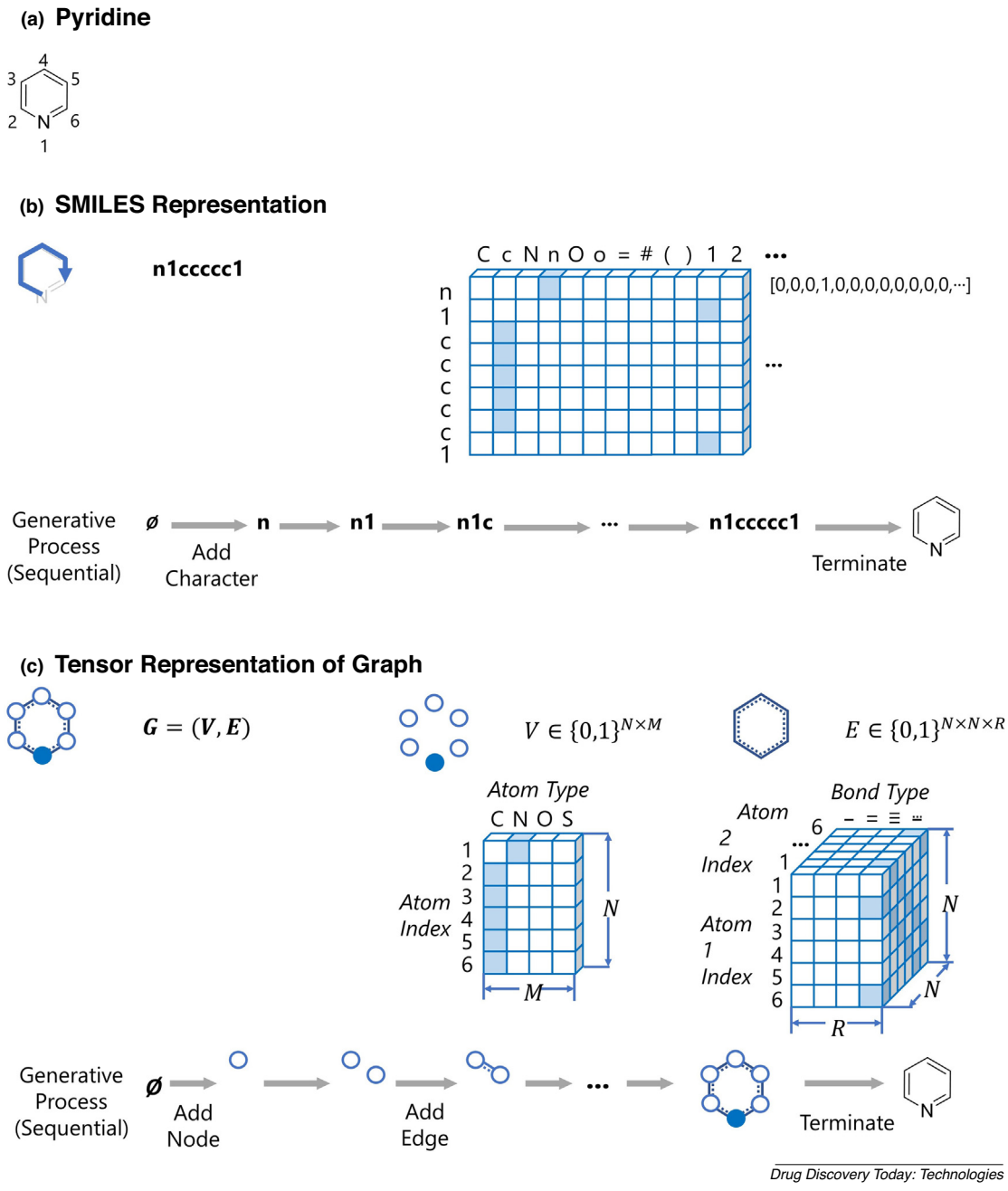

Fig. 1. Example of SMILES representation and graph representation and their sequential generating process. (a) Molecular graph of pyridine; (b) SMILES representation of pyridine, its one-hot format and typical generative process; (c) Tensor representation of pyridine's molecular graph in one-hot.

process (MDP). During training, RNN models learn whether to generate a new atom or bond and decide its type.

RNN-based models are computationally expensive, while they have reliable performance, producing molecules with relatively high valid and novel rates.

GraphNet [20] was the first RNN-based graph generative model, which generates a molecule with three operations: (1) add an atom, (2) connect the new atom to an existing atom, (3) add a new bond. Each step is modeled with a module with a defined probability distribution over possible outcomes: whether to do the operation (with which kind of atom/bond) or not. GraphNet demonstrated significant performance and outperformed contemporary SMILES-based models and VAE-based one-shot generative model GraphVAE, suggesting the potentiality of sequence-like graph knowledge representations for molecular generation.

Li et al. reported MolMP & MolRNN [22] after GraphNet when there were still few graph-based molecular generative models. They improved the generative process with also three

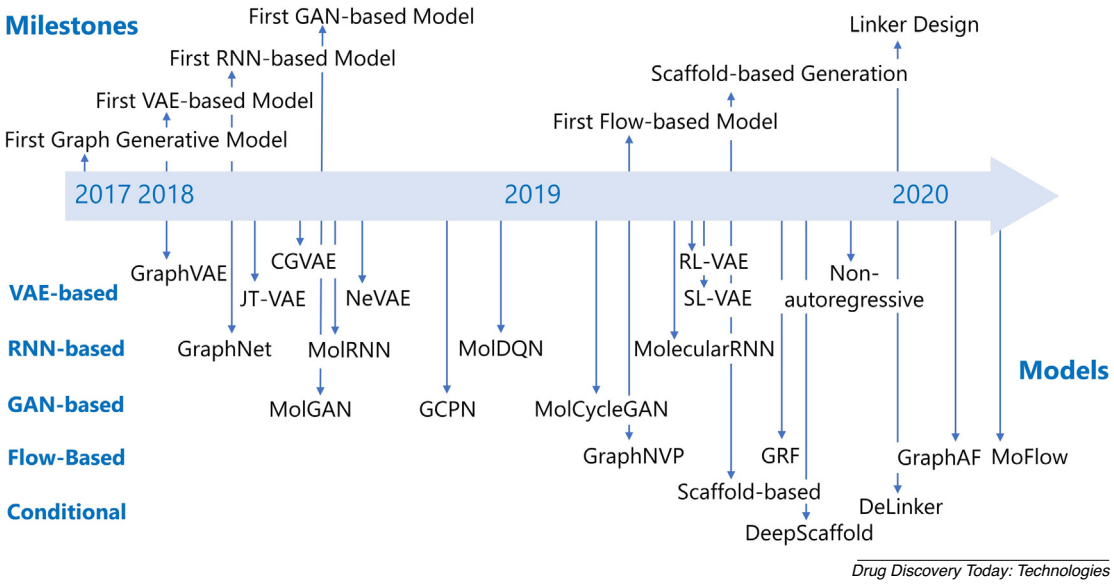

Fig. 2. Timeline of the development of graph-based generative models [19–40] for *de novo* drug design. Above time arrow: important milestones. Below time arrow: specific models.

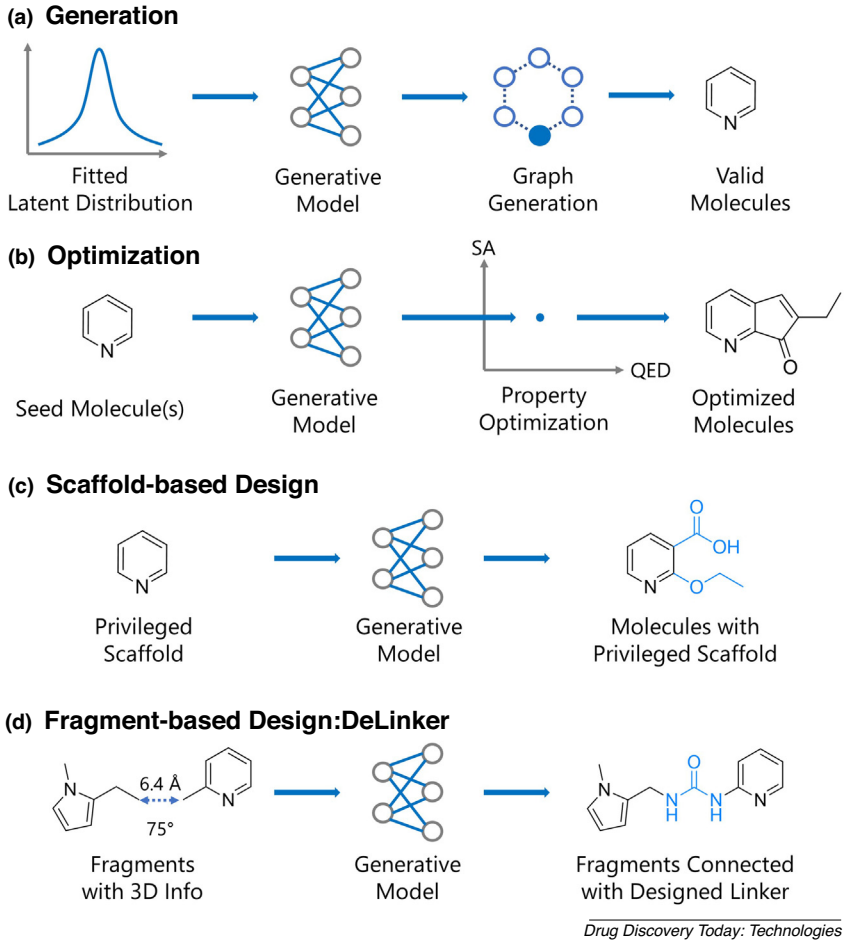

Fig. 3. Illustration of generation, optimization, and conditional generation process. (a) Molecular generation from the defined latent distribution. (b) Optimization to improve specific properties such as drug-likeness and synthetic accessibility. (c) Scaffold-based design. (d) Fragment-based linker design.

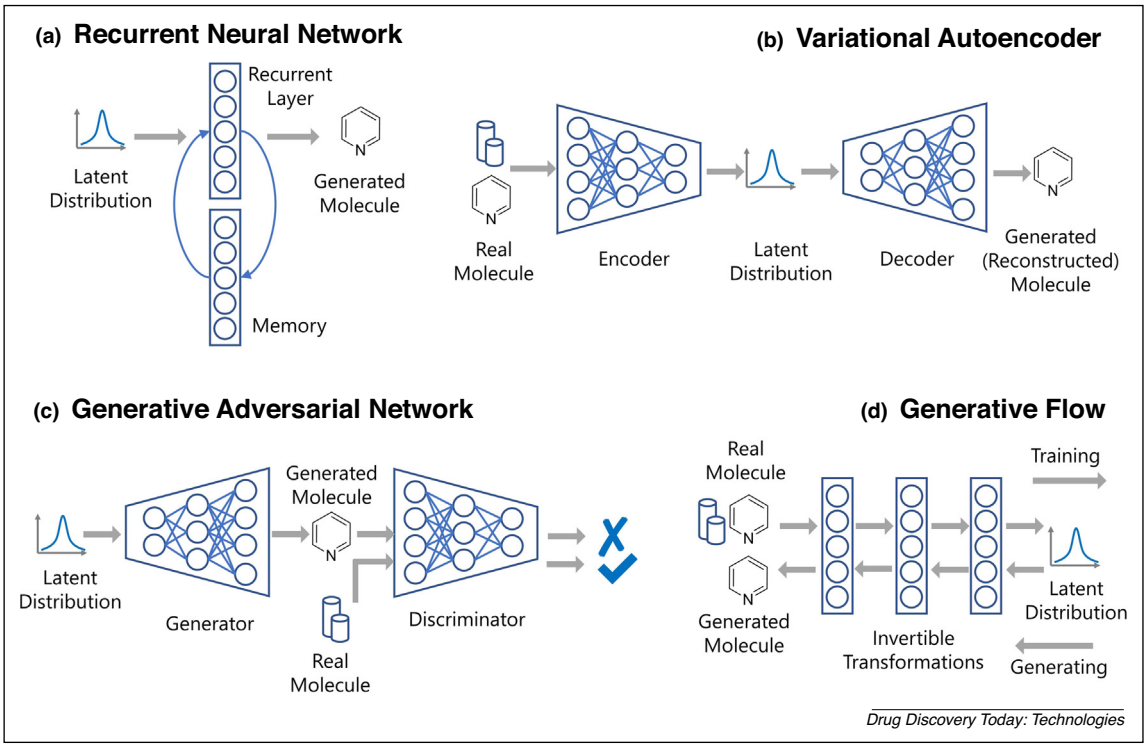

Fig. 4. An overview of current architectures of graph-based generative models. (a) Recurrent neural network; (b) Variational autoencoder; (c) Generative adversarial network; (d) Generative flow.

operations: (1) add an atom and connect it with an existing atom; (2) connect an existing atom to the new atom; (3) terminate the generating process. The generative process adjustment helps avoid repeated adding edge action and reduces total generative steps, which also contributed to the generating performance of larger molecules. Li et al. again proved that graph representations have an advantage over SMILES. When used to generate c-Jun N-terminal kinase 3 (JNK3) and Glycogen synthase kinase 3β (GSK-3β) dual inhibitors, the models achieved a significant enrichment rate and reproduced nearly 10% of actives without seeing them during training.

Disrespect of chemical constraints, especially valency, is a problem that molecular generative models face. Mariya et al. [38] found that the softmax layer prediction for the edge connecting step would always have nonzero values, which indicated that the model would still add bonds between atoms that had already been saturated with chemical bonds. Therefore, a valency-based rejection sampling is used in MolecularRNN, ensuring that the current sum of all bonds for each atom does not exceed the allowed valency when generating a new bond. An additional structure penalty is also introduced for atoms disrespecting valency, resulting in the modification of parameters to respect valency constraints. With these improvements, MolecularRNN does not break chemical valency and achieved a 100% valid rate.

VAE-based generative models

Variational autoencoders (VAEs) [43] encode molecules into compressed latent vectors obeying the latent distribution and decode variant molecules from the vectors [44] (Fig. 4b). Molecules from the training dataset are assumed to be generated by some random progress involving an unobserved continuous random variable, whose distribution is approximated by the latent distribution during training. Ideally, if the latent distribution perfectly fits hypothetical chemical space, VAEs can generate molecules with similar properties with any seed molecules and optimize them towards required properties.

A common drawback of VAE-based generative models is the bias in generated molecules, which is because they are trained by maximizing the evidence lower bound (ELBO) [45] of the marginal likelihood with Kullback-Leibler divergence [46]. Nevertheless, VAEs are stably optimized and can be visualized easily.

The first deep generative model for molecular graph generation was fulfilled with VAE, named GraphVAE [21]. Applying VAE to molecular graphs means graph-to-graph autoencoding that requires approximate graph matching, which has been done by a similarity function that enables the calculation of reconstruction loss in this work. GraphVAE showed good performance when generating small molecular graphs on the QM9 [47] dataset, but the results of the ZINC dataset (13.5% valid) [48,49] suggested that this earliest VAE-

based model with a one-shot generative strategy might not be capable enough for molecules with complicated structures. Succedent work such as NeVAE [25] takes advantage of RNN to generate molecular graph step-by-step. With the addition of masking, NeVAE reached a 100% valid rate. To generate a broader range of molecules with carbon ''mutating into'' other atoms such as nitrogen, NeVAE takes an embedding of atom types subject to Gaussian distribution.

Jin et al. [23] believe that atom by atom generative process would result in chemically invalid intermediaries and delay validation during the generative process. Instead, they took junction trees to represent the scaffold of chemical substructure components. Like a molecular graph that consists of atoms and bonds, a junction tree consists of building blocks and bonds. These building blocks are single rings extracted from the dataset or single atoms, and rings in a junction tree can overlap each other to form condensed rings. Thus, a molecule is represented by a common tensor representation of the molecular graph capturing fine-grained connectivity, plus a junction tree capturing coarse-grained connectivity. Since the molecular graph is assembled with valid chemical substructure building blocks during generation, the issue of broken aromaticity when generating atom by atom can be solved, and the model can generate molecules with more complicated structures especially condensed aromatic rings. This work again showed the superiority of graph tensor representations to SMILES representations.

GAN-based generative models

The generative adversarial network (GAN) [50] is an attractive architecture, including a generator that could be fulfilled with any learnable generative model and a discriminator that evaluates the generator's generated candidates (Fig.4c). GANs' training process is like a zero-sum game: in molecular generation, the discriminator is supposed to distinguish whether a molecule is from the training dataset (of merely valid molecules or molecules with specific properties), while the generator is to ''cheat'' the discriminator by generating molecules that pass for the real molecules. Ideally, the training ends in a draw: the discriminator perfectly discriminates fake molecules from real molecules from the training dataset, and the generator perfectly generates fake molecules identical to those in the training dataset. Nevertheless, GANs can practically suffer from mode collapse: the generator turns to generate simple structures that are easier to mix up with molecules from training datasets.

Compared to other architectures, GANs are harder to optimize and visualize, while they can take various generative processes and can be easily combined with wide-ranging properties required, such as aromaticity and halogen substitution.

Works of GANs at an early stage also take one-shot generative strategies that are most likely feasible only for graphs of small size. MolGAN [30] is a typical GAN-based generative model with a reward network and showed the potentiality of GANs combined with RL for molecular generation and optimization on small molecules of QM9. Follow-up work GCPN (graph convolutional policy network) [29] considered the molecular generative process as an MDP same to RNNs while also use RL for property optimization and showed better performance on larger molecules.

The GAN architecture demonstrated its potentiality in optimized generation towards specific properties in the work of MolCycleGAN [28]. MolCycleGAN is an impressive architecture for lead optimization, which extended the latent space of JT-VAE with CycleGAN (cycle-consistent generative adversarial network). Considering the desired property, *e.g.*, halogen substitution, molecules are divided into two sets: having halogen-substitution or not. The model consists of two GANs forming a cycle: one for generating halogen-substituted molecules from those without halogen substitution, the other is the opposite. The desired property can also be quantitative, *e.g.*, drug-likeness, water-solubility, inhibitory activity, or qualitative, *e.g.*, bioisosteric replacement and a specific number of aromatic rings. Although MolCycleGAN reached an overall physicochemical optimization performance close to previous work, it demonstrated greater potentiality for optimized generation: as long as the dataset can be divided into two large enough parts for training, MolCycleGAN can learn to optimize molecules to gain such desired properties.

Flow-based generative models

Flow-based generative models (generative flows) train an invertible transformation, also called normalizing flow [51] to approximate posterior distributions (Fig. 4d), and have been applied to graph generation recently in 2019 [52]. For flow-based models, the molecules are also assumed to be generated by some random progress involving an unobserved variable, where different from VAE is that flow-based model trains a sequence of invertible functions for generation: during generation, a latent vector is transformed into a tensor representing molecular graph with these functions, while during training, a molecular graph is transformed into the latent vector with inverse functions of these functions.

Flow-based models do not bring in bias in the transformation compared to VAEs and are more stable than GANs. The invertible process also provides convenience for molecular generation and optimization. Although flow-based models have no obvious advantage on generative and optimization performance than other architectures, they have shown great potentialities on computer vision tasks [53] and can be expected to contribute to molecular generation for *de novo* drug design in the future.

Current flow-based models mainly employ one-shot generative strategies. GraphNVP [33], the first invertible, normal-

izing flow-based generative model, outperformed early-stage one-shot models with VAE or GAN architecture. Follow-up work GRF (Graph residual flow) [34] takes a residual flow and reduces the number of trainable parameters to 1% of GraphNVP without reducing performance. Nevertheless, these models did not outperform sequential generative models from the same period. Subsequent work MoFlow [40] introduces a validity correction module to delete bonds from invalid atoms after generating a molecule at once, ensuring perfect generative performance. MoFlow also achieved state-of-the-art optimization performance. We expect these flow-based models will be extended to more applications in *de novo* drug design.

Scaffold-based and fragment-based design

In the development process of generative models (Fig. 2), the main objective of model design has gone through changes: to improve generative performance (Fig. 3a) in the early stage, to improve optimization performance (mainly on drug-likeness) from seed molecules (Fig. 3b) next, to meet specific demands in *de novo* drug design (Fig. 3c, d) recently. The following work starts with certain conditions corresponding to *de novo* drug design's specific situation: scaffold- and fragment-based design.

Scaffold-based generation is a trend of conditional generation since drug discovery often starts from specific core structures (namely scaffold) with high drug-likeness (Fig.3c). Lim et al. [35] proposed a generative model for scaffold conditioned molecular generation with certainty and multiple molecular properties optimized. The model consists of a typical VAE structure, while the scaffold of a molecule is taken in as a parametric argument, and the model completes the molecule based on the scaffold required. When used for scaffold-based epidermal growth factor receptor (EGFR) inhibitor design, this model improved the predicted pIC50 by 1.29. DeepScaffold [36] is another scaffold-based generative model reported simultaneously with similar ideas, while Li et al. included all subscaffolds in the dataset. Therefore, DeepScaffold starts with cyclic skeletons and can generate different classical scaffolds from the same cyclic skeleton. DeepScaffold was tested on three privileged scaffolds for G-protein coupled receptors (GPCRs) and recovered at least 5% actives and 10% known drugs without seeing them during training. These scaffold-based models can both be used for generating bioactive molecules from potential bioactive scaffolds.

Fragment-based drug discovery is another important part of *de novo* drug design, a common way of which is through linking strategy. DeLinker [37] was designed especially for linker generation of fragment-based design (Fig.3d). In this work, 3D structural information: distance and angle between the two fragments to be linked together, is used to initialize the generative process and guide the generation of every atom in the linker. After being trained to reconstruct original linkers, the model has learned the implicit relationship between the linker and its 3D structural information. Given the required distance and angle, DeLinker can generate novel linkers, which meet the structural condition better than database methods, for two initial fragments. Though there are still limitations for 3D molecular generation, this work is no doubt a practical trial.

Conclusion, discussion, and future perspectives

In this review, we have summarized recent graph-based generative models for *de novo* drug design. Difficulties of generating graph data and ensuring valid chemical structure have been overcome, and graph-based generative models have demonstrated their potentiality in molecular generation and *de novo* drug design. Graph representations have brought with feasibility to scaffold- and fragment-based design and overall better generative performance. Nevertheless, SMILES-based models still have their uniqueness in that they have a satisfactory performance with lower computational cost and can benefit from a more mature discipline of natural language processing for text generation.

Graph-based generative models also face general AI technology challenges, such as appropriate datasets and new hypotheses [54]. For example, almost all current graph-based generative models are 2D ligand-based methods, while in drug discovery, 3D structural information and protein-ligand interaction are quite significant. Better 3D representation methods, better architectures, and larger 3D datasets with higher quality are needed to practically generate 3D molecular structures. We expect future work on 3D feature vectors and generative models to provide new 3D molecular generation methods.

It is vital to apply generative models in specific situations in drug discovery. Deep learning has provided generative models with the fancy ability to learn from massive data and the potentiality above chemical intuition in *de novo* drug design. However, there is still more to be done. Firstly, synthetical accessibility [55] of *de novo* generative algorithms is one of the most fundamental challenges. Although some of the models mentioned above have been tested on *in silico* design tasks, only after synthesized can generated molecules head for a real drug. AI methods for retrosynthetic analysis and bioactivity prediction also need precise feedback from experimental data. Secondly, there are also other drug design methods that generative models could be applied to, such as multi-targeted drug design and pharmacophore-based design. Finally, difference in profession makes one feel worlds apart, and it will be hard for chemists and pharmacists who are not familiar with AI or deep learning to use raw generative models in practical work, so easy-to-use drug discovery toolkits packing these generative models are also needed.

Conflict of interest

The authors report no conflicts of interest. The authors alone are responsible for the content and writing of this article.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.ddtec.2020.11.004.

References

[1] Drews J. Drug discovery: a historical perspective. Science 2000;287 (5460):1960–4.
[2] Takenaka T. Classical vs reverse pharmacology in drug discovery. BJU Int 2001;88(Suppl (2)):7–10. discussion 49-50.
[3] Schneider G, Fechner U. Computer-based de novo design of drug-like molecules. Nat Rev Drug Discov 2005;4(8):649–63.
[4] Polishchuk PG, Madzhidov TI, Varnek A. Estimation of the size of drug-like chemical space based on GDB-17 data. J Comput-Aided Mol Des 2013;27(8):675–9.
[5] Macarron R, Banks MN, Bojanic D, Burns DJ, Cirovic DA, Garyantes T, et al. Impact of high-throughput screening in biomedical research. Nat Rev Drug Discov 2011;10(3):188–95.
[6] Pyzer-Knapp EO, Suh C, Gómez-Bombarelli R, Aguilera-Iparraguirre J, Aspuru-Guzik A. What is high-throughput virtual screening? A perspective from organic materials discovery. Ann Rev Mater Res 2015;45:195–216.
[7] Zhavoronkov A, Ivanenkov YA, Aliper A, Veselov MS, Aladinskiy VA, Aladinskaya AV, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat Biotechnol 2019;37(9):1038–40.
[8] Goh GB, Siegel C, Vishnu A, Hodas NO, Baker N. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. arXiv 2017. preprint arXiv: 170606689.
[9] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2020;1–21.
[10] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1):31–6.
[11] Heller SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D. InChI, the IUPAC international chemical identifier. J Cheminform 2015;7(1):23.
[12] Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular de-novo design through deep reinforcement learning. J Cheminform 2017;9(1):48.
[13] Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for molecule generation. Proceedings of the International Conference on Learning Representations 2018.
[14] Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci 2002;42(6):1273–80.
[15] Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model 2010;50(5):742–54.
[16] Eickenberg M, Exarchakis G, Hirn M, Mallat S. Solid harmonic wavelet scattering: predicting quantum molecular energy from invariant descriptors of 3d electronic densities. Adv Neural Inf Process Syst 2017;6540–9.
[17] Hirn M, Mallat S, Poilvert N. Wavelet scattering regression of quantum chemical energies. Multiscale Model Simul 2017;15(2):827–63.
[18] Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, et al. Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds; 2018.
[19] Johnson DD. Learning graphical state transitions; 2016.
[20] Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P. Learning deep generative models of graphs. International Conference on Learning Representations 2018.
[21] Simonovsky M, Komodakis N. Graphvae: towards generation of small graphs using variational autoencoders. In: International Conference on Artificial Neural Networks; 2018.p. 412–22.
[22] Li Y, Zhang L, Liu Z. Multi-objective de novo drug design with conditional graph generative model. J Cheminform 2018;10(1):33.
[23] Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. International Conference on Machine Learning 2018.
[24] Liu Q, Allamanis M, Brockschmidt M, Gaunt A. Constrained graph variational autoencoders for molecule design. Advances in neural information processing systems; 2018;p. 7795–804.
[25] Samanta B, Abir D, Jana G, Chattaraj PK, Ganguly N, Rodriguez MG Nevae: A deep generative model for molecular graphs. Proceedings of the AAAI Conference on Artificial Intelligence. 33. 2019:1110-1117.
[26] Kwon Y, Yoo J, Choi Y-S, Son W-J, Lee D, Kang S. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. J Cheminform 2019;11(1):70.
[27] Bresson X, Laurent T. A two-step graph convolutional decoder for molecule generation. arXiv 2019. preprint arXiv:190603412.
[28] Maziarka , Pocha A, Kaczmarczyk J, Rataj K, Danel T, Warcho M. Mol-CycleGAN: a generative model for molecular optimization. J Cheminform 2020;12(1):1–18.
[29] You J, Liu B, Ying Z, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. Adv Neural Inform Process Syst 2018;6410–21.
[30] De C Nicola, Kipf Thomas. MolGAN: an implicit generative model for small molecular graphs. Intl Conf Mach Learn 2018.
[31] Zhou Z, Kearnes S, Li L, Zare RN, Riley P. Optimization of molecules via deep reinforcement learning. Sci Rep 2019;9(1):10752.
[32] Shi C, Xu M, Zhu Z, Zhang W, Zhang M, Tang J. GraphAF: a flow-based autoregressive model for molecular graph generation. arXiv preprint arXiv:200109382 2020.
[33] Madhawa K, Ishiguro K, Nakago K, Abe M. GraphNVP: an invertible flow model for generating molecular graphs. arXiv 2019. preprint arXiv:190511600.
[34] Honda S, Akita H, Ishiguro K, Nakanishi T, Oono K. Graph residual flow for molecular graph generation. arXiv 2019. preprint arXiv:190913521.
[35] Lim J, Hwang S-Y, Moon S, Kim S, Kim WY. Scaffold-based molecular design with a graph generative model. Chem Sci 2020;11:1153–64.
[36] Li Y, Hu J, Wang Y, Zhang L-R, Zhou J, Liu Z-m. DeepScaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. J Chem Inf Model 2019;60(1):77–91.
[37] Imrie F, Bradley AR, van der Schaar M, Deane CM. Deep generative models for 3D linker design. J Chem Inf Model 2020;60(4):1983–95.
[38] Popova M, Shvets M, Oliva J, Isayev O. MolecularRNN: generating realistic molecular graphs with optimized properties. arXiv 2019. preprint arXiv:190513372.
[39] Kearnes S, Li L, Riley P. Decoding molecular graph embeddings with reinforcement learning. arXiv 2019. preprint arXiv:190408915.
[40] Zang C, Wang F. MoFlow: an invertible flow model for generating molecular graphs. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020. p. 617–26.
[41] Wiering M, Van Otterlo M. Reinforcement learning Adaptation learning and optimization, 12. 2012;p. 3.
[42] Irsoy O, Cardie C. Deep recursive neural networks for compositionality in language. Adv Neural Inf Process Syst 2014;2096–104.
[43] Pu Y, Gan Z, Henao R, Yuan X, Li C, Stevens A, et al. Variational autoencoder for deep learning of images, labels and captions. Adv Neural Inf Process Syst 2016;2352–60.
[44] Kingma DP, Welling M. Auto-encoding variational bayes. arXiv 2013. preprint arXiv:13126114.
[45] Yang X. Understanding the variational lower bound; 2017.
[46] Kullback S. Information theory and statistics. Courier Corporation 1997.
[47] Ramakrishnan R, Dral PO, Rupp M, Von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. Sci Data 2014;1140022.

[48] Sterling T, Irwin JJ. ZINC 15 – ligand discovery for everyone. J Chem Inf Model 2015;55(11):2324–37.
[49] Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG. ZINC: a free tool to discover chemistry for biology. J Chem Inf Model 2012;52 (7):1757–68.
[50] Goodfellow I. NIPS 2016 tutorial: generative adversarial networks. arXiv 2016. preprint arXiv:170100160.
[51] Rezende DJ, Mohamed S. Variational inference with normalizing flows. Proceedings of the 32nd International Conference on Machine Learning 2015;37:1530–8.
[52] Liu J, Kumar A, Ba J, Kiros J, Swersky K. Graph normalizing flows. Adv Neural Infor Process Syst 2019;13556–66.
[53] Shen Y, Qin J, Huang L. Invertible zero-shot recognition flows. European Conference on Computer Vision 2020;12361:614–31.
[54] Schneider P, Walters WP, Plowright AT, Sieroka N, Listgarten J, Goodnow RA, et al. Rethinking drug design in the artificial intelligence era. Nat Rev Drug Discov 2019;1–12.
[55] Gao W, Coley CW. The synthesizability of molecules proposed by generative models. J Chem Inf Model 2020.