Review

For reprint orders, please contact: reprints@future-science.com

Future Medicinal Chemistry

Deep learning for molecular generation

Youjun Xu‡,1, Kangjie Lin‡,2, Shiwei Wang3, Lei Wang1, Chenjing Cai1, Chen Song1, Luhua Lai1,2 & Jianfeng Pei*,1
[1]Center for Quantitative Biology, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, 100871, PR China
[2]BNLMS, State Key Laboratory for Structural Chemistry of Unstable & Stable Species, College of Chemistry & Molecular Engineering, Peking University, Beijing, 100871, PR China
[3]PTN Graduate Program, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, 100871, PR China
*Correspondence author. Tel.:/Fax: +86 10 6275 9669; jfpei@pku.edu.cn
‡Authors contributed equally

De novo drug design aims to generate novel chemical compounds with desirable chemical and pharmacological properties from scratch using computer-based methods. Recently, deep generative neural networks have become a very active research frontier in *de novo* drug discovery, both in theoretical and in experimental evidence, shedding light on a promising new direction of automatic molecular generation and optimization. In this review, we discussed recent development of deep learning models for molecular generation and summarized them as four different generative architectures with four different optimization strategies. We also discussed future directions of deep generative models for *de novo* drug design.

First draft submitted: 30 July 2018; Accepted for publication: 4 January 2019; Published online: 30 January 2019

Keywords: automatic molecular generation • deep generative neural networks • *de novo* drug design • molecular optimization

Computational *de novo* molecular design is, in theory, an important branch of drug design besides virtual screening and has made great contributions to drug discovery [1]. Many traditional structure-based *de novo* drug design methods, including our LigBuilder methods [2,3], have been developed in the past two to three decades to generate novel molecule entities with desired chemical and pharmacological properties from scratch. These methods were well introduced in some recent reviews [4,5].

Recently, deep generative models have been widely used and shown extraordinary results in various aspects, from realistic musical improvisation [6], to changing facial expressions in images [7], to creating realistic looking artworks [8], to translating between source images and target images [9–11]. The generative models have the state-of-the-art performance in representing and generating data in continuous domains. For more complex and discrete data type, such as arithmetic expressions [12], source code [13] and chemical molecules [14,15], there has been an increasing interest in developing generative models for generating realistic and valid data.

Progress in the development of deep generative models has spawned a range of promising proposals to address the issue of molecule generation, shedding light on a promising new direction of *de novo* drug design. Simplified molecular input line entry specification (SMILES) [16] representation of chemical compounds, a string-based representation derived from molecular graphs, was popularly used in this area. Recurrent neural networks (RNNs) [17] are ideal candidates for these representations and consequently, RNN-based generative models with on one-hot encoding were commonly adopted. Recently, with the progress in the area of deep learning on molecular graphs, training deep generative models directly on graphic representation became a feasible alternative.

This review will focus on deep generative models for *de novo* drug design. First, we will give a brief introduction on the common generative architectures including RNNs, autoencoders (AEs; VAE: variational AE; AAE: adversarial AE) [18–20] and generative adversarial networks (GANs) [21]. Second, we will comprehensively review recent developments on various generative models for creating molecules with desired properties based on SMILES and molecular graphic representation, including recent experimental validation. Third, we will summarize these generative models and make comparisons between them. Then, conditional optimization of properties, scaffolds

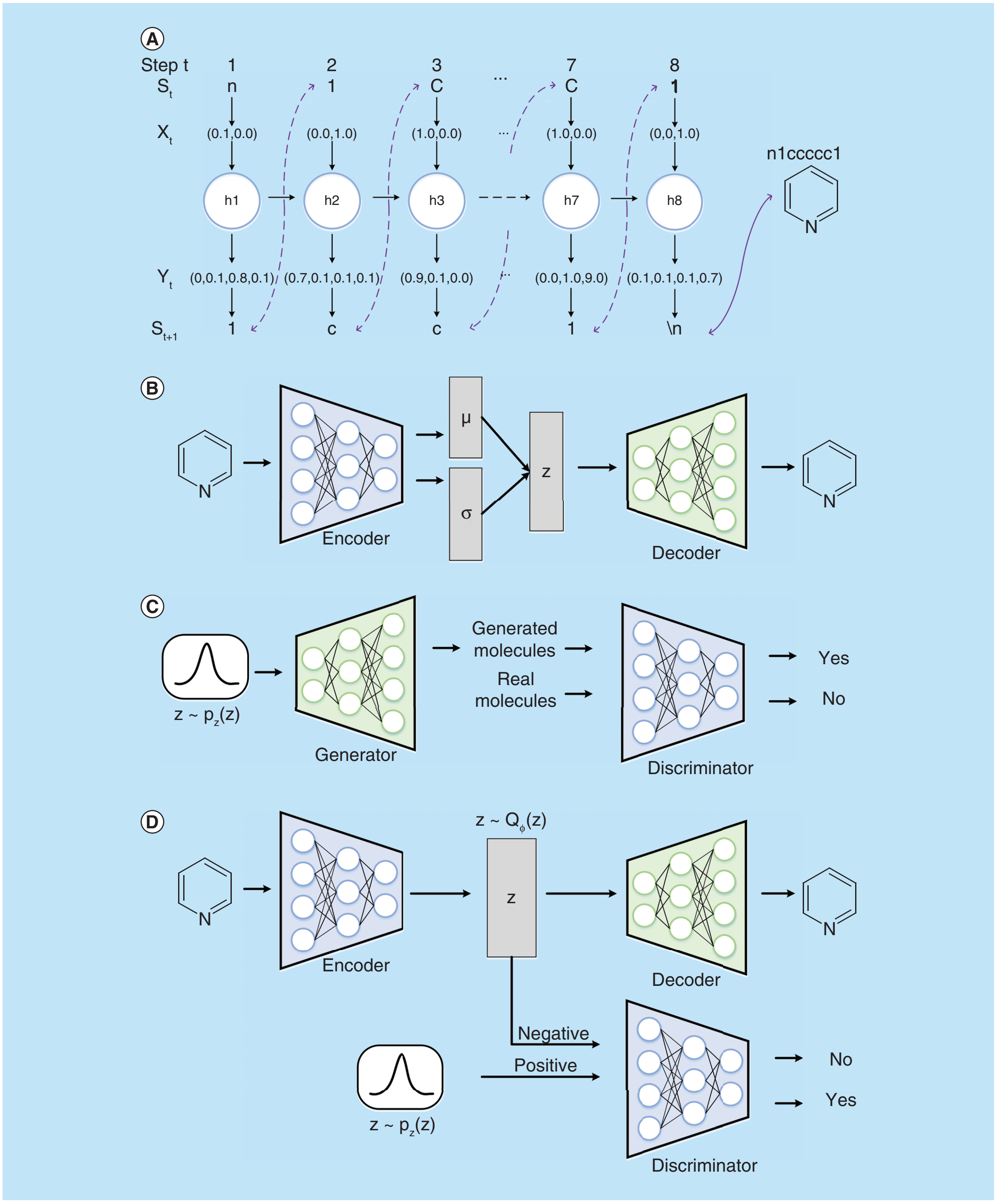

Figure 1. An overview of current architectures of generative neural networks. (A) Recurrent neural network; **(B)** variational autoencoder; **(C)** generative adversarial network; **(D)** adversarial autoencoder.

and targets will be discussed. Finally, we will make a future perspective on generative neural networks for drug discovery.

Architectures of generative neural networks

So far, the common generative architectures for molecular generation include RNN (Figure 1A), VAE (Figure 1B), GAN (Figure 1C), AAE (Figure 1D). These architectures are briefly introduced as follows.

Recurrent neural network

The RNN architecture basically unfolds over time (Figure 1A), which is specialized for processing a sequence of input vectors $\boldsymbol{x}_{1:n} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$. From an initial hidden state vector $\boldsymbol{h}_0$,

$$\boldsymbol{h}_i = f_r(\boldsymbol{h}_{i-1}, \boldsymbol{x}_i)$$

$$\boldsymbol{y}_i = f_o(\boldsymbol{h}_i)$$

In which, $\boldsymbol{h}_i$ is the hidden state vector at the time of i, which stores a representation of the information about all symbols seen in the sequence. f_r is a recursive function, and f_o is an output function. The RNN variants of long short-term memory (LSTM) and gated recurrent unit (GRU) aim to solve the vanishing gradient problem, which comes with a standard RNN. Details of the two variants can be referred to [22] and [23].

Variational autoencoder

This architecture (Figure 1B) is devised to learn a probabilistic generative model as well as its posterior, respectively known as decoder and encoder. We denote the observation as $\boldsymbol{x}$, and the continuous latent variable as $\boldsymbol{z}$. The decoder is modeling the probabilistic generative processed of $\boldsymbol{x}$ given the $\boldsymbol{z}$ through the likelihood $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, where θ is the learnable parameter. The encoder approximates the posterior with a model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ parameterized by ϕ. The encoder and decoder are learned simultaneously by maximizing the evidence lower bound (ELBO) of the marginal likelihood:

$$\text{ELBO}(\phi, \theta) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - \text{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x})\,||\,p(\boldsymbol{z})\right)$$

where, θ and ϕ are differential parameters, and KL is the Kullback–Leibler (KL) divergence. The evidence lower bound can be maximized via gradient ascent.

Generative adversarial network

It is an implicit generative model in the sense that it allows for inference of model parameters without requiring one to specify a likelihood. Seeing in Figure 1C, a GAN consists of two main components: a generative model G, which learns a map from a prior to the data distribution to sample new data points, and a discriminative model D, which learns to classify whether samples come from the real data distribution rather than from G. Those two models are implemented as deep neural networks and trained alternatively with stochastic gradient descent. G and D have different objectives, and they can be seen as two players in a minmax game

$$\min_{G}\max_{D} V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[\log \text{D}(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log(1 - D(G(\boldsymbol{z})))\right]$$

where G tries to generate samples to fool the discriminator and D tries to differentiate samples correctly.

Adversarial autoencoder

Inspired by VAE and GAN, AAE is proposed as a standard AE regularized by an adversarial learning (AL) procedure rather than a KL divergence penalty [20]. While the KL regularization in VAE is usually used to impose a prior distribution on the latent code $\boldsymbol{z}$, the AL regularization in AAE is utilized to match the posterior distribution to a prior distribution. Generally, the posterior distribution in VAE is usually a Gaussian distribution with mean and variance predicted by the encoder (depicted in Figure 1B). Unlike this, the posterior distribution $q_\phi(\boldsymbol{z})$ in AAE is encouraged to match a prior arbitrary distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$. As illustrated in Figure 1D, the AL network is devised attached on the latent code $\boldsymbol{z}$ for inducing $q_\phi(\boldsymbol{z})$ to match $p_{\boldsymbol{z}}(\boldsymbol{z})$. Regarding the encoder of the AE as a generator G, G tries to fool the discriminator D by mimicking $p_{\boldsymbol{z}}(\boldsymbol{z})$. Reconstruction and AL losses of the AAE are optimized alternatively with stochastic gradient descent.

$$\min_{G,\theta}\max_{D} \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(G(x))\right] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\left[\log(1 - D(\mathbf{z}))\right] - \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log p_\theta(\boldsymbol{x}|G(\boldsymbol{x}))\right]$$

Generator quality metrics

Evaluating the quality of molecule generator is of great significance for drug design. The common evaluation metrics are summarized as follows. Let n_s be the number of the randomly generated samples, V be the list of chemically valid molecules, X be the list of molecules from the given training set and T_d is Tanimoto distance based on extended-connectivity fingerprints (ECFP) [24]. Let Rs be the reconstruction fraction of generated molecules that are identical to a compared set of molecules.

$$Q_{\text{valid}} = \frac{|V|}{n_s}$$

$$Q_{\text{novel}} = 1 - \frac{|\text{set}\,(V) \cap X|}{|\text{set}\,(V)|}$$

$$Q_{\text{unique}} = \frac{|\text{set}\,(V)|}{|V|}$$

$$D_{\text{in}} = \frac{1}{|\text{set}\,(V)\,|^2} \sum_{(x,y)\in \text{set}(V)} T_d\,(x, y)$$

$$D_{\text{ex}} = \frac{1}{|\text{set}\,(V)\,|} \sum_{x\in \text{set}(V)} \min_{y\in X} T_d\,(x, y)$$

where, Q_{valid} is the chemically validity rate of generated molecules, Q_{novel} is the fraction of novel out-of-dataset molecules, Q_{unique} is the fraction of unique correct molecules, D_{in} (internal diversity) is the average T_d between generated molecules, and D_{ex} (external diversity) is the average T_d between a generated molecule and its nearest neighbor in the training set.

Additionally, a good generator should correctly model the distributions of important molecular properties. Therefore, molecular properties of the generated dataset (p_G) should be analyzed, and comparisons with the training set (p_{data}) can be performed using KL divergence (D_{KL}) and Jensen–Shannon divergence (D_{JS}).

$$D_{\text{KL}}\,(p_G||p_{\text{data}}) = \sum_{x\in \text{set}(V)} p_G\,(x) \log \frac{p_G\,(x)}{p_{\text{data}}\,(x)}$$

$$D_{\text{JS}}\,(p_G||p_{\text{data}}) = \frac{1}{2} D_{\text{KL}} \left(p_G || \frac{p_G + p_{\text{data}}}{2} \right) + \frac{1}{2} D_{\text{KL}} \left(p_{\text{data}} || \frac{p_G + p_{\text{data}}}{2} \right)$$

Inspired by Fréchet Inception Distance [25], a novel metric for evaluating both structure-level diversity and property-level similarity was proposed by Heusel *et al.*, called Fréchet ChemNet Distance [26]. Mathematically, the form is defined by:

$$d^2\,((m, C)\,,\ (m_w,\ C_w)) = ||m - m_w||_2^2 + \text{Tr}\left(C + C_w - 2\,(\text{CC}_w)^{\frac{1}{2}} \right)$$

where (m, C) and $(m_w,\ C_w)$ represent the mean and covariance of samples from real molecules and generative model, respectively.

Generative models for creating molecules

Generally, there are usually two types of molecular inputs for deep generative models, a linear notation of SMILES [16] and a 2D undirected molecular graph. We will review the published generative models (SMILES- and graph-based generative models) based on four different generative architectures (RNN, VAE, GAN & AAE), with four different optimization strategies (transfer learning [TL] [14], Bayesian optimization [BO] [15], reinforcement learning [RL] [27] and conditional generation [CG] [28,29]). TL is a simple and popular strategy to learn general features on the bigger dataset, which also might be useful for the second task in the smaller data regime. A model is first trained on a large dataset, and the smaller dataset with certain desirable properties can be used to fine-tune this model to generate more biased molecules. BO is a sequential design strategy for global optimization of black-box functions. A Gaussian process (GP) [30] can be used to model any smooth continuous function with few parameters. Given the latent space representation of the molecules as an input, the GP is trained to construct an acquisition function, which evaluates target properties and determines what the next query point should be. This strategy is only well suitable for the expressive AE-based models with a smooth latent space. RL is a joint training strategy of molecular generation with some reward objectives. When a generator model is trained to create molecules, it is simultaneously rewarded by a reward function, which quantifies target properties. Maximizing the reward function is equal to optimizing molecular properties. The idea of CG is based on conditional generative models [31], which is proposed for generation tasks with specific requirements. The given requirements can be converted into a conditional code, which is concatenated to the original input vector for molecular generation. And CG can be easily applied to multiobjective optimization.

SMILES-based generative models

The SMILES string, like natural language, is a popular molecular representation method among molecular generative models. For the SMILES-based generative models, one-hot encoding of a string is fed into a model for learning, then generates a string char-by-char based on the learned distribution.

RNN-based generative models with TL

Case 1

Segler *et al.* [14] first developed an RNN generative model for *de novo* molecular generation. They utilized three stacked LSTM layers to construct the model architecture. The SMILES strings are transformed as one-hot representations. The architecture can predict current state vectors and return output vectors after feeding the model with input vector and initial state vector. As can be imagined, the properties of the generated structures are particularly correlated with the training molecules.

The canonical SMILES strings of 1.4 million molecules from the ChEMBL dataset [32] were used to train a general generative model. The 50,000,000 SMILES symbols were sampled from the general model symbol-by-symbol, leading to 976,327 samples. The quality with 97.7% of Q_{valid} and 89.4% of Q_{novel} suggested this model has learned the syntax rules of SMILES representations. Then, on the concept of TL [33], they fine tuned the general model for creating novel structures with desirable activity against a specific biological target. Using 5-HT_{2A} receptor (protein) [34], *Plasmodium falciparum* (parasite) [35] and *Staphylococcus aureus* (bacteria) [36] as three representative biological targets, they developed three specific target prediction models (TPMs) with ECFP4 (ECFP with a distance of 4). The higher the number of epochs with fine-tuning the general model, the higher the fraction of produced potential actives with high similarity. With fine-tuning the given active data (1239 and 1000 molecules), the model could reproduce 28 and 14% of the unseen test active datasets (1240 and 6051) for *P. falciparum* and *S. aureus,* respectively. In this case, they proposed a *de novo* design cycle scheme by iterating molecule generation ('synthesis'), selection of the best molecules with TPM ('virtual assay') and retraining the general model with the best molecules ('design'). In the case of *S. aureus,* the self-learning model could recreate 6% of the unseen test active molecules, even without a set of known actives to fine-tune. It is worth noted that the TPM may also be an external docking program or self-defined model.

Case 2

Bjerrum *et al.* [37] proposed a similar LSTM-based RNN model combined with a sampling temperature, which rescales the probability distribution of output sequences. The two datasets, fragment-like and drug-like training set, extracted from the ZINC database [38], contained 1,611,889 and 13,195,609 molecules, respectively. They were used to develop the two types of generative models. From Table 1, 50,000 samples were generated for both trained models at a sampling temperature of 1.0. Regarding the generative efficiency, the Q_{valid} is both about 98.0%, and the Q_{novel} is 63.0% and 83.0% in the generated fragment- and drug-like molecules, respectively. The distributions of molecular properties (molar weight**,** predicted logP [logP] [39], number of hydrogen bond acceptors and donors,

Table 1. A summary of simplified molecular input line entry specification-based generative models.

Model type	Model name	Dataset	Molecular size	No. of molecules	No. of generated molecules	Generative performance	Optimization task	Optimization performance	Ref.
RNN-only-based model	Segler *et al.* [14]	ChEMBL	No report	1.4 million	976,327	97.7% (V), ~89.4% (N)	1) 5-HT2A; 2) *Plasmodium falciparum*; 3) *Staphylococcus aureus*	1) >50% for predictive actives; 2) 28% (test Rs), 66.9 (EOR); 3) 14% (test Rs), 155.9 (EOR)	No report
	Bjerrnum *et al.* [37]	ZINC	No report	1,611,889/ 13,195,609	50,000	Fragment: ~98.0% (V), 63.0% (N); Drug like: ~98.0% (V), 83.0% (N)	Retro-synthetic routes of easy/medium/hard group (SAS: 1.9/2.7/3.7)	1) Yield 55.7%/35.2%/26.1%; 2) cost of 1043.8/1690.8/1717.3 US$/100 g; 3) max steps of 2.2/3.7/4.4	No report
	Gupta *et al.* [42]	ChEMBL	34 to 74 characters	541,555	30,107	93% (V), 92% (N)	1) PPARs, Trypsin; 2) Fragment-growing; 3) TRPM8	1) 93–96%(V), 87–88% (N); 2) 97% (V); 3) 81% (N)	No report
RNN-based model with RL	REINVENT [27]	ChEMBL	10 to 50 heavy atoms	1.5 million	12,800	94.0% (V), 90.0% (N)	1) Sulfur-free; 2) Drug analog generation; 3) DRD2 activity model	1) 98% (V); 2) Found celecoxib analog; 3) test AUC of 1, 99.0% (V), >96.0% for predictive actives, 13% (test Rs)	[43]
	ReLeaSE [44]	ChEMBL	No report	1.5 million	1 million	95% (V), 95.3% (N for ZINC), 98.5% (N for ChEMBL)	1) T_{melt}; 2) LogP; 3) Inhibitor of JAK2; 4) Substructure bias	1) 31–53% (V), 95–99% (N); 2) 45–60% (V), 95–99% (N); 3) 70% (V), 92–94% (N); 4) 80–83% (V), 92–99% (N)	[45]
	ChemTS [46]	ZINC	No report	250,000	No report	No report	Penalized logP	An optimization speed of 41 ± 1.6 molecules/min; the best value: 6.56	[47]
RNN- and AE-based model	ChemVAE [15]	QM9/ZINC	<34/<121 characters	108,000/250,000	No report	1) ZINC (8728): 2.67 (avg. logP), 3.18 (avg. SAS), 0.70 (avg. QED), 5.8% (ZINC Rs), 0.15, 0.054 (test MAE of logP, QED), -1.812 (LL), 1.504 (RMSE); 2) QM9 (2839): 0.3 (avg. logP), 4.34 (avg. SAS), 0.47 (avg. QED), 0% (ZINC Rs), 0.16, 0.16, 0.21 (test MAE of HOMO, LUMO, Gap)	For ZINC: 1) 5 × QED - SAS; 2) Penalized logP (44)	1) BO is a better optimization strategy; 2) the best value: 1.98	[48]

V, U, N represent the mentioned Q_{valid} Q_{unique} and Q_{novel} respectively. Rs means the reconstruction rate against the test set. nQED, nlogP, nSAS are the normalization of QED, logP and SAS. EOR is the enrichment over random for the generated molecules that meets the requirements. MCF is the number of generated molecules filtered by medicinal chemistry filters.
AAE: Adversarial autoencoder; AE: Autoencoder; GAN: Generative adversarial network; avg.: Average; BO: Bayesian optimization; D_{ex}: External diversity; D_{in}: Internal diversity; E: Binding energy; HUMO: Highest occupied molecular orbital; IDC: Internal diversity clustering; IS: Internal similarity; LL: Log-likelihood; LUMO: Lowest unoccupied molecular orbital; MAE: Mean average error; MU: Muegge druglikeness filter; PPAR: Peroxisome proliferator-activated receptor; QED: Quantitative estimate of drug-likeness; RL: Reinforcement learning; RMSE: Root mean square error; RNN: Recurrent neural network; SAS: Synthetic accessibility scoring; T_{melt}: Melting temperature.

Table 1. A summary of simplified molecular input line entry specification-based generative models (cont.).

Model type	Model name	Dataset	Molecular size	No. of molecules	No. of generated molecules	Generative performance	Optimization task	Optimization performance	Ref.
	Grammar VAE [49]	ZINC	<39 heavy atoms	250,000	100,000	53.7% (Rs), 7.2% (V), -1.739 (LL), 1.404 (RMSE)	Penalized logP	The best value: 2.94	[50]
	SD-VAE [51]	ZINC	<39 heavy atoms	250,000	100,000	76.2% (Rs), 43.5% (V), -1.697 (LL), 1.366 (RMSE)	Penalized logP	The best value: 4.04	[52]
	AAE [53]	ChEMBL	<121 characters	~1.3 million	No report	Gaussian AAE: 77.4% (V) Uniform AAE: 78.3% (V)	1) Drug analog generation; 2) DRD2 activity model	1) Found celecoxib analog; 2) Avg. predicted score of >0.95; 11.5% (D_{ex} >0.35)	No report
	ECAAE [54]	ZINC	<58 characters	1.8 million	10,000	No report	1) Structural analogs; 2) logP, SAS; 3) logP, SAS, E; 4) JAK3	1) 93.6%, 3.28% for Tanimoto similarity and hamming distance; 2) ρ: 0.613 and 0.413; 3) ρ : 0.804, 0.593 and 0.406; 4) IC_{50} = 6.73 uM	No report
GAN- and RNN-based models with RL	ORGAN [55]	QM9	<52 characters	5000	No report	80.3% (V), 0.61 (D_{in}), 0.49 (nQED), 0.25 (nSAS), 0.31 (nlogP)	1) nQED; 2) nSAS; 3) nlogP; 4) All the above three	1) 88.2% (V), 0.55 (D_{in}), 0.52 (nQED); 2) 96.5% (V), 0.92 (D_{in}), 0.83 (nSAS); 3) 94.7% (V), 0.76 (D_{in}), 0.55 (nlogP) 4) 96.1% (V), 0.923 (D_{in}), 0.52 (nQED), 0.71 (nSAS), 0.53 (nlogP)	[56]
	ORGANIC [57]	QM9/ZINC	<10 heavy atoms/no report	5000/15,000	No report	0.2–99.9% (V), 86% (N), 0.1–0.4 (nQED)	1) nQED; 2) RO5	1) 0.8–0.9; 2) No improvement	[58]
	ATNC [59]	ChemDiv	<91 characters	15,000	1) 157,986 (IS); 2) 101,652 (IDC); 3) 176,342 (MU); 4) 156,605 (SP3)	1) 72% (V), 77% (N) 2) 71% (V), 86% (N) 3) 74% (V), 76% (N) 4) 74% (V), 73% (N)	1) D_{in}; 2) No. of unique heterocycles; 3) No. of clusters; 4) No. of singletons; 5) No. of MCF; 6) *In vitro* validation	1) 0.87; 2) 2782–3287; 3) 1219–1285; 4) 12,110–13,493; 5) 916–1213 (among 30,000 molecules); 6) 7 of 50 with inhibiton potency	No report
	RANC [60]	ChemDiv/ ZINC	<91characters/ no report	15,000/15,000	896,000/320,000	58%/76% (V), 48%/76% (N)	1) D_{in}; 2) No. of fragments; 3) No. of clusters; 4) No. of singletons; 5) No. of heterocycles; 6) No. of cluster size; 7) No. of MCF	1) 0.86/0.85; 2) 256,000/341,000; 3) 704/434; 4) 12,797/16,330; 5) 2286/1277; 6) 10.2/8.5; 7) 7.1/23.7; (ChemDiv/ZINC, both 20,000 molecules)	No report

V, U, N represent the mentioned Q_{valid} Q_{unique} and Q_{novel} respectively. Rs means the reconstruction rate against the test set. nQED, nlogP, nSAS are the normalization of QED, logP and SAS. EOR is the enrichment over random for the generated molecules that meets the requirements. MCF is the number of generated molecules filtered by medicinal chemistry filters.
AAE: Adversarial autoencoder; AE: Autoencoder; GAN: Generative adversarial network; avg.: Average; BO: Bayesian optimization; D_{ex}: External diversity; D_{in}: Internal diversity; E: Binding energy; HUMO: Highest occupied molecular orbital; IDC: Internal diversity clustering; IS: Internal similarity; LL: Log-likelihood; LUMO: Lowest unoccupied molecular orbital; MAE: Mean average error; MU: Muegge druglikeness filter; PPAR: Peroxisome proliferator-activated receptor; QED: Quantitative estimate of drug-likeness; RL: Reinforcement learning; RMSE: Root mean square error; RNN: Recurrent neural network; SAS: Synthetic accessibility scoring; T_{melt}: Melting temperature.

number of rotatable bonds and topological polar surface area) were compared between the generated and training molecules. The well-matched distributions for all the calculated properties reflected that the models only generated molecules in the vicinity of the training set, which was a small expansion in chemical space. It could still help to generate larger *in silico* libraries with similar properties. Synthetic feasibility was evaluated by synthetic accessibility scoring (SAS) [40] and retrosynthetic analysis with Wiley ChemPlanner [41]. The synthetic feasibility of the generated molecules was well correlated with their SAS scores. Retrosynthetic routes for the majority of generated molecules were identified by ChemPlanner, which was extremely helpful to automatically create novel molecules.

Case 3

Another similar LSTM-based RNN model with different applications was also developed by Gupta *et al.* [42] based on 541,555 molecules from the ChEMBL dataset [61]. As shown for Table 1, they sampled 30,107 strings, leading to the Q_{unique} of 93% and the Q_{novel} of 92%. Considering TL, three specific ligand subsets: 367 peroxisome proliferator-activated receptor γ (PPARγ) inhibitors; 1490 trypsin inhibitors; and five structurally diverse transient receptor potential M8 (TRPM8) blockers, were used to fine-tune the pretrained model, respectively. For the PPARγ, among a set of 1000 generated samples, 96% were chemically valid with the Q_{unique} of 90% and the Q_{novel} of 88%. For the 1000 generated samples toward the trypsin inhibitors, 93% were valid with the Q_{unique} of 87% and the Q_{novel} of 93%. For the 'low-data' situation of TRPM8 blockers, the fined-tuned model generated 100 molecules with the Q_{novel} of 94% and the Q_{unique} of 81%, and this model showed potentially useful for hit-to-lead optimization, resulting in the potential actives with high structural similarity. They also proposed an expanded application of RNN-based generative models to fragment growing. Given a key fragment binding to thrombin, a library of molecules could be generated instantly without extensive similarity searching or external scoring.

Case 4

Subsequently, Merk *et al.* [62] experimentally validated the practicality of Gupta *et al.*'s generative models through synthesizing the agonists of retinoid X receptors and PPAR under the guidance of the fine-tuned model. They synthesized the top five active molecules and tested *in vitro* activity, revealing that four of them were bioactive with the highest EC_{50} value of 60 ± 20 nM. Besides, the above compounds were synthesizable within five steps from commercially available materials. Specifically, they used only 25 fatty-acid analogs with known agonistic activity on retinoid X receptors and/or PPARs as input, performing fragment-based growing from fragment '-COOH' and eventually sampling 1000 SMILES strings. This experimental validation delivered the promising potential of deep generative models for future medicinal chemistry.

RNN-based generative models with RL

By using the above data-driven methods, the RNN-based models attempt to learn the underlying probability distribution over a large chemical set. And the fine-tuning with maximum likelihood estimation (MLE) [63] would cause a risk of forgetting what was initially learned by the RNN. To reduce this risk, the concept of RL has been integrated to molecular generation. RL is usually suitable for solving dynamic decision problems. A policy-based RL generative method aims to fine-tune a pretrained RNN-based generator toward creating desirable molecules through learning an imposed episodic likelihood considering both prior likelihood and a defined reward function.

REINVENT

Olivecrona *et al.* [27] developed an RNN-based generative model for molecular *de novo* design through augmented episodic likelihood-based RL. They proposed a policy-based RL to fine-tune an RNN-based agent for generating molecules with given desirable properties. The architecture consists of two main compartments with the same RNN-based architecture: a Prior network and an Agent network (Figure 2A). The Prior network composed of three GRU layers was trained using MLE on the training dataset, which was able to learn both the syntax of SMILES and distribution of molecular structure. And the learned probability distributions were used to initialize the Agent network. An augmented likelihood was defined as a combination of the Prior network and a user-defined scoring function for rating the generated sequences. Formally, $\log p(A)_{\mathbb{U}} = \log p(A)_{\mathrm{prior}} + \sigma S(A)$. σ is a weight parameter of the scoring function. Through learning this augmented likelihood, the Agent network, with preserving the ability of the Prior network, was forced to update its policy for increasing the expected score of the generated molecules.

After the Prior network was trained on the ChEMBL dataset (including 1.5 million molecules), 94% for Q_{valid} and 90% for Q_{novel} were obtained on the 12,800 generated samples. Coupling with three different scoring functions

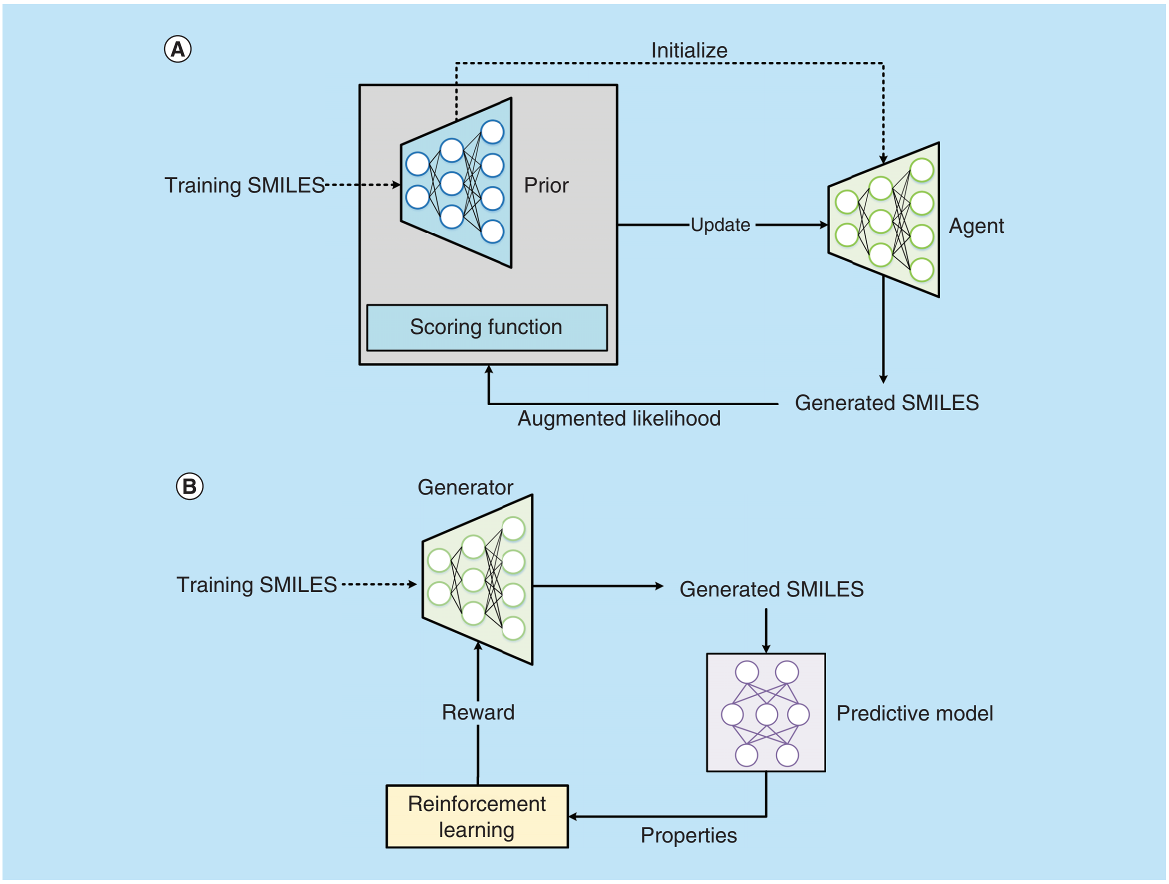

Figure 2. Current architectures of recurrent neural network- and reinforcement learning-based SMILES generative model. (A) REINVENT; **(B)** ReLeaSE.

and the ChEMBL dataset, they proposed three tasks to validate the reliability of the RNN-RL-based models. The results are summarized in Table 1. The first task was to generate sulfur-free molecules and they compared the Agent network with three policy gradient-based methods (Action basis, REINFORCE [64], REINFORCE + Prior) with different cost function definitions as a proof-of-concept. The results demonstrated that the Agent performed much better than the three methods in regard to generating sulfur-free SMILES. And the Agent produced the similar distributions of molecular properties to the Prior. In the second task, the model was trained to generate analogs to the drug celecoxib through similarity guidance. Even when they trained the reduced Prior network with removing all analogs of celecoxib, the Agent could still generate mostly similar compounds. The last example aimed to optimize the Agent toward generating inhibitors against a biological target dopamine receptor type 2 (DRD2) [65]. The DRD2 activity was predicted by an external Support Vector Machine (SVM) classifier with the test AUC of 1.00. The agents derived from the canonical and reduced Prior produced 99 and 99% of valid SMILES and, 97 and 96% of predicted actives, respectively. And the two agents managed to recover 13 and 7% of the test actives, suggesting that the model has learned to generate 'novel' active molecules against DRD2, which were highly similar to the training actives. Compared with the Prior only using MLE, the Agent using augmented likelihood has a significant advantage on generating molecules with desirable properties.

ReLeaSE

Popova *et al.* [44] implemented another RL-based RNN model for generating new chemical compounds with desired properties, which is called ReLeaSE (Reinforcement Learning for Structural Evolution). Different from the REINVENT, ReLeaSE combines the two deep neural networks (generative model *G* and predictive model *P*), shown in Figure 2B. *G* is a stack-augmented RNN (Stack-RNN) architecture to learn hidden rules of forming sequences of letters for generating valid SMILES molecules. During generation, the Stack-RNN takes the valence for all atoms, ring opening and closure, as well as brackets into consideration. P is analogous to a Quantitative Structure-Activity Relationship (QSAR) model for molecular properties prediction with only taking SMILES string as an input vector. It is based on a deep neural network consisting of embedding layer, LSTM layer and two dense layers. The two models were trained separately. G was trained with 1.5 million molecules from the ChEMBL dataset and produced 1 million samples with Q_{valid} of 95.0%, Q_{novel} of 95.3% (against ZINC) and 98.5% (against ChEMBL), and the median SAS of 3.1. Over 99.5% of generated molecules with SAS of below 6 can be regarded as synthetically accessible. In regard to P, the performance on logP and melting temperature (T_{melt}) [66] was comparable with the conventional machine learning models. Some different reward functions (R) that biased properties distribution of molecules including: T_{melt}; logP; inhibitor of JAK2 (R: the exponent of pIC_{50}); substructure bias (R: the exponent of number of benzene rings and total number of small groups substituents). The results are summarized in Table 1, showing that the Stack-RNN generative model with counter cells was suitable for learning the hidden rules of SMILES. All the models had a high Q_{novel} (>94%) against the ZINC and ChEMBL. The additional RL training strategy would lead to an obvious decrease of Q_{valid}, although the generated molecules could meet the optimization objectives. This can be explained via the incompleteness in single-objective optimization but now different properties still difficult to be optimized integrally. Meanwhile, multiobjective *de novo* design of several desirable characteristics is a realistic route of drug discovery but none of the RL strategy could afford currently.

ChemTS

Combined with Monte Carlo tree search (MCTS), Yang *et al.* [46] proposed a new RNN model called ChemTS to implement optimization of penalized logP (see Grammar VAE). For a search tree, each node is related to a SMILES symbol and starting from root node to terminal node corresponding to a complete SMILES string. Through iteration of selection, expansion, simulation and backpropagation, especially the rollout process in simulation, different SMILES strings could be generated by sampling distributions of different symbols. The reward function of molecule m was defined as $r(m) = r(m)/(1 + r(m))$ if m is valid SMILES, otherwise $r(m) = -1$, where $J(m) = \text{penalized\_logP}(m)$. After about 250,000 molecules from the ZINC dataset were used to train the model with this reward function, ChemTS could generate molecules with high penalized logP (the best one of 6.56) in a speed of optimization (40.89 molecules per minute). These results presented that ChemTS has a better optimization efficiency than ChemVAE+BO (see ChemVAE) and Grammar VAE+BO (see Grammar VAE). However, no analysis of generated structures was reported.

RNN & AE-based generative models with BO or CG

The VAE-based models usually contain two parts: encoder: learning to represent molecules in a continuous manner that facilitates the prediction and optimization of their properties; and decoder: learning to map an optimized continuous representation back into a molecular with improved properties. The mean and variance of latent features are approximated and obedient to a distribution $\mathcal{N}(0,\ \boldsymbol{I})$. And BO is often adopted on the latent space for producing molecules with better properties.

ChemVAE

Chemical VAE (called ChemVAE, Figure 3A) [15], the first VAE-based *de novo* molecular generative model, was reported by Gómez-Bombarelli and coworkers. The architecture of ChemVAE consists of an encoder, a decoder and a predictor. The goal of encoder was to convert the discrete representations of molecules (SMILES strings in this case) into real-valued fix-dimensional continuous vectors (latent space). Then the decoder managed to transform the vectors to SMILES strings. The encoder and decoder combined to construct the initial VAE framework and the philosophy of VAE was to minimize the reconstruction error during training. To ensure the validity of decoded SMILES strings, ChemVAE adds Gaussian noise to the encoder with penalty term guaranteeing the valid decoding, forcing the decoder to learn how to decode points in latent space. Instead of limiting the VAE to generate valid SMILES strings, they adopted the RDKit to filter the invalid ones and that is why the Q_{valid} of SMILES generated ranging only from 1 to 70%. To achieve *de novo* design, a predictive model based on multilayer perceptron, was joined into VAE to predict the molecular properties from latent space. The datasets used for training ChemVAE were from the QM9 [67] (with 108,000 molecules fewer than nine heavy atoms) and ZINC database (with 250,000 drug-like compounds), respectively.

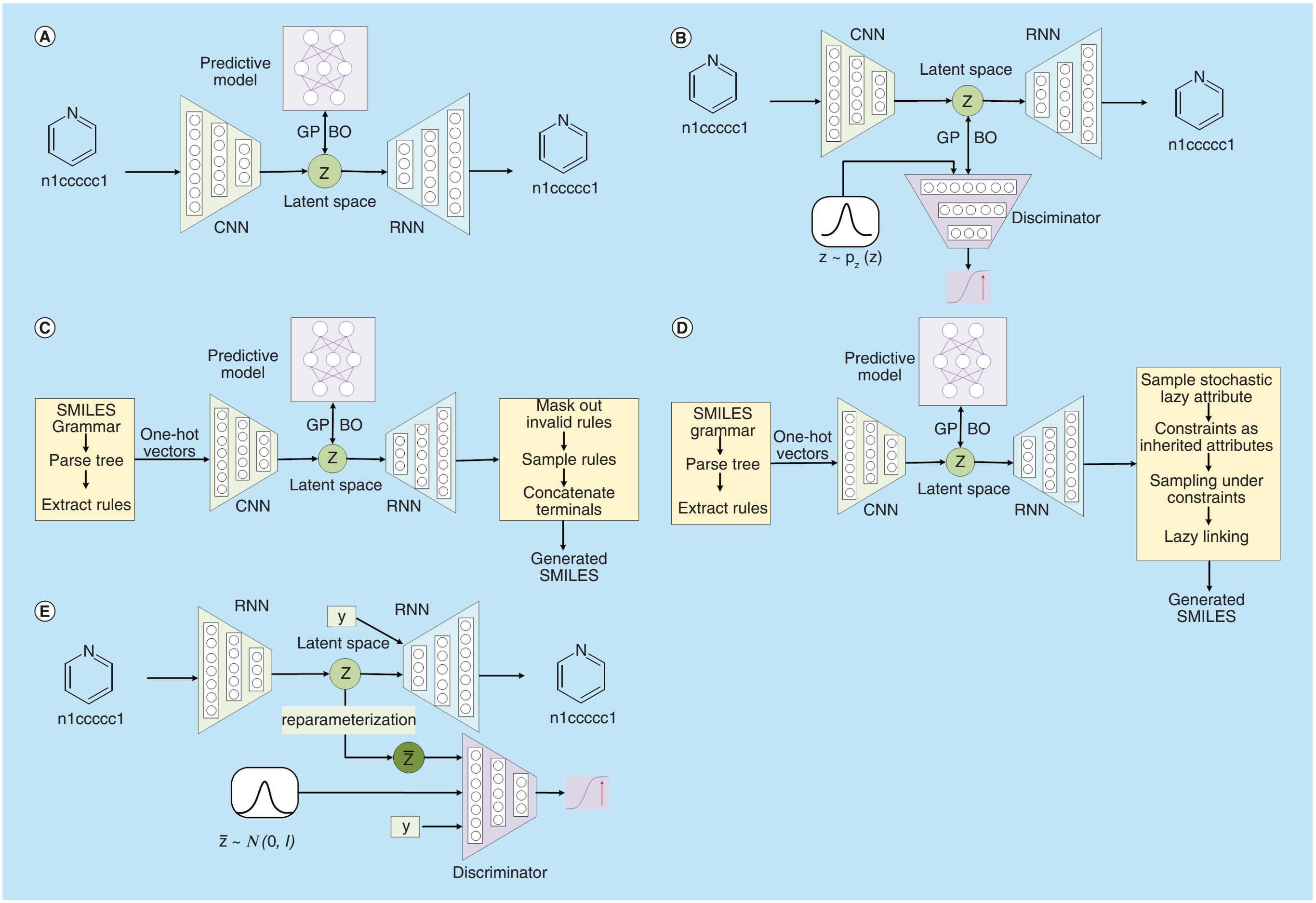

Figure 3. Current architectures of recurrent neural network- and variational autoencoder-based SMILES generative model. (A) ChemVAE; **(B)** adversarial autoencoder; **(C)** Grammar VAE; **(D)** syntax-directed variational autoencoder; **(E)** entangled conditional adversarial autoencoder.

Also, it is worth mentioning that the encoding and decoding processes were both probabilistic. The sampling probability of the latent points was in inverse proportion to the Euclidean distance to the original points as well as the decoding rate, indicating the potential correlation of molecular structures in latent space. Statistically, 30 latent molecules could be found in the vicinity of one native latent point. Moreover, spherical interpolation of molecules was also available in latent space. In contrast to the genetic algorithm, ChemVAE generated more similar molecules to the input dataset. To realize drug discovery, maximizing desired molecular properties and screening the high-quality lead compounds were considered more crucial than just generating similar molecules. Therefore, the predictive model was jointly trained with VAE to predict and even bias the properties (logP, SAS and quantitative estimate of drug-likeness [QED]) distribution of generated structures. Compared with other prediction-only models, ChemVAE displayed similar prediction error. Moreover, to demonstrate the optimization performance, they trained the GP model using 500/1000/2000 diverse molecules as its optimization efficiency was correlated to the amount of training set. The objective function was defined as 5 × QED - SAS, a rough metric of finding the most drug-like molecule with simple synthesis route. Starting from a starting point, the GP could end up reaching a high scoring region, performing much better than using random Gaussian search or the genetic algorithm. As mentioned by the authors, future work may focus on valid SMILES string generation or better objective function to capture core desirable traits.

Grammar VAE

Proposed by Kusner *et al.*, Grammar VAE [49] utilized a context-free grammar (CFG) to form a parse tree, which is decomposed into a sequence of production rules. Shown in Figure 3C, these rules are fed into an encoder with convolutional neural network architecture, then an RNN architecture as a decoder for generating syntactically valid SMILES. A CFG was defined as 4-tuple $G = (V, T, R, S)$, containing a finite set of nonterminal symbols V, a finite set of terminal symbols T, a finite set of production rules R and a distinct start symbol S. The SMILES strings can be generated via adopting production rules (sampling from start symbol till no nonterminals left) recursively. While ChemVAE fails to produce many valid sequences, the goal of Grammar VAE is to teach VAE, the explicit grammars of SMILES. The principle of Grammar VAE could be analyzed as encoding: parsing SMILES string into a parse tree and decomposed the tree into a series of production rules; translating the production rules to one-hot vectors and every dimension of the vectors maps to a production rule; mapping the collection of one-hot vectors as a matrix X to a latent vector $\boldsymbol{z}$; and decoding: parsing a continuous vector $\boldsymbol{z}$ to produce unnormalized log probability vectors F and every dimension of F corresponds to a production rule like above one-hot encoding; writing the collection of F and rows of F could be used to select a sequence of valid production rules.

After training the above encoder and decoder using the ZINC database, the results indicated that Grammar VAE provided smoother interpolation and more valid molecules than ChemVAE. The latent molecular representations were used to construct a sparse GP with a better predictive performance on low log-likelihood (LL) and root mean square error (RMSE) of -1.739 and 1.404 than ChemVAE (-1.812 and 1.504). A combinational objective of penalized logP for a given molecule m is defined by:

$$\text{penalized\_logP}\,(m) = \text{logP}\,(m) - \text{SAS}\,(m) - \text{ring\_penalty}\,(m)$$

Where logP(m) is a key term in charactering the druglikeness of m, SAS(m) is a measurement of synthetic accessibility and ring_penalty(m) is used to avoid generating unrealistically large carbon rings. On this basis, BO was applied to produce molecules with improved properties. The top three molecules from Grammar VAE had higher scores than that from ChemVAE. The Rs and Q_{valid} were 53.7 and 7.2%, respectively, better than those of ChemVAE (44.6 and 0.70%). However, Grammar VAE did not tackle the problem of the semantics of SMILES.

Syntax-directed VAE

To take both the syntax and semantics of SMILES into account, syntax-directed VAE (SD-VAE) was proposed by Dai *et al.* [51] depicted in Figure 3D. In addition to utilizing the CFG-based decoder to capture the syntax of SMILES, they addressed the semantics, including ring bonds and valence of atoms. This approach enforces the constraints on the decoder via adding the offline syntax-directed translation check into on-the-fly generation process, thus ensures the sampling of not only syntactically but also semantically SMILES sequences. As the CFG defined similar to Grammar VAE, SD-VAE enriches the CFG with attributes and rules. Thus, the encoder is achieved by introducing the two attributes (inherited and synthesized attributes) to the nonterminal symbols. The tree generation process of the decoder with stochastic lazy attribute can be decomposed as stochastic generation of attributes in the absence of synthesized attributes; constrained sampling with inherited attributes to generate subtree; calculation of synthesized attribute according to generated subtree; lazy linking of atoms.

For a fair comparison, the same training dataset from ZINC was used to train the SD-VAE. Compared with the Rs and Q_{valid} of SD-VAE (76.2 and 43.5%), the performance was much better than the previous models. Like before, the penalized logP was optimized using BO and it was found that SD-VAE could search richer molecules with higher scores than ChemVAE and Grammar VAE. Meanwhile, they also measured the D_{in} metric of generated molecules and demonstrated that SD-VAE could still produce more diverse yet valid molecules in a more compact decoding space. The projected 2D space was visualized and it was found that the latent space of SD-VAE was characterized by having smooth differences between neighboring molecules, and more complicated decoded structures. In summary, SD-VAE provided a more consistent and practical VAE-based model than previous ones while maintaining an almost equivalent computational cost.

Adversarial autoencoder

As illustrated in Figure 3B, the AAE architecture was constructed based on a standard AE, mainly via adding an adversarial network to induce the latent vector code z to match a certain target distribution $p_z(z)$. Fingerprint-based AAE models were first constructed by Kadurin *et al.* [68,69], and they mainly focused on generating molecular fingerprints rather than real molecules, which would not be reviewed here. A SMILES-based AAE molecular generative model, proposed by Blaschke *et al.* [53], was built using 1.3 million SMILES strings extracted from the ChEMBL database. Four different AE architectures called NoTeacher VAE, Teacher VAE, Gaussian AAE and

Uniform AAE were explored for *de novo* molecular generation. As a result, both Uniform AAE (78.3%) and Gaussian AAE (77.4%) performed slightly better in Q_{valid} than Teacher VAE (77.6%) while NoTeacher VAE (19.3%) had the worst Q_{valid} in the validation set. Moreover, the successful reconstruction of celecoxib and close analogs demonstrated the smoothest latent space of Uniform AAE among the four models.

Furthermore, an external TPM based on SVM, estimating the probability of active molecules against DRD2 target, was used as an objective. The SVM model was constructed similar to the above REINVENT. During optimization, the fraction of generated active compounds was proportional to the predicted BO score in the specific latent regions, where the score values were predicted by previous SVM model. In short, the AAE model was able to generate structures with higher Q_{valid} than ChemVAE. And it could perform BO to search for potential drug candidates toward a specific target through the external SVM predictive model on the known labeled dataset, which cannot be avoided.

Entangled conditional AAE

Recently, Polykovskiy *et al.* [54] proposed a conditional AAE model called entangled conditional AAE (ECAAE) shown in Figure 3E. Since supervised AAE [20] cannot deal with disentanglement issues and leads to inconsistency conditional generation, ECAAE integrates predictive and joint disentanglement approaches into supervised AAE to overcome this issue. Combined with the reparameterization trick, ECAAE tries to improve the interpretability of latent space. The 1.8 million drug-like molecules from ZINC were used to train ECAAE for implementing conditional molecular generation. First, using 166-bit Molecular Access System (MACCS) fingerprints of known potent molecules as conditions, they tested the performance of generating structural analogs. By comparing Tanimoto similarity and Hamming distance, they found ECAAE performed best with 93.6 and 3.28%, respectively. The diversity of generated molecules has a certain improvement. Second, the authors evaluated continuous properties like logP and SAS, and the results suggested that ECAAE could generate molecules with moderately correlated properties (Pearson correlation coefficient ρ: 0.613 and 0.413). Subsequently, after computing the binding energy E of 14,000 molecules with AutoDock Vina [70], the conditions of logP, SAS and E were used to train semisupervised ECAAE model and achieved an improved ρ with logP of 0.804, SAS of 0.593 and E of 0.406. Conditioned on given E (-11.1), logP and SAS, ECAAE also produced two of generated molecules with E of -11.7. Finally, based on 300,000 structures generated by ECAAE, they discovered a selective inhibitor against JAK3 (IC_{50} = 6.73 uM) over JAK2, B-raf and c-Raf through a series of filters, simulations and expert experience.

RNN & GAN-based generative models with RL

Objective reinforced GAN & objective reinforced GAN for inverse-design chemistry

Building upon SeqGAN [71], an objective reinforced GAN (ORGAN) was raised by Guimaraes *et al.* [55], in which the GAN architecture is combined with reward functions with RL to generate SMILES strings. From Figure 4A, G_θ is a LSTM-based generator parameterized by θ, that produces high-quality sequences $\boldsymbol{X}_{1:T} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$. And a discriminator D_ϕ parameterized by ϕ is a convolutional neural network specifically for sequence classification tasks. G_θ is trained to fool D_ϕ and D_ϕ to classify real and generated SMILES sequences. G_θ is trained as an agent through the REINFORCE algorithm and the reward function R was supplied by D_ϕ and the objectives O_i, which is defined as $R(\boldsymbol{X}_{1:T}) = \lambda D_\phi(\boldsymbol{X}_{1:T}) + (1-\lambda)O_i(\boldsymbol{X}_{1:T})$, λ is a trade-off parameter (default 0.5). ORGAN was trained on 5000 molecules from the QM9 dataset. The objectives of normalized logP, SAS and QED were optimized for generating biased molecules. The results are shown in Table 1. ORGAN often showed a high Q_{valid} of 88.2–96.5% and a D_{in} of 0.55–0.92. Guided by the three single objectives, the generated molecules would produce a distribution with better properties than the RNN-based baseline model trained by MLE.

Later, with the same architecture of the ORGAN, ORGANIC (ORGAN for Inverse-design Chemistry) [57] was proposed and capable to optimize a distribution over molecular space according to certain desired properties. The model covered three different applications on melting points, drug discovery and organic photovoltaics. In the case of drug discovery, they tried to optimize the QED and Linpinski's rule of five (RO5) [72], and found that ORGANIC could only improve the distribution of QED from the range of 0.1–0.4 to 0.8–0.9. Regarding the categorical data in RO5, ORGANIC was helpless for improving the distribution. A major limitation of the ORGAN paradigm is the greatly vary rate of nonvalid molecules (0.2–99.9%). For valid molecules, many repetitive patterns were found. Such performance was greatly dependent on the training set and optimized metrics. The authors also ascribed this limitation to the rough landscape of the chemical space where a small change could lead to striking effects.

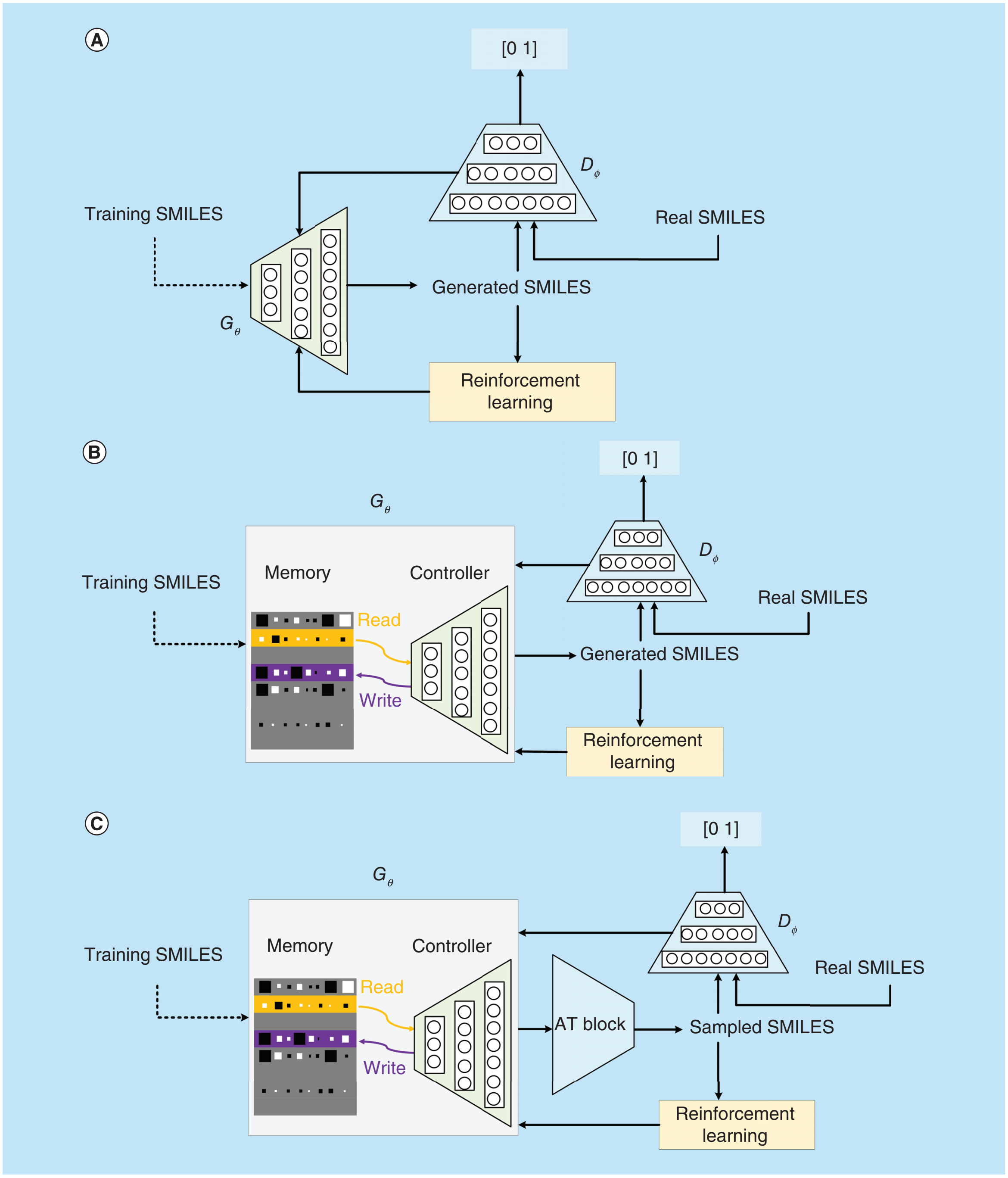

Figure 4. Current architectures of recurrent neural network-, reinforcement learning- and adversarial learning-based SMILES generative model. (A) ORGAN/ORGANIC; **(B)** RANC; **(C)** ATNC.

Reinforced adversarial neural computer

Based on the ORGANIC paradigm, Putin *et al.* constructed an architecture named Reinforced Adversarial Neural Computer (RANC) [60], shown in Figure 4B. RANC consists of a generator G_θ (differentiable neural computer [DNC]) [73], a discriminator D_ϕ and objective functions O_i. Specifically, DNC is an LSTM controller with external memory (like Stack-RNN) and its advantages lay in its powerful memory to reconstruct and generate complex and much longer SMILES strings than LSTM. The definition of the reward function $R(\boldsymbol{X}_{1:T})$ is the same as the ORGANIC. G_θ and D_ϕ were pretrained for a given number of training epochs, and the objective of G_θ is to maximize O_i and fool the D_ϕ. Meanwhile, the action value function of candidate states (partial sequences) was calculated by N-times Monte Carlo search.

First, they utilized two datasets ZINC (15,000 molecules) and ChemDiv (15,000 molecules) [110] rather than only ChemDiv to make comparisons. Instead of using four different objective reward functions, they chose only the QED and RO5 as the reward functions and evaluated the RANC's and ORGANIC's performance on the generated SMILES strings of 896,000 and 320,000. For the two datasets, RANC (48 and 76%) produced a higher Q_{unique} than ORGANIC (18 and 24%) on the longer SMILES strings (average length of 46 and 40 vs 23 and 23). But the lower Q_{valid} (58 and 76% vs 87 and 92%) was revealed. It was suggested that DNC could afford the generation on long SMILES strings. Then in the chemical analysis section, RANC sampled less molecules (20,000) in two datasets. Additionally, the authors investigated the similarity distribution and QED of generated molecules and found that RANC generated more diverse molecules and displayed higher QED scores than ORGANIC. Ten representative generated structures of both models were also shown and demonstrated that RANC could sample more complex and attractive lead drug compounds. It can be concluded from the above comparison that RANC was more suitable for generating more unique, diverse and complex structures than ORGANIC. However, it suffered from similar problems like ORGANIC, including less efficient in controlling valid SMILES strings.

Adversarial threshold neural computer

Similarly, Putin *et al.* [59] also proposed adversarial threshold neural computer (ATNC, Figure 4C), which is an extension of RANC with a specific adversarial threshold block (AT, a copy of D_ϕ but responds behind the original D_ϕ). AT block is a model-based RL approach to simulate environment reactions and select the most favorable actions for the trained agent. For the model-free approach of ORGANIC and RANC, most of model actions receive negative rewards from the environment, which will drastically reduce the quality of the generated molecules. During training, the pretrained weights of D_ϕ were copied to AT and then the RL training began. G_θ generated K samples at first, then AT would select the most likely molecules to training data until the number of chosen samples reached the demand of D_ϕ.

The 15,000 drug-like molecules from ChemDiv were used to train the ANTC. To make a fair comparison, ATNC applied almost the same hyperparameters as ORGANIC and evaluated two models via predefined four different objective reward functions: internal similarity (for generating similar molecular structures, IS); internal diversity clustering (for generating diverse molecular structures, IDC); Muegge drug-likeness filter (for generating drug-like molecular structures, MU); presence or absence of sp^3-rich fragments (for generating at least one sp^3-rich molecular structures, SP3). In general, the ATNC model produced the Q_{valid} of 71–74% and the Q_{unique} of 73–86%, and showed better stability than ORGANIC model (a Q_{valid} of 7–83%, and a Q_{unique} of 22–91%). Furthermore, by analyzing the generated molecules, ATNC presented better drug-likeness properties through evaluating four basic molecular descriptors (number of atoms, molar weight, logP, topological polar surface area) and five chemical statistical features (Table 1). Finally, they performed *in vitro* validation of molecules to evaluate the capacity of ATNC model. After selecting and purchasing top 50 predicted novel compounds without reported activity against kinase family enzymes, seven molecules were tested with inhibition potency. Further analysis using nearest structural analogs with known activity was also performed by searching databases like PubChem and ChEMBL, demonstrating the efficacy of ATNC in finding novel scaffolds.

Graph-based generative models

Graphs are usually regarded as a fundamental structural data, which are widely adopted in the field of biological and chemical network systems. Given an example of chemical molecule, a 2D structure is a graph with nodes as its atoms and edges between two nodes as its bonds. There has been a surge of great interest in studying DL models on graphs, with useful applications across fields such as chemistry [74–76] or computer vision [77]. Fueled by the success of RNNs, VAEs, GANs in image, audio and text generation, DL generative models on molecular graphs have recently allowed the direct generation of valid molecular graphs.

Recurrent graph-based generative models with CG

The recurrent graph-based generative models for molecular graph usually adopt a sequential process, which generates one atom at a time and connects each node to the existing partial graph by creating edges one by one. Such model defines a distribution over the sequence of graph generating decisions by defining a probability distribution over possible outcomes for each step. Each of the decision steps is implemented by deep neural network. Recurrent graph-based generative model is graph structured with structured memory, while the LSTM (or GRU) stores information in flat vectors.

GraphNet

The architecture was designed by Li *et al.* [28], depicted in Figure 5A. As this model makes use of the structure of molecular graph to create representations of atoms and bonds via an information propagation process (GRU architecture), these representations are used to make sequential graph building decisions. Each of the decision steps is modeled by three modules: adding a new node or not (with probabilities provided by a f_{addnode} module); adding a new edge or not (probabilities provided by f_{addedge} module); and picking one node to connect to the new node with typed edges (probabilities provided by f_{nodes} module). These probabilistic decision-making modules are parameterized by training with known molecular graphs. In the process of training, fixed ordering and uniform random ordering of node permutation were taken into consideration.

The ChEMBL database was extracted with the limit of at most 20 heavy atoms. The numbers of training, validation and test set are 130,830, 26,166 and 104,664, respectively. These sets were used to train and test GraphNet models. The results are summarized in Table 2. The negative log-likelihood losses on the test set with fixed ordering and random ordering are 20.55 and 58.36, respectively. The generative ability of both models was tested with 100,000 generated samples. The models trained with fixed ordering and random ordering acquired a Q_{valid} of 97.52 and 95.98%, and a Q_{novel} of 90.01 and 95.54%, respectively, which is significantly better than the SMILES-based generative models (a Q_{valid} of 93.59% and a Q_{novel} of 83.95%). The graph model was also trained on the ZINC dataset, where a few benchmark results are available for comparison. The model trained with fixed ordering achieved a Q_{valid} of 89.2%, and a Q_{novel} of 89.1%, and 74.3 and 74.3% with random ordering. The excellent performance on the ZINC dataset was superior to that of ChemVAE (0.01%, 0.01%), Grammar VAE (34.9%, 2.9%) and GraphVAE (13.5%, NaN).

Conditional graph generation task had been explored with three different conditions including the number of atoms, edges and aromatic rings in a molecule. Given the ChEMBL training subset only containing molecules of 0, 1, 3 aromatics rings, the graph model has the interpolation ability to produce the 2-rings molecules (a Q_{valid} of 91.5%, and a Q_{novel} of 91.3%) under the condition of two aromatic rings. In addition, the model performed an extrapolation ability to produce the 4-rings molecules (a Q_{valid} of 84.8%, and a Q_{novel} of 84.0%), which suggested that the conditional sequential graph model showed potential promise of inferential ability.

MolMP & MolRNN

Similar to GraphNet architecture from Li *et al.* [28], the MolMP generator, implemented by Li *et al.* [29], builds a molecular graph by iteratively refining its intermediate structure with three transition actions (f_{append}, f_{connect} and $f_{\text{terminate}}$). The process starts from the empty graph $\mathcal{G}_0$. At step i, a graph transition is selected from the set of three actions based on $\mathcal{G}_i$, which is performed on $\mathcal{G}_i$ to get the graph structure for the next step $\mathcal{G}_{i+1}$. At the final step, termination operation is performed and the model outputs the final graph $\mathcal{G}^*$. First f_{append}: this action adds a new atom to $\mathcal{G}_i$ and connects it to an existing atom with a new bond. Second, f_{connect}: this action connects the existing atoms V_i to the new atom $v^*$with typed bonds. Third, termination: this action terminates the generation process. The entire process is shown in Figure 5B. Compared with GraphNet, the actions of f_{addnode}, f_{addedge} and f_{nodes} are merged into a single f_{append} step, and f_{connect} is used to avoid the repeated operation of adding edges. It helps to reduce the number of steps during generation. While GraphNet uses GRU to obtain atomic representations, which are integrated to molecular representation by Gated Sum, MolMP uses Graph Convolutional Network (GCN) and average pooling for atomic and molecular representations. When it comes to MolRNN, a graph-level recurrent unit is added to the MolMP architecture, which is implemented by three GRU layers.

The ChEMBL dataset (containing 1,488,640 molecules with up to 50 heavy atoms) was used to construct the MolMP and MolRNN models. From Table 1, the MolRNN model achieved better performance with the negative log-likelihood loss of 24.08 than the MolMP with the loss of 26.25. The generated 300,000 samples were evaluated for each model, leading to a best Q_{valid} and Q_{novel} of 96.3 and 98.8% for the MolMP, and 97.0 and 98.5% for the MolRNN. And the graph-based models have the advantages on the Q_{valid} and Q_{novel} over SMILES-based models with the corresponding ranges of 80.4–93.5% and 96.5–98.6%.

The analysis of common mistakes made by the two kinds of models was performed and compared. The most common one of invalid outputs for the SMILES-based models is caused by grammar mistakes (such as unclosed

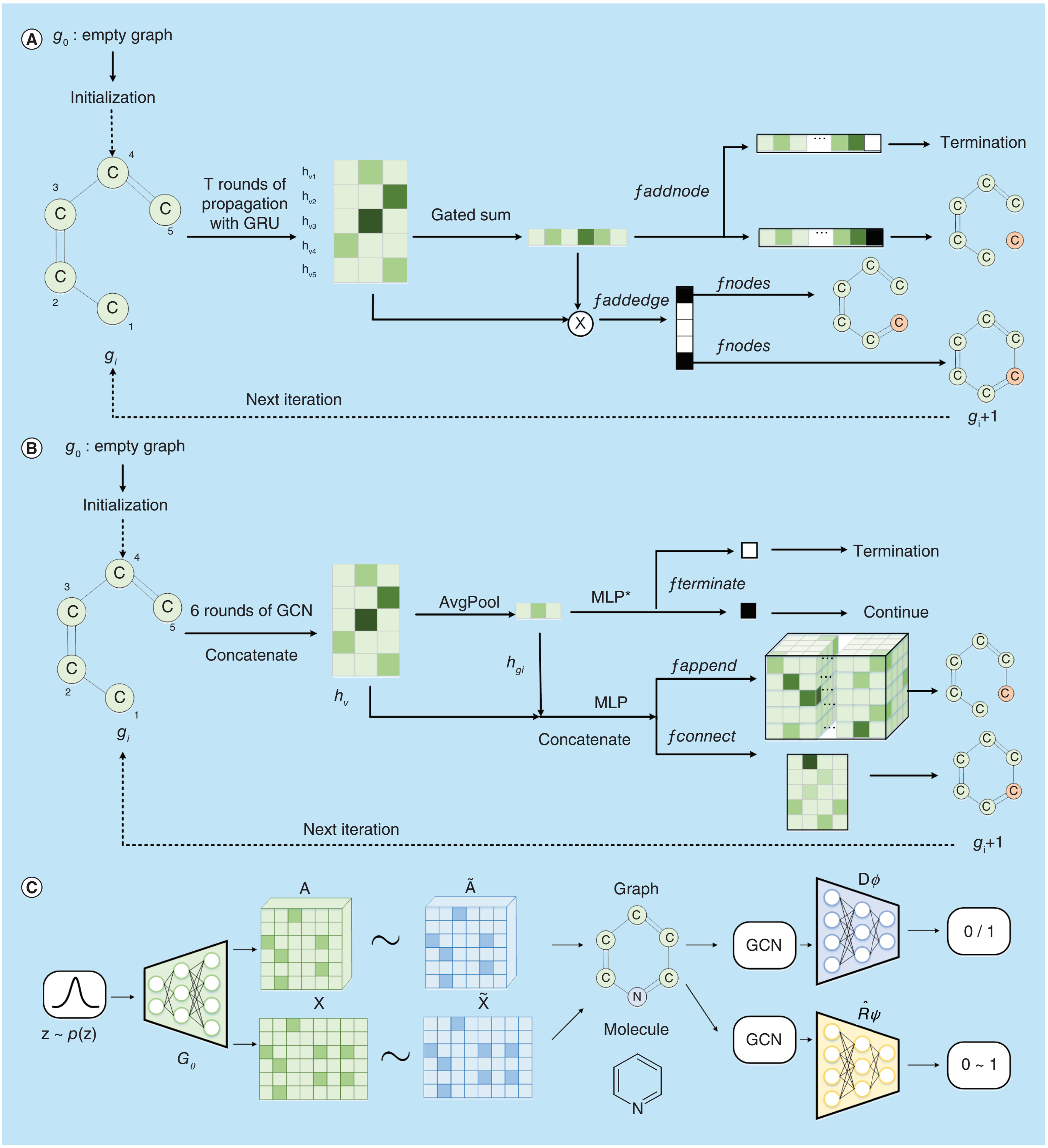

Figure 5. Current architectures of recurrent-based and generative adversarial network-based graph generative model. (A) GraphNet; **(B)** MolMP/MolRNN; **(C)** MolGAN.

Table 2. A summary of graph-based generative models.

Model type	Model name	Dataset	Molecular size	Number of molecules	Number of generated molecules	Generative performance	Optimization task	Optimization performance	Ref.
VAE-based models	GraphVAE [78]	QM9	<10 heavy atoms	~114,000 (training), 10,000 (test), 10,000 (validation)	10,000	55.7% (V), 76.0% (U), 61.6% (N)	Given a histogram of heavy atoms as 4D label	56.5% (V), 66.6% (U), 66.1% (N)	No report
		ZINC	<21/31/39 heavy atoms	~52,500/~230,000/ 250,000	No report	34.1% (V)/18.5% (V)/13.5% (V)			
RNN- and VAE-based models	NeVAE [79]	ZINC	<39 heavy atoms	~10,000	5000	No masking: 58.0% (V) 100% (N); Masking: 100% (V), 100% (N)	Penalized logP	LL: -1.631; RMSE: 1.231; 100% (V). The average value: 0.36. The best value: 2.32	[80]
		QM9	<10 heavy atoms	~10,000	No report	No masking: 67.0% (V) 100% (N); Masking: 100% (V), 100% (N)			
	CGVAE [81]	QM9/ZINC	<10/<39 heavy atoms	~134,000/250,000	20,000	100% (V), 94.4% (N), 98.6% (U)/100% (V), 100% (N), 99.8% (U)	nQED	The best value: 0.9383	No report
	JT-VAE [82]	ZINC	<39 heavy atoms	220,011 (training), 24,445 (validation), 5000 (testing)	Rs: 100 molecules (100-times) V:1000 samples (100-times)	76.7% (train Rs), 100% (V)	Penalized logP	LL: -1.658; RMSE: 1.290. The best value: 5.30	[83]
RNN-based models	GraphNet [28]	ChEMBL	<21 heavy atoms	130,830 (training), 26, 166 (test), 104, 664 (validation)	100,000	Fixed ordering: 97.5% (V), 90.0% (N); random ordering: 96.0% (V), 95.5% (N)	Given 0, 1, 3-ring molecules, generate 2-ring and 4-ring	2-ring: 91.5% (V), 91.3% (N); 4-ring: 84.8% (V), 84.0% (N)	No report
		ZINC	<39 heavy atoms	No report	100,000	Fixed ordering: 89.2% (V), 89.1% (N); random ordering: 74.3% (V), 74.3% (N)			
	MolMP [29]	ChEMBL	<51 heavy atoms	1,488,640	300,000	96.3% (V), 98.8% (N)			[84]

Rs means the reconstruction rate against the test set. nQED, nlogP, nSAS are the normalization of QED, logP and SAS.
D_{in}: Internal diversity; EOR: Enrichment over random; MW: Molar weight; GAN: Generative adversarial network; LL: Log-likelihood; QED: Quantitative estimate of drug-likeness; RMSE: Root mean square error; RNN: Recurrent neural network; SAS: Synthetic accessibility scoring; VAE: Variational autoencoder.

Table 2. A summary of graph-based generative models (cont.).

Model type	Model name	Dataset	Molecular size	Number of molecules	Number of generated molecules	Generative performance	Optimization task	Optimization performance	Ref.
	MolRNN [29]	ChEMBL	<51 heavy atoms	1,488,640	300,000	97.0% (V), 98.5% (N)	1) Scaffold; 2) SAS and QED; 3) JNK3 and GSK-3β	1) 90.0–98.2% (V), 0.45–0.815 (D_{in}), EOR >150; 2) 92.9–99.7% (V), 0.798–0.864 (D_{in}), EOR >30; 3) 93.2–95.5% (V), 0.783–0.854 (D_{in}), EOR >40	
	GCPN [85]	ZINC	<39 heavy atoms	250,000	100,000	100% (V)	1) Penalized logP; 2) QED; 3) MW, logP targeting	1) The best value: 7.98. 2) The best value: 0.948. 3) High success rate in generating diverse molecules within the target range	[86]
GAN-based models	MolGAN [87]	QM9	9 heavy atoms	133,885	6400	98.1% (V), 10.4% (U), 94.2% (N)	1) nQED; 2) nSAS; 3) nlogP; 4) all the above three	1) 100% (V), 2.2% (U), 0.97 (D_{in}), 0.62 (nQED); 2) 100% (V), 2.1% (U), 0.95 (D_{in}), 0.95 (nSAS); 3) 99.8% (V), 2.0% (U), 0.99 (D_{in}), 0.89 (nlogP); 4) 98.0% (V), 2.3% (U), 0.93 (D_{in}), 0.51 (nQED), 0.82 (nSAS), 0.69 (nlogP)	[88]

Rs means the reconstruction rate against the test set. nQED, nlogP, nSAS are the normalization of QED, logP and SAS.
D_{in}: Internal diversity; EOR: Enrichment over random; MW: Molar weight; GAN: Generative adversarial network; LL: Log-likelihood; QED: Quantitative estimate of drug-likeness; RMSE: Root mean square error; RNN: Recurrent neural network; SAS: Synthetic accessibility scoring; VAE: Variational autoencoder.

parentheses or unpaired ring numberings), while the majority cause of invalid outputs is broken aromaticity. It can be likely corrected by literately refining the number explicit hydrogens of aromatic atoms.

Conditional graph generation was tested with three preferable objectives in drug design. First, scaffold-based generation. Second, SAS and QED-guided generation; third, dual inhibitor against JNK3 [89] and GSK-3β [90]. The given requirements are converted into the conditional code c, which is concatenated to $\mathcal{G}_i$ then influenced the corresponding transition. Compared with fine-tuning-based methods, such conditional generative models can be easily applied to multiobjective and multitask settings. Among the three conditional tasks, the graph-based models usually have a higher Q_{valid} than the SMILES-based models, which is similar to the unconditional tasks. In the view of generated molecules by matching the corresponding conditions, a high enrichment rate over random level shows that CG is an effective and practical approach for multiple objectives. Besides CG, RL and AL also can been adopted to a similar graph-based generative architecture for property optimization and targeting [85].

VAE-based generative models with BO

GraphVAE

GraphVAE [78] is devised to convert a molecular graph into a vector in a continuous code space (encoder) then translate this vector into a graph (decoder), shown in Figure 6A. A molecular graph is characterized by $\mathcal{G} = (\mathcal{A}, \mathcal{E}, \mathcal{F})$ with its adjacency matrix $\mathcal{A}$, edge attribute tensor $\mathcal{E}$ and node attribute tensor $\mathcal{F}$. A probabilistic architecture of VAE was used to jointly train an encoder $q_\phi(\boldsymbol{z}|G)$ and a decoder $p_\theta(G|\boldsymbol{z})$ to map between the space of graphs G and the continuous embedding $\boldsymbol{z} \in \mathbb{R}^D$, where ϕ and θ are learned parameters. A regularization term, KL-divergence, is added into the latent code space with a prior isotropic Gaussian distribution $p(\boldsymbol{z}) = N(0, I)$, which is aimed to approximate the two distributions of $q_\phi(\boldsymbol{z}|G)$ and $p(\boldsymbol{z})$. In the implementation of differential computing, graph encoder used edge-conditioned graph convolutions [91] to embed a graph into $2D$ features as mean and variance, respectively. For graph decoder, the authors skillfully transform a nondifferential problem of discrete graph generation into a differential one of probabilistic graph decoder with the predefined limit of maximum K nodes. An approximate graph matching algorithm of max-pooling matching by [92] was adopted to calculate the reconstruction loss of molecular graphs.

The QM9 dataset contains about 134k organic molecules of up to nine heavy atoms with four distinct atomic numbers and four bond types. The dataset was split with 10,000 molecules for validation, 10,000 molecules for testing and the remaining for training. According to the decoder of GraphVAE, 10,000 samples were drawn from the latent space. The Q_{valid}, Q_{unique} and Q_{novel} from the best model are 55.7, 76.0 and 61.6%, respectively. Only when considering the canonical ordering, the Q_{valid} was improved to 81.0%, but the Q_{unique} and Q_{novel} had a certain reduction. Conditional (given a histogram of heavy atoms as 4D label y) generative models were tested on GraphVAE. The best index of Q_{valid}, accuracy, Q_{unique} and Q_{novel} respectively is 56.5, 46.7, 66.6 and 66.1%. The ZINC dataset containing about 250,000 molecules of up to 38 heavy atoms was tested on GraphVAE, leading to a low Q_{valid} of 13.5%. It suggested that GraphVAE would be not well suitable for large and diverse molecules.

VAE & RNN-based generative models with BO

RNN-based models usually only emphasis on step-wise molecular generation, while VAE-based models focus on both molecular latent representation (with an encoder) and molecular generation (with a decoder). Generally, a VAE-based generative model with predictive molecular representations is usually preferred but may have a low performance on molecular generation. An RNN-based model allows a dynamic generation to create variable-sized molecular graphs. However, creating a molecule atom by atom would generate chemically invalid intermediaries, as previously mentioned about broken aromaticity. To overcome these limitations, Jin *et al.* [82] and Samanta *et al.* [79] utilized a combination of the VAE- and RNN-based models to implement effective molecular generation.

NeVAE

It is a step-wise generative model for undirected molecular graphs, is based on VAEs [79]. For each molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of atom features $\mathcal{F} = \{f_u\}_{u\in\mathcal{V}}$ and edge weights $\mathcal{Y} = \{y_{\text{uv}}\}_{(u,v)\in\mathcal{E}}$, the encoder of NeVAE is an inference model $q_\phi(\boldsymbol{z}_u|\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{Y})$ that defines a probabilistic encoding for each atom by extracting atomic information from K different layers ($\boldsymbol{c}_u(1, 2, ..., K)$, $u \in \mathcal{V}$), shown in Figure 6B. This information is fed into a neural network ($\phi^{\text{enc}}(\boldsymbol{c}_u(1, 2, ..., K))$) to make the product $\boldsymbol{z}_u$ obey the distribution of $\mathcal{N}(0, \boldsymbol{I})$ for each atom u.

The atom-based embedding strategy is invariant to permutations of the atoms and do not depend on the number of atoms and bonds. Thus the encoder allows for variable-sized molecular graphs. For the decoder, for each atom u

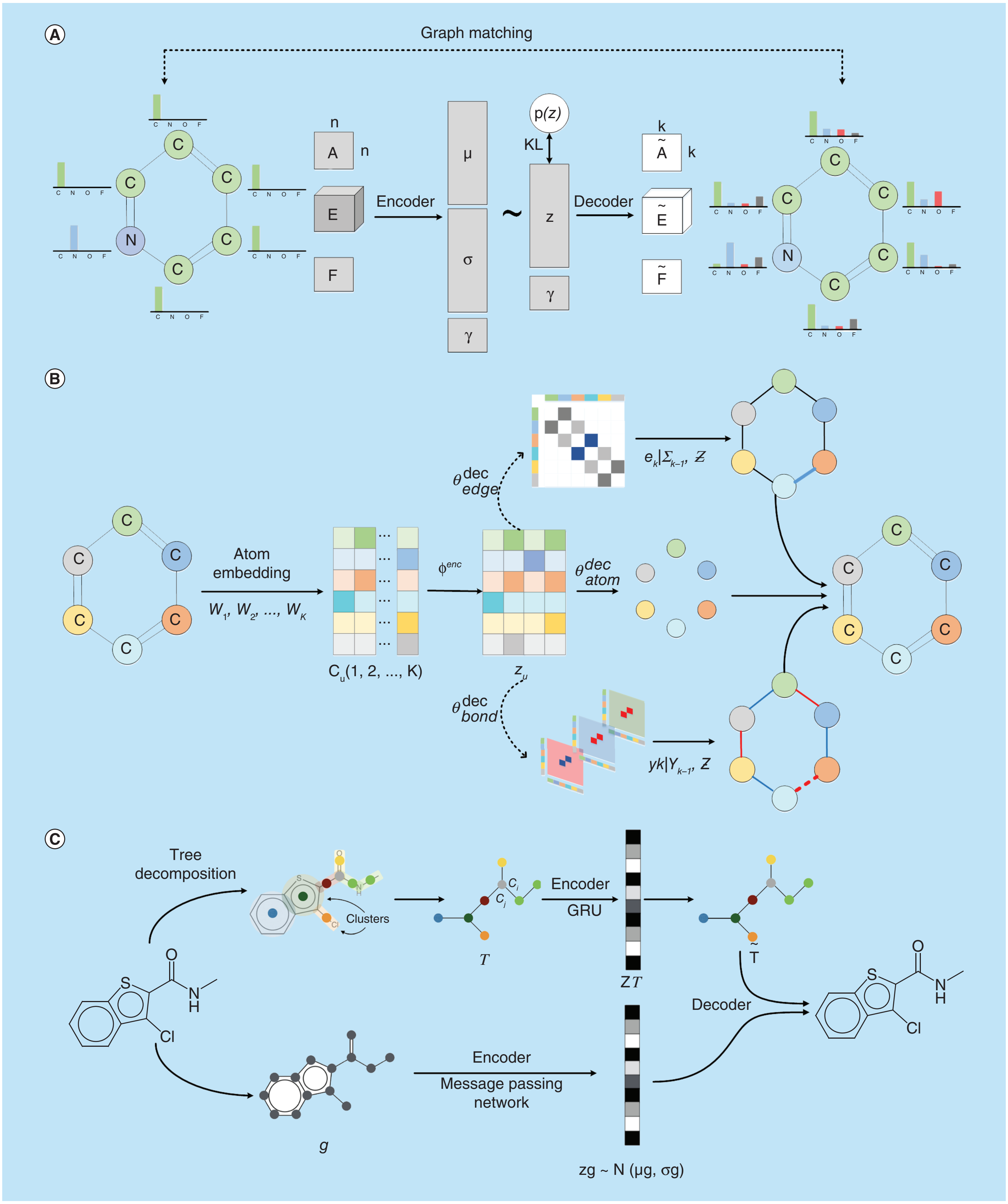

Figure 6. Current architectures of variational autoencoder-based graph generative model. (A) GraphVAE; **(B)** NeVAE; **(C)** JT-VAE.

with latent variable $\boldsymbol{z}_u$, it is fed into a nonlinearity $\theta^{\mathrm{dec}}_{\mathrm{atom}}$ with softmax then produces the atomic probabilistic feature $\widehat{f}_u$. For all potential edges (u, v), each pairwise latent representation $(\boldsymbol{z}_u, \boldsymbol{z}_v)$ is fed into a nonlinear operation $\theta^{\mathrm{dec}}_{\mathrm{edge}}$. Then, the edges are sampled one by one from a softmax distribution of the previous outputs. To determine edge weights (bond type), each edge latent representation with the edge weight $(\boldsymbol{z}_u, \boldsymbol{z}_v, m)$ is given by a nonlinearity $\theta^{\mathrm{dec}}_{\mathrm{bond}}$, and the outputs are used to construct a softmax distribution from which edge weights can be sampled. Every time a new edge is updated, the corresponding edge weight value also gets updated. A whole molecular graph is decoded out with dynamic recurrent updating of the edges and edge weights.

About 10,000 molecules were sampled respectively from the ZINC and QM9 dataset, and the corresponding average numbers of atoms are 44 and 21. About 5000 samples were used to test the NeVAE model. The vanilla NeVAE on ZINC and QM9 subset had the Q_{valid} of 58.0 and 67.0%, and the Q_{novel} of 100 and 100%. By using masking, the NeVAE had the perfect performance (100%) of Q_{valid} and Q_{novel} on both subsets, which suggested that masking can help to prevent the generation of certain undesirable edges and edges weights, and the decoder can guarantee a set of local structural and functional properties during the generating process. The 3000 molecules (90% for training, 10% for testing) sampled from ZINC dataset were used to develop sparse GP with the latent representations. The LL and RMSE on the test set are -1.631 and 1.231, respectively. BO was applied to NeVAE model for identifying novel molecules with a high value of the penalized logP. Starting from 500 points sampled from the training set, five iterations of 50 latent vectors (samples) were recommended by the sparse GP using the expected improvement heuristic [93], leading to a 100% Q_{valid}, a 66% of molecules with score >0, the average score of 0.36 and the best score of 2.32. A similar VAE architecture [81] with high generative performance was also proposed based on graph encoder (Gated Graph Neural Networks) [94] and graph decoder (recurrent graph-based generative procedure).

Junction tree variational autoencoder

Junction tree VAE (JT-VAE) is a molecular generator based on a tree-structured scaffold over chemical substructures [82]. Based on prior chemistry knowledge, a molecular graph is mapped into a cycle-free junction tree by a tree decomposition algorithm. The architecture is shown in Figure 6C. Formally, given a graph $\mathcal{G}$, the corresponding junction tree $\mathcal{T}_{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \chi)$ contains a set of clusters $\mathcal{V} = \{C_1, \ldots, C_n\}$, edges $\mathcal{E}$ and label vocabulary χ of clusters, each of which $C_i = (\mathcal{V}_i, \mathcal{E}_i)$ is an induced subgraph. The ZINC dataset (250,000 molecules) was used to perform tree decomposition, leading to the vocabulary size $|\chi|$ of 750. The encoder contains a graph encoder $q_\phi(\boldsymbol{z}_{\mathcal{G}}|\mathcal{G})$ with message passing network [74] and a tree encoder $q_\psi(\boldsymbol{z}_{\mathcal{T}}|\mathcal{T})$ with tree message passing network using GRU [95], which are respectively used to capture fine- and coarse-grained connectivity. The latent representation is then decoded back into a molecular graph in two stages. First, a tree decoder $p(\mathcal{T}|\boldsymbol{z}_{\mathcal{T}})$ reproduces a junction tree with two-step prediction topological and label prediction using GRU. Second, considering the reproduced junction tree, a decoder $p(\mathcal{G}|\mathcal{T}, \boldsymbol{z}_{\mathcal{G}})$ is used to predict the fine-grained connectivity to reproduce the full molecular graph.

The ZINC dataset (220,011 for training, 24,445 for validation and 5000 for testing) was used to train the JT-VAE model. Three tasks were tested. First, for molecular reconstruction and validity, since both encoding and decoding processes are stochastic, the Rs was 76.7% for 100 molecules with ten-times encodings and each encoding with ten-times decodings. The Q_{valid} for 1000 sampled latent vectors with 100-times decodings is 100%. Given the latent representations of JT-VAE, the Sparse Gaussian Process (SGP) model was constructed with the LL and RMSE of -1.658 and 1.290, respectively. The batch BO for the target $y(\cdot)$ was performed for five iterations using the expected improvement heuristic. JT-VAE found over 50 molecules with $y(\cdot) \geq 3.5$. In which, the best one had a high value of 5.30. To demonstrate the smoothness of the learned latent space, constrained optimization was performed with jointly training JT-VAE and a property predictor of $y(\cdot)$. Start from 800 molecules with the lowest $y(\cdot)$, 80-step sampling of gradient ascent was applied to decode 80 molecules from the trajectories of the latent space. With a constraint of Tanimoto similarity with 0.4, 80% of decoded molecules have an average similarity of 0.51, with an average property improvement of 0.84.

GAN-based generative models with RL

Generally, VAE models with reasonable latent space typically allow for easier and more stable optimization than implicit generative models such as GAN. However, all of the available VAE-based or RNN-based models on a molecular graph are computationally expensive. Cao *et al.* [87] used implicit GANs combined with RL to produce small molecular graphs with certain desired properties.

Table 3. Overview of the efforts made by companies and academic groups on molecule generation.

Organization	Affiliation	Model type	Ref.
Company	InSilico Medicine, Inc.	Fingerprint-based, SMILES-based	[53,54,59,60,68,69]
	AstraZeneca, Inc.	SMILES-based	[14,27,53]
	Wildcard Pharmaceutical Consulting	SMILES-based	[37]
Academic group	Swiss Federal Institute of Technology (ETH), Schneider *et al.*	SMILES-based	[42,62]
	University of North Carolina, Tropsha *et al.*	SMILES-based	[44]
	University of Tokyo, Tsuda *et al.*	SMILES-based	[46]
	Harvard University, Aspuru-Guzik *et al.*	SMILES-based	[15,55,57]
	Alan Turing Institute, Kusner *et al.*	SMILES-based	[49]
	Georgia Institute of Technology, Dai *et al.*	SMILES-based	[51]
	University of Paris-Est, Simonovsky *et al.*	Graph-based	[78]
	Max Planck Institute for Software Systems, Gomez-Rodriguez *et al.*	Graph-based	[79]
	MIT Computer Science & Artificial Intelligence Lab, Jaakkola *et al.*	Graph-based	[82]
	DeepMind, Li *et al.*	Graph-based	[28]
	Peking University, Liu *et al.*	Graph-based	[29]
	University of Amsterdam, Kipf *et al.*	Graph-based	[87]
	Microsoft Research, Gaunt *et al.*	Graph-based	[81]
	Stanford University, Pande *et al.*	Graph-based	[85]

SMILES: Simplified molecular input line entry specification.

MolGAN

MolGAN is a generator G_θ for molecular generation with a discriminator D_ϕ for AL and a reward network $\hat{R}_\psi$ for RL [87]. As shown in Figure 5C, G_θ takes a sample from a prior distribution and generates an annotated molecular graph $\mathcal{G}$ with an adjacency tensor $\boldsymbol{A} \in \mathbb{R}^{N \times N \times Y}$ and a node feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times T}$ with one-hot encoding, where N is the number of atoms, T is the number of atom types and Y is the number of bond types. D_ϕ learns to distinguish both samples from the training dataset and G_θ. G_θ learns to approximate the distribution of training dataset then generates valid molecules. The $\hat{R}_\psi$ is used to approximate the reward function of a sample and optimize molecule generation toward the external metrics using RL. Both $\hat{R}_\psi$ and D_ϕ are based on the relational-GCN network [96], which is invariant to node order permutations.

The QM9 dataset contains 133,885 organic compounds up to nine heavy atoms: carbon (C), oxygen (O), nitrogen (N) and fluorine (F). MolGAN was trained on the full dataset for generating molecular graphs with the maximum number of atoms nine. And the 6400 samples were used to evaluate the MolGAN. Without the optimization process of RL, the MolGAN showed a Q_{valid}, Q_{unique} and Q_{novel} of 98.1, 10.4 and 94.2%, respectively. Three objectives of QED, SAS and logP were separately and simultaneously optimized with RL. MolGAN obtained the Q_{valid}, Q_{unique} and D_{in} of 98–100%, 2.0—2.3% and 0.93–0.99, respectively. The best values of the three objectives were 0.62, 0.95 and 0.89, respectively.

Conclusion and discussion

In order to accelerate drug discovery, some companies and academic groups in Table 3 have made efforts to implement *de novo* drug design with deep generative models. We have comprehensively reviewed most of these recently reported SMILES- and graph-based generative models and discussed their advantages and limitations, summarized in Figure 7.

For the SMILES-based models, due to the nature of SMILES strings, RNN-based architectures are often ideal candidates for dynamic SMILES generation with char-by-char. Generally, the RNN-only generative models can produce chemically valid SMILES strings after learning the SMILES syntax. In drug discovery, designing molecules with desired molecular properties is of a great importance. There are four optimization strategies to achieve this biased molecular design. With the concept of TL, based on a general model trained on a large dataset, a small dataset with certain desirable properties (such as specific ligands against a biological target) is used to fine-tune this general model for generating biased molecules, which obviously depends on a small known labeled dataset. Seeing the works of ChemVAE, GammerVAE and SD-VAE, a smooth latent space based on SMILES has been tried to build, in which BO can be well performed with an external scoring function. The output molecules can usually obtain a

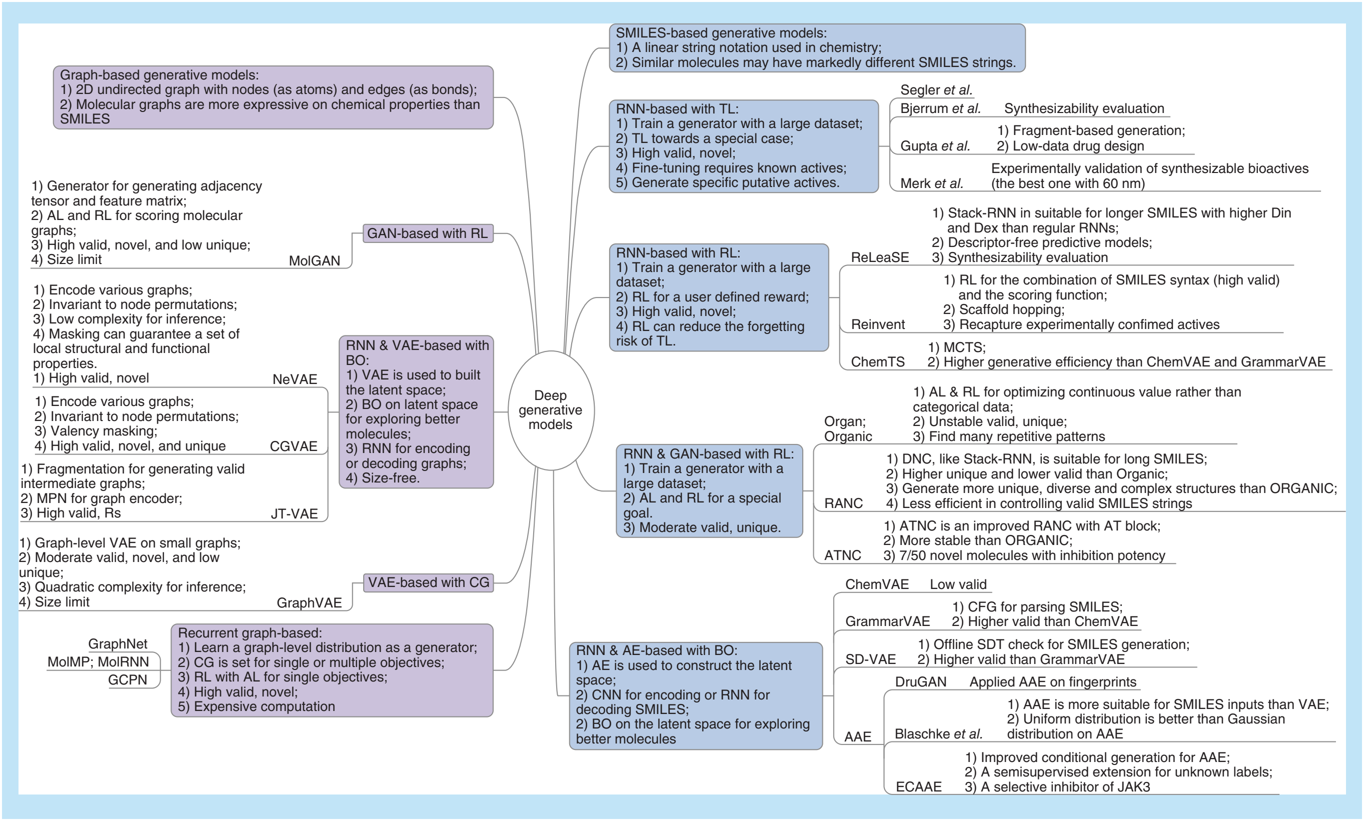
Deep generative models
SMILES-based generative models: 1) A linear string notation used in chemistry; 2) Similar molecules may have markedly different SMILES strings.
RNN-based with TL: 1) Train a generator with a large dataset; 2) TL towards a special case; 3) High valid, novel; 4) Fine-tuning requires known actives; 5) Generate specific putative actives.
Segler et al.
Bjerrum et al. Synthesizability evaluation
Gupta et al. 1) Fragment-based generation; 2) Low-data drug design
Merk et al. Experimentally validation of synthesizable bioactives (the best one with 60 nm)
RNN-based with RL: 1) Train a generator with a large dataset; 2) RL for a user defined reward; 3) High valid, novel; 4) RL can reduce the forgetting risk of TL.
ReLeaSE 1) Stack-RNN in suitable for longer SMILES with higher Din and Dex than regular RNNs; 2) Descriptor-free predictive models; 3) Synthesizability evaluation
Reinvent 1) RL for the combination of SMILES syntax (high valid) and the scoring function; 2) Scaffold hopping; 3) Recapture experimentally confimed actives
ChemTS 1) MCTS; 2) Higher generative efficiency than ChemVAE and GrammarVAE
RNN & GAN-based with RL: 1) Train a generator with a large dataset; 2) AL and RL for a special goal. 3) Moderate valid, unique.
Organ; Organic 1) AL & RL for optimizing continuous value rather than categorical data; 2) Unstable valid, unique; 3) Find many repetitive patterns
RANC 1) DNC, like Stack-RNN, is suitable for long SMILES; 2) Higher unique and lower valid than Organic; 3) Generate more unique, diverse and complex structures than ORGANIC; 4) Less efficient in controlling valid SMILES strings
ATNC 1) ATNC is an improved RANC with AT block; 2) More stable than ORGANIC; 3) 7/50 novel molecules with inhibition potency
RNN & AE-based with BO: 1) AE is used to construct the latent space; 2) CNN for encoding or RNN for decoding SMILES; 2) BO on the latent space for exploring better molecules
ChemVAE Low valid
GrammarVAE 1) CFG for parsing SMILES; 2) Higher valid than ChemVAE
SD-VAE 1) Offline SDT check for SMILES generation; 2) Higher valid than GrammarVAE
AAE
DruGAN Applied AAE on fingerprints
Blaschke et al. 1) AAE is more suitable for SMILES inputs than VAE; 2) Uniform distribution is better than Gaussian distribution on AAE
ECAAE 1) Improved conditional generation for AAE; 2) A semisupervised extension for unknown labels; 3) A selective inhibitor of JAK3
Graph-based generative models: 1) 2D undirected graph with nodes (as atoms) and edges (as bonds); 2) Molecular graphs are more expressive on chemical properties than SMILES
GAN-based with RL
MolGAN 1) Generator for generating adjacency tensor and feature matrix; 2) AL and RL for scoring molecular graphs; 3) High valid, novel, and low unique; 4) Size limit
RNN & VAE-based with BO: 1) VAE is used to built the latent space; 2) BO on latent space for exploring better molecules; 3) RNN for encoding or decoding graphs; 4) Size-free.
NeVAE 1) Encode various graphs; 2) Invariant to node permutations; 3) Low complexity for inference; 4) Masking can guarantee a set of local structural and functional properties. 1) High valid, novel
CGVAE 1) Encode various graphs; 2) Invariant to node permutations; 3) Valency masking; 4) High valid, novel, and unique
JT-VAE 1) Fragmentation for generating valid intermediate graphs; 2) MPN for graph encoder; 3) High valid, Rs
VAE-based with CG
GraphVAE 1) Graph-level VAE on small graphs; 2) Moderate valid, novel, and low unique; 3) Quadratic complexity for inference; 4) Size limit
Recurrent graph-based: 1) Learn a graph-level distribution as a generator; 2) CG is set for single or multiple objectives; 3) RL with AL for single objectives; 4) High valid, novel; 5) Expensive computation
GraphNet
MolMP; MolRNN
GCPN

Figure 7. Summary of all mentioned generative approaches.

better property than the previously sampled molecules. But the Q_{valid} of VAE-SMILES-based models is relatively low. This state has been improved for VAE graph-based models. As for RL, the definition of reward function is a key step especially multiparametric scoring functions revealing combined target characteristics. Nevertheless, current RL-based models only tackle the single-task problem to maximize an expected reward function. By using AL, the model could actively learn valid sequences via discriminating the valid SMILES created by the generator from the training data while sometimes it may suffer in creating simple structures like 'CCCCCCOC' without constraints, and that is why it requires RL to define reward functions to mitigate such a phenomenon. Most current emerging models such as ATNC and RANC, two most promising models, are tracking the combination of AL and RL.

In addition, the SMILES-based models still have other problems. First, the selection of model evaluation metrics is of the most important issues to address. For example, the judgment of Q_{unique}, Q_{novel}, and diversity of generated molecules in different models are not easy to compare as they applied different datasets and focused on different reward functions. Another common problem of SMILES-based generation lies in fragment-based growing. As the fragments of molecules are not well defined in SMILES notation, it is much more difficult to perform fragment-based generation in SMILES-based models than that in graph-based ones. Moreover, multiobjective reward functions are hard to be trained smoothly in current SMILES-based models via a simultaneous way since they might be opposed to each other.

The graph-based generative models are directly trained on 2D molecular graphs with nodes as atoms and edges as bonds. Due to discrete graph structures with arbitrary connectivity, learning to generate graphs is difficult. All the graph-based architectures are carefully elaborated for graph generation. One-shot prediction of an adjacency tensor at once was adopted by GraphVAE and MolGAN to output the connectivity of atoms, which is most likely feasible only for graphs of small size of nine heavy atoms. But it usually leads to a low validity and uniqueness for a larger size of molecular graphs. In drug discovery, the size of molecular graphs is always varied and larger than ten. Stepwise graph-based generative models are proposed for solving these issues. And more computational decisions are introduced to incrementally construct molecular graphs in the way of atom-by-atom (GraphNet, MolMP and MolRNN), edge-by-edge (NeVAE) and fragment-by-fragment (JT-VAE). The canonical ordering

Table 4. Overview of related datasets used by generative models.

Database	Types of molecules	Number of the total molecules	Ref.
ChEMBL	Bioactive drug-like small molecules	1.8 million	[32]
ZINC	Commercially available compounds for virtual screening	230 million	[38]
QM9	Up to nine heavy atoms with quantum chemistry structures and properties	134,000	[67]
ChemDiv	Shelf-available screening compounds	1.6 million	[104]

from RDKit is preferred to learn a molecular graph. However, Li *et al.* empirically found some evidence that the canonical ordering is not always the best ordering, and the linearization order matters when learning a graph [13]. It may suggest that there is a big potential for actually learning an ordering. In fact, the order-free algorithms are proposed to cleverly sidestep this hurdle, such as node (atom) embedding (such as GCN, relational-GCN) and graph matching, which are invariant to atom permutations. What is more, the trick of masking and tree decomposition is used to significantly improve the validity (100%) in molecule generation, which preserves the chemical feasibility of atom valence and intermediate structures, respectively.

With these above efforts, the graph-based generative models usually have a higher validity and novelty than the SMILES-based models, even reaching to a perfect level for validity. But such models often overlook the uniqueness of generated molecules especially for GraphVAE and MolGAN. The reconstruction ability for both generative models often seems to be neglected. Considering the attention of the beginning researches, it should be an important metric for evaluating the learning ability. D_{in} and D_{ex} are also seldom measured together. Thus seven metrics should be recommended to evaluate molecular generative models: Rs, Q_{valid}, Q_{unique}, Q_{novel}, D_{in}, D_{ex} and Fréchet ChemNet distance.

Taking into account the requirements of *de novo* drug design, different conditions are designed for optimizing molecular generation. The property-based optimization of logP, SAS and QED is often popularly performed using BO, RL or conditioning code for improving the target properties, leading to a positive shift of the corresponding property distribution. In which, conditioning code can be used as label features for optimizing different properties simultaneously, instead of directly optimizing the combination of these properties (ChemVAE) or alternately optimizing these properties (ORGAN). The scaffold- or fragment-based generation is performed given a scaffold or fragment of the known active molecule, which is helpful for hit-to-lead optimization. The optimization of molecule generation against targets is challenged by a few generative models, which show promising results for biased drug design. However, this process extremely depends on a set of known labeled actives.

Based on the reported results, it is believed that generative neural networks can be established as promising directions in *de novo* drug discovery. However, for drug design in chemistry, it is far from enough for generating molecules that are almost right, even bioactive, because of unimaginable complexity. For such a 'needle in the haystack' problem, more experimental evidences still need to be provided for explaining the charm of deep generative models.

Future perspective

De novo drug design has developed over several decades. Both ligand- and structure-based methods have made great progress. Structure-based drug design methods provide powerful tools for drug discovery and can reduce drug R&D time and cost. LigBuilder [2,3] is one of the typical examples for structure-based drug design methods. These methods construct novel molecules by assembling small pieces (atoms or fragments) together into a binding pocket of interest. The attempt to design innovative bioactive molecules is actually a search process through a virtually infinite chemical space, which can be regarded as a multidimensional descriptor space containing all the small molecules that could in principle be created. A variety of combinatorial search strategies, such as breadth- and depth-first search, Monte Carlo search and evolutionary algorithms have been successfully applied to find novel drug-like molecules from the chemical space, though they still confront many limitations [97]. New algorithms need to be proposed to solve the problem. In the recent years, deep learning methods have walked into the spotlight for their great breakthroughs in computer vision, speech recognition and natural language processing and so on. It is deserved to introduce the powerful tools to *de novo* drug design for their potential to solve the difficulties faced in traditional drug design methods. According to our review, all the deep-learning-based *de novo* drug design methods published until now are ligand-based methods, which means that they learn information only from known ligands represented by SMILES or graph. Obviously, these methods showed much difference to traditional structure-based drug design methods. Some difficulties hinder the application of deep learning to structure-based drug design. The first problem is the representation of 3D protein and ligand structures. The performance of deep learning methods often depends on the quality of data representation. Good representations efficiently capture the most critical information, while poor representations create a noisy distribution. The second problem is conformation optimization. It is not necessary to consider the bond rotation and fragment orientation when generating SMILES or graphs, but it is one of the most important aspects when generating 3D chemical structures because we need to find the best conformation of the molecule. The structure diversity sharply increases when considering the 3D information, challenging the power of deep learning methods. The third difficulty is scoring function. The training of machine learning significantly relies on the feedback of result metrics. For drug design methods, scoring function was used to discord the negative designs and pick the best compounds with best conformation from a large pool of accessible possibilities. The performance of drug design model depends on the quality of scoring functions. Though important, finding a scoring function with high accuracy remains difficult. Some of the recent works apply deep learning methods to scoring functions and successfully increase the accuracy. We believe more accurate scoring functions can be originated from advanced machine learning methods [98–102]. Segler *et al.* [103] combined Monte Carlo tree search with deep neural network and symbolic rules to perform chemical synthesis planning. Deep learning methods work effectively both in scoring function and in retrosynthetic analysis, and we are optimistic that these methods can be further applied to generate 3D chemical structures with the constraints of target information.

Virtual screening is a popular computer-based drug discovery method by screening out the most promising compounds from large chemical databases. Though virtual libraries can be used, virtual screening can only explore limited chemical space. Generative models belong to *de novo* design approach and novel molecular structures with specific properties can be generated. Most compounds in a chemical database can be synthesized and purchased while the generated structures may not be synthesis accessible, which can be improved by integrating synthesis accessible analysis tool in the future.

Executive summary

Generative neural networks are competent for molecular generation

- Both simplified molecular input line entry specification- and graph-based generative models can powerfully capture hidden rules via learning a given large dataset (Table 4).
- Based on what to learn, the generative models can create novel and valid molecules with similar distributions of molecular descriptors.

Objective optimization of generated molecules is essential to *de novo* drug design

- Transfer learning is a simple and popular optimization strategy depending on fine-tuning a general model using a small dataset with certain specific property. It is often applied to recurrent neural network-only-based models.
- Bayesian optimization is a sequential optimization strategy based on a smooth latent space only constructed by autoencoder models to find the best molecules with certain optimal property.
- Reinforcement learning is a preferable optimization strategy via jointly training, not only preserving the generative ability, but maximizing the reward objective about certain property. Generative adversarial network- and recurrent neural network-based generative models often use this method for biased drug design.
- Conditional generation is a spring-up optimization strategy. During the optimization process, one or more objectives are converted into one or more conditional codes, which are concatenated to the original input vector for guiding biased molecular generation. It can be applied to multiobjective optimization, and may be suitable for the four generative models with ingenious computational design.

Systemic evaluation of molecular generation is crucial for effective & practical *de novo* drug design

- Generative quality level: Q_{valid}, Q_{unique}, Q_{novel}, and Rs.
- Molecular structural level: D_{in} and D_{ex}, calculated with extended-connectivity fingerprint-based Tanimoto distance; Fréchet ChemNet distance (FCD).
- Molecular property level: D_{KL} and D_{JS} based on molecular properties, including quantitative estimate of drug-likeness, logP; FCD.
- Molecular synthetic level: synthetic accessibility scoring and retrosynthetic analysis.

Financial & competing interests disclosure

The authors have no relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript. This includes employment, consultancies, honoraria, stock ownership or options, expert testimony, grants or patents received or pending, or royalties.

No writing assistance was utilized in the production of this manuscript.

References

1. Hartenfeller M, Schneider G. *De novo* drug design. In: *Chemoinformatics and Computational Chemical Biology.* Humana Press, NJ, USA, 299–323 (2010).
2. Wang R, Gao Y, Lai L. LigBuilder: a multi-purpose program for structure-based drug design. *Mol. Model. Annu.* 6(7–8), 498–516 (2000).
3. Yuan Y, Pei J, Lai L. LigBuilder 2: a practical *de novo* drug design approach. *J. Chem. Inf. Model.* 51(5), 1083–1091 (2011).
4. Kutchukian PS, Shakhnovich EI. *De novo* design: balancing novelty and confined chemical space. *Expert Opin. Drug Discov.* 5(8), 789–812 (2010).
5. Sheng C, Zhang W. Fragment informatics and computational fragment-based drug design: an overview and update. *Med. Res. Rev.* 33(3), 554–598 (2013).
6. Jaques N, Gu S, Bahdanau D, Hernández-Lobato JM, Turner RE, Eck D. Sequence tutor: conservative fine-tuning of sequence generation models with KL-control. Presented at: *34th International Conference on Machine Learning.* Sydney, Australia, 6–11 August 2017.
7. Choi Y, Choi M-J, Kim M, Ha J-W, Kim S, Choo J. StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. Presented at: *2018 IEEE Conference on Computer Vision and Pattern Recognition.* Salt Lake City, AL, USA, 18–23 June 2018.
8. Elgammal AM, Liu B, Elhoseiny M, Mazzone M. CAN: creative adversarial networks, generating "art" by learning about styles and deviating from style norms. Presented at: *8th International Conference on Computational Creativity.* Atlanta, GA, USA, 20–22 June 2017.
9. Zhu J-Y, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. Presented at: *2017 IEEE International Conference on Computer Vision.* Venice, Italy, 22–29 October 2017.
10. Yi Z, Zhang H, Tan P, Gong M. DualGAN: unsupervised dual learning for image-to-image translation. Presented at: *2017 IEEE International Conference on Computer Vision.* IEEE Computer Society, Venice, Italy, 22–29 October 2017.
11. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. Presented at: *2017 IEEE Conference on Computer Vision and Pattern Recognition.* Venice, Italy, 22–29 October 2017.
12. Kusner MJ, Hernández-Lobato JM. Gans for sequences of discrete elements with the gumbel-softmax distribution. Presented at: *30th Annual Conference on Neural Information Processing Systems.* Barcelona, Spain, 5–10 December 2016.
13. Riedel S, Bosnjak M, Rocktäschel T. Programming with a differentiable forth interpreter. Presented at: *Proceedings of the 34th International Conference on Machine Learning.* Sydney, Australia, 6–11 August 2017.
14. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* 4(1), 120–131 (2018).
15. Gómez-Bombarelli R, Duvenaud DK, Hernández-Lobato JM *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* 4(2), 268–276 (2018).
16. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* 28(1), 31–36 (1988).
17. Mikolov T, Karafiát M, Burget L, Černocky J, Khudanpur S. Recurrent neural network based language model. Presented at: *11th Annual Conference of the International Speech Communication Association.* Makuhari, Chiba, Japan, 26–30 September 2010.
18. Kingma DP, Welling M. Auto-encoding variational bayes. Presented at: *2nd International Conference on Learning Representations.* Banff, Canada, 14–16 April 2014.
19. Rezende DJ, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. Presented at: *31st International Conference on Machine Learning.* Beijing, China, 21–26 June 2014.
20. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial autoencoders. Presented at: *4th International Conference on Learning Representations.* San Juan, PR, USA, 2–4 May 2016.
21. Goodfellow I, Pouget-Abadie J, Mirza M *et al.* Generative adversarial nets. Presented at: *27th Annual Conference on Neural Information Processing Systems.* Montreal, Quebec, Canada, 8–13 December 2014.
22. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 9(8), 1735–1780 (1997).
23. Cho K, Van Merriënboer B, Gulcehre C *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. Presented at: *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar, 25–29 October 2014.
24. Rogers D, Hahn M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* 50(5), 742–754 (2010).

25. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. Presented at: *31th Conference on Neural Information Processing Systems.* Long Beach, CA, USA, 4–9 December 2017.

26. Preuer K, Renz P, Unterthiner T, Hochreiter S, Klambauer G. Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.* 58(9), 1736–1741 (2018).

27. Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular *de-novo* design through deep reinforcement learning. *J. Cheminform.* 9(1), 1–14 (2017).

28. Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P. Learning deep generative models of graphs. Presented at: *6th International Conference on Learning Representations.* Vancouver, BC, Canada, 30 April–3 May 2018.

29. Li Y, Zhang L, Liu Z. Multi-objective *de novo* drug design with conditional graph generative model. *J. Cheminform.* 10(1), 33 (2018).

30. Rasmussen CE. Gaussian processes in machine learning. In: *Advanced Lectures on Machine Learning.* Springer, Berlin, Germany, 63–71 (2004).

31. Dai Z, Damianou A, González J, Lawrence N. Variational auto-encoded deep Gaussian processes. Presented at: *4th International Conference on Learning Representations.* PR, USA, 2–4 May 2016.

32. Gaulton A, Bellis LJ, Bento AP *et al.* ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* 40(D1), D1100–D1107 (2011).

33. Cirecsan DC, Meier U, Schmidhuber J. Transfer learning for Latin and Chinese characters with deep neural networks. Presented at: *2012 International Joint Conference on Neural Networks.* Brisbane, Australia, 10–15 June 2012.

34. Cook Jr EH, Fletcher KE, Wainwright M, Marks N, Yan S, Leventhal BL. Primary structure of the human platelet serotonin 5-HT2A receptor: identity with frontal cortex serotonin 5-HT2A receptor. *J. Neurochem.* 63(2), 465–469 (1994).

35. Rich SM, Leendertz FH, Xu G *et al.* The origin of malignant malaria. *Proc. Natl Acad. Sci. USA* 106(35), 14902–14907 (2009).

36. Masalha M, Borovok I, Schreiber R, Aharonowitz Y, Cohen G. Analysis of transcription of the *Staphylococcus aureus* aerobic class Ib and anaerobic class III ribonucleotide reductase genes in response to oxygen. *J. Bacteriol.* 183(24), 7260–7272 (2001).

37. Bjerrum EJ, Threlfall R. Molecular generation with recurrent neural networks (RNNs). *CoRR*, abs/1705.04612 (2017).

38. Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG. ZINC: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.* 52(7), 1757–1768 (2012).

39. Wildman SA, Crippen GM. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.* 39(5), 868–873 (1999).

40. Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* 1(1), 8 (2009).

41. Wiley ChemPlanner. https://www.cas.org/products/scifinder-n/chemplanner

42. Gupta A, Müller AT, Huisman BJH, Fuchs JA, Schneider P, Schneider G. Generative recurrent networks for *de novo* drug design. *Mol. Inform.* 37(1–2), 1700111 (2018).

43. REINVENT. https://github.com/MarcusOlivecrona/REINVENT

44. Popova M, Isayev O, Tropsha A. Deep reinforcement learning for *de novo* drug design. *Sci. Adv.* 4(July), 1–15 (2018).

45. ReLeaSE. https://github.com/isayev/ReLeaSE

46. Yang X, Zhang J, Yoshizoe K, Terayama K, Tsuda K. ChemTS: an efficient python library for *de novo* molecular generation. *Sci. Technol. Adv. Mater.* 18(1), 972–976 (2017).

47. ChemTS. https://github.com/tsudalab/ChemTS

48. ChemVAE. https://github.com/aspuru-guzik-group/chemical_vae

49. Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. Presented at: *34th International Conference on Machine Learning.* Sydney, Australia, 6–11 August 2017.

50. Grammar VAE. https://github.com/mkusner/grammarVAE

51. Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for structured data. Presented at: *6th International Conference on Learning Representations.* Vancouver, BC, Canada, 30 April–3 May 2018.

52. SD-VAE. https://github.com/Hanjun-Dai/sdvae

53. Blaschke T, Olivecrona M, Engkvist O, Bajorath J, Chen H. Application of generative autoencoder in *de novo* molecular design. *Mol. Inform.* 37(1–2), 1700123 (2018).

54. Polykovskiy D, Zhebrak A, Vetrov D *et al.* Entangled conditional adversarial autoencoder for *de novo* drug discovery. *Mol. Pharm.* 15(10), 4398–4405 (2018).

55. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. *CoRR*, abs/1705.10843 (2017).

56. ORGAN. https://github.com/gablg1/ORGAN

57. Sanchez-Lengeling B, Outeiral C, Guimaraes GL, Aspuru-Guzik AA. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv.*, 5309668 (2017).

58. ORGANIC. https://github.com/aspuru-guzik-group/ORGANIC

59. Putin E, Asadulaev A, Vanhaelen Q *et al.* Adversarial threshold neural computer for molecular *de novo* design. *Mol. Pharm.* 15, 4386–4397 (2018).

60. Putin E, Asadulaev A, Ivanenkov Y *et al.* Reinforced adversarial neural computer for *de novo* molecular design. *J. Chem. Inf. Model.* 58(6), 1194–1204 (2017).

61. Bento AP, Gaulton A, Hersey A *et al.* The ChEMBL bioactivity database: an update. *Nucleic Acids Res.* 42(D1), D1083–D1090 (2014).

62. Merk D, Friedrich L, Grisoni F, Schneider G. *De novo* design of bioactive small molecules by artificial intelligence. *Mol. Inform.* 37(1–2), 1700153 (2018).

63. Sundermeyer M, Schlüter R, Ney H. LSTM neural networks for language modeling. Presented at: *13th Annual Conference of the International Speech Communication Association.* Portland, OR, USA, 9–13 September 2012.

64. Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8(3–4), 229–256 (1992).

65. Madras BK. History of the discovery of the antipsychotic dopamine D2 receptor: a basis for the dopamine hypothesis of schizophrenia. *J. Hist. Neurosci.* 22(1), 62–78 (2013).

66. Tetko IV, Sushko Y, Novotarskyi S *et al.* How accurately can we predict the melting points of drug-like compounds? *J. Chem. Inf. Model.* 54(12), 3320–3329 (2014).

67. Ramakrishnan R, Dral PO, Rupp M, Von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* 1, 140022 (2014).

68. Kadurin A, Aliper A, Kazennov A *et al.* The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget.* 8(7), 10883–10890 (2016).

69. Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. druGAN: an advanced generative adversarial autoencoder model for *de novo* generation of new molecules with desired molecular properties *in silico*. *Mol. Pharm.* 14(9), 3098–3104 (2017).

70. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* 31(2), 455–461 (2010).

71. Yu L, Zhang W, Wang J, Yu Y. SeqGAN: sequence generative adversarial nets with policy gradient. Presented at: *31st AAAI Conference on Artificial Intelligence.* San Francisco, CA, USA, 4–9 February 2017.

72. Lipinski CA, Lombardo F, Dominy BW, Feeney PJ. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* 23(1–3), 3–25 (1997).

73. Graves A, Wayne G, Reynolds M *et al.* Hybrid computing using a neural network with dynamic external memory. *Nature* 538(7626), 471 (2016).

74. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. Presented at: *34th International Conference on Machine Learning.* Sydney, Australia, 6–11 August 2017.

75. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided. Mol. Des.* 30(8), 595–608 (2016).

76. Duvenaud DK, Maclaurin D, Iparraguirre J *et al.* Convolutional networks on graphs for learning molecular fingerprints. Presented at: *29th Annual Conference on Neural Information Processing Systems.* Montreal, Canada, 7–12 December 2015.

77. Ktena SI, Parisot S, Ferrante E *et al.* Distance metric learning using graph convolutional networks: application to functional brain networks. Presented at: *20th Medical Image Computing and Computer Assisted Intervention.* Quebec City, Quebec, Canada, 10–14 September 2017.

78. Simonovsky M, Komodakis N. GraphVAE: towards generation of small graphs using variational autoencoders. Presented at: *27th International Conference on Artificial Neural Networks.* Rhodes, Greece, 4–7 October 2018.

79. Samanta B, De A, Ganguly N, Gomez-Rodriguez M. Designing random graph models using variational autoencoders with applications to chemical design. Presented at: *35th International Conference on Machine Learning.* Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018.

80. NeVAE. https://github.com/Networks-Learning/nevae

81. Liu Q, Allamanis M, Brockschmidt M, Gaunt AL. Constrained graph variational autoencoders for molecule design. Presented at: *32nd Annual Conference on Neural Information Processing Systems.* Montreal, Canada, 3–8 December 2018.

82. Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. Presented at: *35th International Conference on Machine Learning.* Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018.

83. JT-VAE. https://github.com/wengong-jin/icml18-jtnn

84. MolMP and MolRNN. https://github.com/kevinid/molecule_generator

85. You J, Liu B, Ying R, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. Presented at: *32nd Annual Conference on Neural Information Processing Systems.* Montreal, Canada, 3–8 December 2018.

86. GCPN. https://github.com/bowenliu16/rl_graph_generation

87. De Cao N, Kipf T. MolGAN: an implicit generative model for small molecular graphs. Presented at: *35th International Conference on Machine Learning.* Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018.

88. MolGAN. https://github.com/nicola-decao/MolGAN

89. Ip YT, Davis RJ. Signal transduction by the c-Jun N-terminal kinase (JNK) – from inflammation to development. *Curr. Opin. Cell Biol.* 10(2), 205–219 (1998).

90. Stambolic V, Woodgett JR. Mitogen inactivation of glycogen synthase kinase-3$β$ in intact cells via serine 9 phosphorylation. *Biochem. J.* 303(3), 701–704 (1994).

91. Simonovsky M, Komodakis N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. Presented at: *2017 IEEE Conference on Computer Vision and Pattern Recognition.* Honolulu, Hawaii, 21–26 July 2017.

92. Cho M, Sun J, Duchenne O, Ponce J. Finding matches in a haystack: a max-pooling strategy for graph matching in the presence of outliers. Presented at: *2014 IEEE Conference on Computer Vision and Pattern Recognition.* Columbus, OH, USA, 23–28 June 2014.

93. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* 13(4), 455–492 (1998).

94. Li Y, Tarlow D, Brockschmidt M, Zemel R. Gated graph sequence neural networks. Presented at: *4th International Conference on Learning Representations.* San Juan, PR, USA, 2–4 May 2016.

95. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. Presented at: *28th Aunnal Conference on Neural Information Processing Systems.* Montreal, Canada, 8–13 December 2014.

96. Schlichtkrull M, Kipf TN, Bloem P *et al.* Modeling relational data with graph convolutional networks. Presented at: *15th Extended Semantic Web Conference.* Heraklion, Crete, Greece, 3–7 June 2018.

97. Schneider G, Fechner U. Computer-based *de novo* design of drug-like molecules. *Nat. Rev. Drug Discov.* 4(8), 649 (2005).

98. Wallach I, Dzamba M, Heifets A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855 (2015).

99. Pereira JC, Caffarena ER, dos Santos CN. Boosting docking-based virtual screening with deep learning. *J. Chem. Inf. Model.* 56(12), 2495–2506 (2016).

100. Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein–ligand binding affinity. *CoRR*, abs/1703.10603 (2017).

101. Ericksen SS, Wu H, Zhang H *et al.* Machine learning consensus scoring improves performance across targets in structure-based virtual screening. *J. Chem. Inf. Model.* 57(7), 1579–1590 (2017).

102. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR. Protein–ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.* 57(4), 942–957 (2017).

103. Segler MHS, Preuss M, Waller MP. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* 555(7698), 604 (2018).

104. ChemDiv. http://www.chemdiv.com