



Generative Deep Learning for Targeted Compound Design

Tiago Sousa, João Correia, Vítor Pereira, and Miguel Rocha*

Cite This: J. Chem. I	nf. Model. 2021, 61, 5343–5361	🔇 Read C	Dnline	
ACCESS	III Metrics & More		E Article Recommendations	

ABSTRACT: In the past few years, *de novo* molecular design has increasingly been using generative models from the emergent field of Deep Learning, proposing novel compounds that are likely to possess desired properties or activities. *De novo* molecular design finds applications in different fields ranging from drug discovery and materials sciences to biotechnology. A panoply of deep generative models, including architectures as Recurrent Neural Networks, Autoencoders, and Generative Adversarial Networks, can be trained on existing data sets and provide for the generation of novel compounds. Typically, the new compounds follow the



same underlying statistical distributions of properties exhibited on the training data set Additionally, different optimization strategies, including transfer learning, Bayesian optimization, reinforcement learning, and conditional generation, can direct the generation process toward desired aims, regarding their biological activities, synthesis processes or chemical features. Given the recent emergence of these technologies and their relevance, this work presents a systematic and critical review on deep generative models and related optimization methods for targeted compound design, and their applications.

KEYWORDS: Deep Learning, De Novo Molecular Design, Architectures, Recurrent Neural Networks, Generative Adversarial Networks, Autoencoders, Generative Model, Optimization

INTRODUCTION

De novo molecular design aims to create new chemical entities with desired properties and/or activities. These properties may be easily quantifiable, such as molecular weight, or somewhat more abstract, as is the case of toxicity. This is an inherently difficult task owing to the immense search space of around 10^{33} — 10^{80} feasible molecules from which only a small fraction typically have the desired traits.¹ As such, *de novo* molecular design was, for many years, and mostly remains a process of almost exclusive trial and error, with human expert knowledge and intuition about chemistry playing a major role.²

Meanwhile, the high costs associated with developing new molecules, reaching \$2.8 billion dollars for a single compound, have also led to the implementation of computational tools capable of assisting the process. These have proven valuable and have found wide usage in practical applications.^{2,3} A forthright approach consists in enumerating all possible molecules that conform to valency rules and do not include chemically unstable functional groups. A notable example is the Chemical Space project, where this technique was employed to generate 166 billion molecules.^{4,5} Another technique, reaction-based *de novo* design, uses a set of known chemical reactions to combine various readily available building blocks into new molecules. This process can be guided by a similarity criterion to a known molecule of interest, giving rise to a large number of new similar molecules while ensuring their synthetic plausibility.^{5,6}

Evolutionary Algorithms (EAs) have also been successfully applied to *de novo* molecular design. As a recent example, AutoGrow4⁷ uses an EA to create new predicted ligands. At each iteration, new molecules are created using a mutation operator, that performs an *in silico* chemical reaction, or a crossover operator that merges two compounds into a new one by randomly combining their decorating moieties. Grammatical Evolution on string representations and evolving molecular graphs provide alternative approaches that enable EAs to generate novel compounds targeting desired properties.^{8,9}

Although useful, these methods still leave room for improvement. For instance, enumeration often leads to molecules that are too difficult to synthesize, and reaction-based design is fundamentally restricted in its ability to explore the chemical space, both important aspects of molecular design. EAs, while computationally efficient and capable of performing on par with other recent approaches, rely on expertly encoded operations, possibly limiting the search space and not leveraging the large amounts of data currently available.¹⁰

Received: December 29, 2020 Published: October 26, 2021



In the second se

© 2021 The Authors. Published by American Chemical Society

<table-cell>20to do to do

Figure 1. Acetaminophen (center) under various molecular representations. Top-left: Sequence based representations. Prior to being fed to the models, these sequences are also usually one-hot encoded. Top-right: Graph-based representations. While connection matrices are a suitable input for standard architectures, graphs can also be directly handled using graph neural networks. Bottom: Three dimensional representations, images from PubChem.²⁶ Graphs may be enhanced by including 3D information as node attributes, such as internal distances and angles, or based on a coordinate system such as Cartesian space. Molecular surfaces can be voxelized into a 3D grid for easier processing.

3D based

More recently, advances in Deep Learning (DL) sparked a surge of novel approaches to these problems.¹¹ Over the past decade, DL has proven successful in multiple fields, such as computer vision, speech recognition, and translation, pushing the state-of-the-art forward and surpassing other Machine Learning approaches.^{12,13} DL refers to the use of artificial neural networks with multiple hidden layers.¹² A common intuition is that successive layers can learn higher-level abstractions of the inputs.

Within DL, generative modeling aims at capturing an underlying data generation process, some unknown probabilistic distribution from which a data set was sampled, and has been successfully used for creative tasks, such as writing, composing music, and painting.¹⁴ It usually deals with unlabeled data and attempts to create a model capable of generating new observations that closely resemble those from the training data and not simply producing copies.¹⁴

Improving on earlier approaches, which employed more traditional machine learning methods such as Gaussian Mixture Models, deep generative models have recently found use in the generation of novel molecular entities.^{14–16} Starting around 2017, with works like that of Gomez-Bombarelli et al.,¹⁰ Yuan et al.¹⁷ and Segler et al.,¹⁸ a large number of novel approaches have

been put forward employing various neural network architectures and molecular representations.^{16,19} Alongside these works, several reviews have also sought to condense the plethora of different approaches, shedding light into and discussing the different architectures, generation of various molecular representations and use of comparative metrics.^{16,19–22}

Due to the vastness of chemical space and the costs associated with testing possible compounds, a rational exploration with regard to the desired properties is preferable. As such, a number of distinct approaches have been developed for directing and controlling the generating process of molecules toward compounds with defined chemical properties or desired activities.

In this arena, the evolution has been very fast, with novel methods appearing at an impressive rate. A systematic review of the main advances of DL in the generation of focused molecules seems particularly relevant for practitioners interested in understanding the main features of each method, their main advantages, and their limitations. Several reviews covering the recent explosion of interest in generating molecules leveraging DL have been published. These, however, have mainly focused on the various architectures and molecular representations while only briefly touching on the various methods for the targeted

generation of compounds. Particularly, Schwalbe-Koda and Gomez-Bombarelli²⁰ mention some of these methods, but they primarily focused on the several molecular generation schemes. Likewise, several other reviews also make note of a couple of these methods while not delving into further discussion.^{19,21–24} Closer to our work, Sanchez-Lengeling and Aspuru-Guzik¹⁶ directly discussed the inverse molecular design problem, touching on some methods for controlling the properties of generated molecules.

Notwithstanding previous efforts on reviewing this field, we feel that a more rigorous approach to this subject, containing a more systematic coverage of the methods, can be important for researchers working on these topics. To that end, here we aim to provide a comprehensive review of DL methods for the targeted generation of novel compounds. As such, after an introduction to molecular representations, we present the most common deep generative models and the underlying neural network architectures. We, then, focus on the different optimization approaches that allow to focus the search on molecules with desired properties or activities, closing with a review of the main practical applications.

REPRESENTING MOLECULES

The use of Machine Learning (ML) for chemical applications requires the conversion of chemical compounds to a machinereadable format suitable for computer processing. Although trivial names such as "benzene" and "caffeine" are easy to remember, they carry little to no information on the structure and properties of the underlying compound. Systematic terminology, such as the International Union of Pure and Applied Chemistry (IUPAC) nomenclature of organic chemistry, can be very lengthy and not specify the full structure of a compound. As such, several notation systems have been developed to provide a suitable form of representing molecules,²⁵ each with a distinct impact on the performance and outcomes of chemical ML methods. These are reviewed next and are also illustrated by the example provided by Figure 1.

1D Sequences. Line notations, or sequences, represent chemical structures as human readable strings of characters. Although many line notations have been developed, such as the SYBYL²⁷ and the Wiswesser²⁸ line notations, some have achieved greater popularity. Next, we emphasize the linear notations more commonly used in DL models for compound generation.

SMILES. The Simplified Molecular Input Line Entry System (SMILES) notation allows the representation of molecules as sequences of "tokens". It is built through a depth-first traversal of the molecular graph, encoding atoms and how they connect in a simple and machine-friendly form.²⁹ As the process of encoding a molecule into a string can start at different locations of the molecule, there exists a one-to-many relationship where a single molecule can be represented by several different SMILES. As such, canonicalization algorithms have been developed to ensure that a one-to-one relationship is possible.^{25,30} The SMILES syntax also presents an added difficulty.

Rings and branches require symbols to occur in pairs which often leads to syntactically invalid SMILES strings. Interestingly, the one-to-many relationship between molecules and noncanonical SMILES can be leveraged to perform data augmentation in DL approaches, by allowing one to expand a given data set with the various possible SMILES of the molecules it contains. This was first suggested by Bjerrum³¹ to improve on prediction tasks and later applied to molecular generation by Bjerrum and Sattarov³² followed by several other teams.^{33–36}

InChl. The International Chemical Identifier (InChI) system, proposed by the IUPAC, consists of a notation language that represents molecules as layered strings of characters and aims to be a machine-readable unique representation of a structure.³⁷ Here, molecules are encoded as predefined layers of information that are arranged in a specific order. InChI starts by specifying a core parent structure to which further information may be added. Each layer is separated by a delimiter "/" and prefixed by a lower case letter identifying the layer (except for the first layer).

Although very versatile in representing compounds, generating molecules represented as InChI in DL has not been successful. This was first observed by Gomez-Bombarelli et al.¹⁰ who attributed this result to the added complexity of the InChI syntax, when compared with SMILES. A later work by Winter et al.³⁸ also noted inferior performance when translating from SMILES to InChI, reporting that the model failed to learn, identifying the same probable cause. Nonetheless, InChI provides a unique identifier for chemical structures which can be exploited to derive a canonical SMILES string.

DeepSMILES. O'Boyle and Dalke³⁹ proposed DeepSMILES, an adaptation of the SMILES syntax focusing on two of the major causes of generating invalid SMILES, unmatched ring and parentheses closures. This is addressed by using a single symbol to indicate rings and by denoting branches solely using closing parentheses.

SELFIES. Krenn et al.⁴⁰ introduced a new linear notation for constrained graphs termed SELFreferencIng Embedded Strings (SELFIES). It is capable of enforcing the generation of syntactically and semantically valid graphs and is readily translated to and from them. The team compared its performance against SMILES, DeepSMILES, and Kusner et al. Grammar-VAE, reporting improvements on validity, reconstruction accuracy, and diversity of generated molecules.

2D (Chemical) Structures. A popular solution is to store compounds as graphs, where nodes represent atoms and edges represent bonds. These molecular graphs are commonly implemented using an adjacency matrix that specifies which atoms are connected and the respective bond order/type. Furthermore, nodes and edges can have associated properties, such as, and respectively, relative spatial location and bond order/type. This format allows the encoding of detailed topological data in a readily processable form.⁴¹

3D Structures. Representing a molecule as a simple connection table of atoms overlooks its three-dimensional conformation and consequently disregards valuable information. Representing the arrangement of atoms in space can be accomplished by coupling a coordinate system to a connection table. A common solution is to represent the molecule inside a Cartesian space, assigning each atom in the connection table spatial coordinates (*x*, *y*, *z*).²⁵ An alternative is to use internal coordinates, such as bond length, bond angle, and torsion angle, to describe the position of each atom relative to its neighbors, foregoing the need for a fixed coordinate system.²⁵

Although useful, these two solutions only depict a molecule as a three-dimensional graph without any volume. In reality, a molecule has an electron cloud surrounding its atoms from which many of its properties arise. In molecular surfaces, a molecule is represented as a closed surface that delimits the volume it occupies. Particularly, this surface outlines a threshold value of electron density in the electron cloud that surrounds the molecule. The molecular surface can also be associated with properties, such as the electrostatic or hydrophobicity potential at particular locations.²⁵

Databases. There are currently various repositories offering vast collections of molecules. Some are more specialized, providing focused libraries of known active compounds, such as DrugBank,⁴² while others store more diverse compounds as is the case of ChEMBL.⁴³ The assorted molecular representations can usually be directly obtained from a repository or derived from SMILES or InChI in a straightforward manner using a cheminformatics package, such as the open source toolkit RDKit⁴⁴ or the Chemistry Development Kit (CDK).⁴⁵ Several of commonly used databases in the *de novo* drug design are identified in Table 1.

Table 1. Databases of Interest^a

database	molecules	information			
ChEMBL ⁴³	2M compounds	bioactive drug-like small molecules			
ExCAPE-DB46	1M compounds	active/inactive molecules by target			
ZINC ⁴⁷	750M compounds	drug-like molecules, available for purchase			
PubChem ²⁶	111M compounds	mostly small molecules			
DrugBank ⁴²	13K drug entries	approved and experimental drugs			
GDB-17 ⁴	166B compounds	combinatorially generated molecules			
REAL database ⁴⁸	1.95B compounds	database of enumerated structures			
Tox21 ⁴⁹	11K compounds	toxicity data for various assays			
QM8 ⁵⁰	22K compounds	electronic spectra and excited state energy			
QM9 ⁵¹	134K compounds	geometric, energetic, electronic, thermodynamic			
PDBbind ⁵²	17K compounds	3D structures and binding affinity			
^{<i>a</i>} Number of available molecules reported as of October 2020.					

DEEP LEARNING MODELS FOR DE NOVO MOLECULAR DESIGN

Architectures. Recently, generative DL has emerged as a promising development for *de novo* molecular design, where deep neural networks are employed as generative models. This specific application has attracted considerable attention, with several novel architectures being proposed, that are briefly reviewed next, being also illustrated in Figure 2.

Recurrent Neural Networks. RNNs assume a sequential structure in the data, one where a sample is composed of a set of steps. This assumption is implemented by processing an input consecutively and introducing a connection carrying the output from previous steps into the current step. However, as the number of steps increases, RNNs can suffer from vanishing or exploding gradients during backpropagation, impairing the training process and making the learning of long-term dependencies extremely difficult. In practice, this is handled by using specialized units such as gated recurrent units (GRUs)⁵³ or Long Short-Term Memory (LSTM),⁵⁴ which introduce gates, learnable parameters controlling the flow of information through the steps.^{12,13}

Generative Adversarial Networks. GANs define a pair of networks, a generator, and a discriminator, trained in competition with each other. The generator is intended to transform random noise into real looking data and is trained to maximize the synthetic samples classified as real by the

discriminator. Meanwhile, the discriminator is trained to better discern between generated and real data. The training framework resembles a competition, with both networks constantly improving and adapting to each other.^{12,13,55}

Autoencoders. Autoencoders (AEs) are neural networks trained to copy their input into the output with restrictions imposed as to not simply learn the identity function. They are usually thought of as two separate parts, an encoder that transforms the input into a more compact latent state, and a decoder that reconstructs the input from this representation. Both are trained together to minimize the information lost from reconstructing.^{12,13}

Variational Autoencoders (VAEs) are a special type of AE, which assume that the data was sampled from an arbitrary statistical distribution. The encoder transforms its input into the parameters of a multidimensional statistical distribution, that is, a set of means and standard deviations. A sampling then occurs, where a point is drawn from the encoded distribution and fed into the decoder that reconstructs it into the original input. The objective function used for training consists of a term penalizing reconstruction errors and a term restricting the parameters encoded to be close to a normal distribution. This stochastic process acts to regularize the network while constraining the encoded parameters close to those of a normal distribution helps in forming a useful latent space.^{13,56}

Adversarial Autoencoders (AAEs) are an alternative to VAEs that employ adversarial training for structuring the latent space. In particular, the encoder transforms its input into a single point in the latent space. A discriminator network then attempts to discern between samples of a prior statistical distribution and encoded points. As such, the encoder can also be viewed as a generator engaged in a competition with the discriminator, ultimately balancing between the reconstruction and adversarial error.⁵⁷

Generating Molecules. There have been several approaches to applying generative DL to molecular generation, mainly differing on the chosen molecular representation. As such, usually more than one method surfaced for generating each of the main representations discussed in section.

Borrowing from the natural language processing field, molecules can be generated as sequences, such as SMILES, by using RNNs. Specifically, when using RNNs as a generative model, each token in the string is encoded as a one-hot vector and the network is trained to predict the next character in the sequence. The generation of new data is achieved by running the network autoregressively, that is, using its output as the input for the next time-step. This process is usually seeded with a special start token and the generation of a molecule ends when a special stop token is sampled. These two tokens are also respectively prefixed and appended to each molecule during training, Figure 3 illustrates the generative process.

Several research groups have employed this method with a stacked RNN, usually with LSTM cells, leading to good rates of validity, novelty, and diversity.^{18,35,58,59} More complex architectures such as VAEs and GANs have also been employed to generate molecules as strings; however, these also employ a RNN for the sequence generation process, either as the decoder or the generator.^{10,60,61}

Despite some limitations of sequence-based approaches, such as the need to learn a complex syntax and the mismatch between the edit distance of two SMILES and the underlying molecular similarity, these methods have produced impressive results

pubs.acs.org/jcim

Review



Figure 2. Top-left: Three layer Recurrent Neural Network (RNN) both rolled and unrolled. In each layer, the output of a step, besides flowing to the next layer, also flows to the next step of the layer itself. These recurrent connections are depicted in the unfolded view of the network as vertical arrows. Top-right: Variational Autoencoder (VAE) where the input is encoded to the parameters of a statistical distribution, namely, the means (μ) and standard deviation (σ). In practice, these correspond to two vectors which, on the sampling step, are interpreted as a set of means and standard deviations. Bottom-left: Generative Adversarial Network (GAN) composed by a generator and a discriminator. Training seeks not a minimum but a useful equilibrium between the generator and the discriminator. Bottom-right: Adversarial Autoencoder (AAE) where the attached discriminator must discern between encoded points and samples drawn from a prior statistical distribution.



Figure 3. Three layer RNN, unfolded over four time-steps. In autoregressive sequence generation, the process is started with a special start token, here "G". The model then predicts the next token, which is sampled and used as input for the next step. Generation ends when a stop token is predicted.

including some instances of experimental validation of the generated molecules. ^{17,62–64}

In an attempt to present molecules in a more natural and intuitive form, several methods have been proposed to directly generate molecular graphs. The generation process can be modeled as a sequence of decisions that progressively builds a graph. As Figure 4 describes, this approach usually employs multiple networks with specific functions, such as adding new nodes or adding edges between existing ones. In this paradigm, Li et al.⁶⁵ used two Graph Neural Networks (GNNs) to build graphs, one deciding whether to add a new node followed by another network deciding whether, and where, to add new edges

Review



Figure 4. Left: In sequential graph generation, a graph is built by evaluating a current partial graph, adding a node/edge and repeating until the network outputs a stop signal. Right: In the one-shot generation of graphs, probabilities over the full adjacency matrix and node/edge attribute tensors are produced. The graph is then obtained by taking a sample or the *argmax* of these outputs.



Figure 5. Left: General procedure for the generation of 3D shapes as proposed by Skalic et al.⁷² The convolutional decoder of a VAE is used to produce a 3D molecular shape which is converted to SMILES by a captioning network. Right: General process for generating molecules as 3D point sets, proposed by Gebauer et al.⁷³ It is conceptually similar to the sequential graph generation, operating on point sets with an internal coordinate system.

between the existing nodes. Liu et al.⁶⁶ used a similar generation procedure, employing Gated Graph Neural Networks (GGNNs) to build a VAE. The decoding process starts by using a linear classifier to add attributes to a fixed number of unconnected nodes. Edges between these nodes are then progressively added with two densely connected networks, with one deciding the target node and the other deciding the type of edge. At each step, the attributes of the nodes are updated with a GGNN and the process ends when a special stop node is selected. Furthermore, invalid operations can be masked during the generation allowing, for example, the enforcement of valency rules. More recently, Mercado et al.⁶⁷ compared six GNN architectures coupled with a tiered Multi Layer Perceptron (MLP) for sequentially building molecular graphs, reporting that GGNNs showed the best performance for both speed and quality of the generated structures.

An alternative is to generate graphs in a one-shot fashion by directly outputting an adjacency matrix and corresponding attribute tensors. De Cao and Kipf⁶⁸ applied this approach with MolGAN, a GAN whose generator outputs probabilities over the adjacency matrix and the annotation matrix of a molecular graph. Following a similar approach, Simonovsky and Komodakis⁶⁹ employed a VAE with a decoder that outputs three probability distributions, one over the adjacency matrix, one over the edge attribute tensor, and another over node attribute tensor. Furthermore, Ma et al.⁷⁰ proposed a regularization scheme capable of enforcing validity constraints

method		representation	architecture	ref
Transfer Learning		SMILES	Stacked RNN	18, 34, 59, 62, 63, 82
		SMILES		83
		Graph	GNN	84
		3D point sets	SchNet +2 MLP	73
Reinforcement Learning	Pretrain + RL	SMILES	Stacked RNN	18, 58, 82, 85
		SMILES	VAE	86
		Graph	two RNNs	87
	Adversarial + RL	SMILES	GAN	60, 88–90
		Graph	GNNs	91, 92
		Graph	GAN	68
Latent Space Navigation	Bayesian Optimization	SMILES	VAE	10, 93
		SMILES	AAE and VAE	94
		SMILES (production rules)	VAE	95, 96
		Graph (junction trees)	VAE	97
		Graph	VAE	98
	Gradient Ascent	Graph (junction trees)	VAE	97
		Graph	VAE	66, 99
	DL Model	SMILES	$GAN + AE^{32}$	36
		Graph (junction trees)	CycleGAN + VAE ⁹⁷	100
	GA	SMILES	AE	101
	PSO	SMILES	AE	102
	CLaSS	SMILES	VAE	103
Conditioned	Conditioned	SMILES	VAE	61
		SMILES	Stacked RNN	104
		SMILES	Two AAEs	105
		SMILES (production rules)	Two GANs	106
		SELFIES	VAE	107
		Graph	GNNs	65, 84
		Graph	VAE	69
		Graph (junction trees)	VAE	108
		3D shape	VAE + RNN	72
		3D shape	VAE + GAN	109
	semisupervised	SMILES	VAE	110
		SMILES	AAE	64
		Graph (scaffold extension)	VAE	111

Table 2. Methods for the Directed Generation of Molecules

to graphs generated in the same manner, significantly improving the validity of generated molecules.

Molecules are ultimately three-dimensional objects, with electron clouds surrounding their atoms and multiple possible spatial arrangements or stereoisomers. By generating molecules as sequences or graphs, important information is omitted, possibly hindering de novo design. Furthermore, determining the relevant conformations is not a trivial problem, as even small molecules can have many possible conformations.⁷¹ As such, some attempts have been made toward generating molecules as three-dimensional entities. Skalic et al.⁷² proposed to generate voxelized molecular shapes with a VAE and then caption them into SMILES with a separate network. A different approach was proposed by Gebauer et al.⁷³ where molecules are generated as point sets, iteratively built based on the pairwise distance to previously placed points/atoms. Figure 5 outlines the general process of these two last approaches. More recently, Ragoza et al.⁷⁴ proposed a method to generate molecules as 3D atomic density grids and then apply an optimization algorithm to find the best fitting 3D chemical structure.

Evaluating Generative Models. Given the rapid growth this field is experiencing, the development of systematic and robust methods to assess the performance of novel approaches is essential to help guide future work. Aiming to help address this,

Preuer et al.⁷⁵ introduced the Fréchet ChemNet Distance (FCD), a metric for comparing the generated molecules against the training data set. This score measures the distance between the hidden representations of the two sets of molecules in 'ChemNet', a recent multitask network for predicting biological activities. As this network was trained to predict the bioactivities of about 6000 assays, the team proposes that FCD combines into a single metric a multitude of important molecular features, which is therefore useful to evaluate generative models.

Arús-Pous et al.⁷⁶ proposed a method to evaluate how well a generative model learns to cover the relevant chemical space. According to the team, this can be accomplished by training the model on a fraction of a large enumerated data set, such as GDB-13, and then tracking the percentage of the total data set the model can recover, how uniform the coverage is, and also whether it generates molecules outside the data set. These results can then be compared to an ideal model, directly sampling the data set, that serves as an upper bound for performance. Furthermore, the team introduced a method for evaluating the quality of the training process by comparing the negative log-likelihood of the sampled, training and evaluation sets throughout the training process.

Brown et al.⁷⁷ introduced GuacaMol, a framework for benchmarking models for *de novo* molecular design. This

pubs.acs.org/jcim

Review



Figure 6. In transfer learning, a general model is first trained on a large data set and then fine-tuned toward generating the desired properties with a smaller, focused, data set.

framework was divided into two main sets of benchmarks, distribution-learning and goal-directed generation, aiming to emulate the two main use cases of these models. The first set measures how well the models learn to generate new molecules, using validity, uniqueness and novelty rates, and also whether they match the properties of the training data set, measuring the FCD and the divergence in the distribution of a variety of physicochemical descriptors. The second set evaluates the targeted generation performance, including benchmarks such as optimizing given molecular features or properties, generating molecules similar to a target compound or recover a specific target molecules. In addition, the team also included a benchmark for assessing molecular quality, leveraging rule sets for building high-throughput screening libraries. Lastly, a standardized data set alongside a number of baselines were also released with this framework to help compare novel approaches. Polykovskiy et al.⁷⁸ proposed the benchmarking framework MOSES, for evaluating the distribution learning performance of generative models. To assess whether the models can generate new molecules, it measures the validity, uniqueness and novelty rates along with the internal diversity and the fraction of generated molecules that pass a set of structural filters for molecular quality. The framework also provides a set of metrics meant to evaluate how well the model learned features of the training data set. For this purpose, it presents the FCD, the distance between the distribution of various physicochemical properties, the Tanimoto Similarity to a nearest neighbor, the cosine similarity of Bemis-Murcko scaffolds, and also the cosine similarity of BRICS fragments. These last two metrics help compare molecules at a substructure level, while the first three help to capture more abstract chemical and biological similarities. The team also released various useful baselines, as well as a standardized data set with a recommended train, test and scaffold test split.

Building on these early works, other approaches to evaluating molecular generative models have continued to be developed. Specifically, Renz et al.⁷⁹ highlighted some shortcomings of currently used evaluation metrics. Specifically, the team details how a trivial model can excel in distribution-learning benchmarks and also how goal-directed generation can exploit biases in the scoring functions, producing compounds with high scores but of little practical use. Cieplinski et al.⁸⁰ proposed to better represent real discovery problems by using docking as a benchmark of the different methods of goal-directed generation.

Zhang et al.⁸¹ improved on their earlier work, measuring the coverage of chemical space by generative models,⁷⁶ by also evaluating the coverage of functional groups and ring systems. Furthermore, the team provided results for various, recently introduced, generative model architectures allowing for their comparison as well as providing useful baselines for future works.

GENERATING COMPOUNDS OF INTEREST

The automatic generation of novel molecules usually targets specific properties and characteristics, such as solubility or bioactivity. As such, the ability to create not just new, but also focused, molecules is of interest. Table 2 summarizes state-ofthe-art approaches for targeted compound design, further described in this section.

Screening. Testing large numbers of compounds to see if they show evidence of having desired properties is usually one of the first steps in the drug discovery pipeline. This usually entails a virtual screening for bioactivity either based on a target, as in docking, or on known ligands, as in ML classifiers and similarity searching.¹¹² This process can be used on its own or after other methods of biasing the generation process. Docking refers to a process of fitting a molecule to the binding site of a given target whose 3D structure is known. This is usually attained by scoring different poses (spatial orientations) of a molecule relative to its target. The score is calculated by a scoring function, usually hinged on predictive changes in Gibbs free energy.¹¹³

Yuan et al.,¹⁷ for example, docked generated molecules against VEGFR-2, a mediator of the VEGF angiogenesis pathway, to choose compounds to be synthesized. Similarly,

Polykovskiy et al.⁶⁴ docked molecules against Janus kinase 2 and 3 as part of the selection process for synthesis.

An alternative to predict how well a molecule may bind to a receptor is through the use of supervised machine learning classifiers trained to distinguish between known actives and not actives. An example of this approach is the work done by Olivecrona et al.⁵⁸ where a support vector machine was trained to predict activity toward the Dopamine type 2 Receptor (DRD2).

In a similarity search, the similarity between a molecule and a set of molecules that are known to be active is determined. A common approach is to compute the Tanimoto coefficient between the fingerprints of each molecule. Kadurin et al.¹¹⁴ used



Review



Figure 7. Top: The model is first pretrained through maximum likelihood estimation, learning the structure of the output space along with general chemical rules. Then, using RL, the model is optimized for specific properties such as binding affinity or solubility. While similar in concept to transfer learning, the use of RL allows one to bias the model toward a wider range of objectives. Bottom: Directed generation with RL and GAN. This method leverages adversarial training to produce feasible molecules and RL to bias the generation toward desired properties.

this method to search PubChem for molecules similar to the fingerprints generated by their model.

Transfer Learning. Despite that a simple screening of unspecific model outputs may lead to finding molecules of interest, the process is somewhat inefficient as a large number of the generated molecules end up being discarded. A less wasteful approach would be to first bias the model toward producing (more) molecules meeting the desired properties. This can be achieved with transfer learning, a training procedure where a model first learns to perform a similar task, but for which larger data sets exist, later being fine-tuned on the intended data.¹ This approach assumes that several of the underlying attributes learned on the first set are transferable to the second set. In molecule generation, this is usually applied on sequence-based approaches where a large data set such as ZINC or ChEMBL first helps to learn the syntax of the string representation and then a smaller, targeted, data set biases the model toward particular attributes, such as a given biological activity, as it is illustrated by Figure 6.

Transfer learning has been successfully applied to fine-tune stacked RNNs generating SMILES. In 2017, Segler et al.¹⁸ applied this method to a RNN with three LSTM layers generating SMILES. The model was later fine-tuned on known ligands of specific receptors and reported to successfully recover molecules from a hold-out test set.

Merk et al.^{62,63} employed the same method, where their finetuned SMILES-based RNN was used to generate compounds to be later synthesized. The team then performed *in vitro* activity testing, reporting that four out of five, and in a later work two out of four, were active. Gupta et al.⁵⁹ experimented with fine-tuning on small data sets, reporting that even just a set of five molecules can lead to a model capable of generating unseen actives. Moreover, Moret et al.³⁴ also looked into the applicability of transfer learning in low data regimes when combined with data augmentation. With just five dissimilar natural products, they were able to generate structurally diverse molecules covering a broad range of scaffolds. Departing from sequence-based representations, Gebauer et al.⁷³ used transfer learning with



Figure 8. Here, the latent space of an AE is used as a reversible and continuous molecular representation allowing for the application of various optimization algorithms.

their point set based model to target a specific value range of HOMO-LUMO gap, a molecular property relevant for the development of organic semiconductors.

Transfer learning can also be used as part of a larger procedure, as a mean to accelerate training and improve results. Notably, Li et al.⁸⁴ fine-tuned their model before generating molecules conditioned for a specific bioactivity profile. In a similar vein, Blaschke et al.⁸² used transfer learning to focus the model toward features relevant to their objective, facilitating their subsequent reinforcement learning procedure.

Reinforcement Learning. A different approach to bias models toward generating molecules of interest is reinforcement learning (RL). RL outlines a framework where an agent, or system, interacts with an environment through a sequence of actions that are dictated by a policy and evaluated by a reward signal. The agent must iteratively revise the policy to improve the cumulative rewards over the full sequence of adopting the best set of actions in a given environment.^{115,116} In *de novo* generation of molecules, RL has been applied in both sequence and graph-based approaches.

One application is to first pretrain a model through maximum likelihood estimation and then optimize it in a RL framework toward generating molecules with desired properties. This concept is presented in Figure 7 (top). In this vein, Segler et al.¹⁸ biased their stacked RNN by coupling it to a prediction model and iteratively fine-tuning on the generated active compounds. With just eight iterations, the team reported the successful recovery of active compounds from the test set. Olivecrona et al.⁵⁸ trained a stacked RNN to generate molecules as SMILES and, using RL, optimized it toward generating analogues to celocoxib and generating molecules predicted as active against DRD2. Popova et al.⁸⁷ applied this concept to the sequential generation of graphs. Their model, termed MolecularRNN, was first trained to generate diverse realistic samples and then optimized for either quantitative estimate of drug-likeness (QED), melting point, and octanol-water partition coefficient (log *P*) penalized by synthetic accessibility (SA) and large rings.

Blaschke et al.⁸² developed a production ready generative method, termed REINVENT2.0, based on a stacked RNN leveraging randomized SMILES and reinforcement learning. Later, Blaschke et al.⁸⁵ proposed a method to improve diversity in the REINVENT framework, termed memory-assisted RL. This method creates "buckets" grouping similar generated molecules, once a bucket reaches a set capacity, subsequent molecules falling in that cluster are penalized. This memory unit, therefore, helps lead the model to unexplored areas of chemical space. The framework was employed to target a specific range of log P and optimize the predicted activity for HTR1A and for DRD2. The team noted an increase in the generation of diverse scaffolds, while producing highly scored compounds.

Zhavoronkov et al.⁸⁶ used RL to optimize a SMILES based VAE toward generating selective DDR1 kinase inhibitors. The objective was based on the predictions of an ensemble of three Self-Organizing Maps (SOMs) predicting general activity toward kinases, selectivity for DDR1, and novelty of generated molecules. From the generated compounds, six were selected for synthesis and in vitro testing with two of those being reported as both active and stable. Further in vivo testing was also performed on one molecule, with reasonable pharmacokinetic properties being reported. Lastly, the authors also noted the substantial reduction in both time and costs of their DL based approach compared to traditional drug development pipelines.

GANs and RL can also be combined to generate realistic, but optimized, molecules. Figure 7 (bottom) outlines this method. More specifically, a GAN is trained in a RL framework combining the adversarial reward with other relevant objectives. This method was employed by Guimaraes et al.⁶⁰ in ORGAN to generate SMILES optimizing molecular properties such as log *P*, SA, and QED. Improving on the previous method, Sanchez-Lengeling et al.⁸⁸ proposed ORGANIC, optimizing for melting point, drug-likeness with QED, and Lipinski's rule-of-five and finally for nonfullerene electron acceptors for use in organic solar cells. Putin et al.⁸⁹ proposed ATNC, improving ORGANIC with a differentiable neural computer as generator and a novel reward function to improve the diversity of generated structures. With this model, the team optimized for similarity to known kinase inhibitors and synthesized a molecule similar to a generated one.

With a similar method but departing from sequence-based generation, You et al.⁹¹ proposed GCPN to sequentially generate molecular graphs with optimized properties. Reporting that it can be used to target specific ranges of log *P* and molecular weight and also optimize log *P* penalized by SA and large rings, while constrained by similarity to a starting molecule. Karimi et al.⁹² employed a similar molecular generative process to generate new drug combinations. Specifically, the proposed method aimed to directly generate sets of novel molecules that could be useful as disease-specific drug combinations. To this

end, the team employed a RL process to sequentially generate sets of molecular graphs guided by a chemical validity reward, an adversarial reward enforcing SA and drug-likeness, and a network-based reward to help target the desired disease by incorporating prior knowledge from gene–gene, gene-disease, and disease–disease networks.

Also combining GANs and RL, De Cao and Kipf⁶⁸ proposed the generation of molecular graphs in a one-shot fashion in their model MolGAN.

Exploration and Exploitation of Molecules Latent Space. Instead of optimizing models for the desired properties, models based on the AEs architecture provide a latent representation of molecules that can be used for property optimization or targeted generation of compounds. These methods, which are illustrated in Figure 8, make use of well structured latent spaces, for which VAE and AAE are common choices. Different approaches have been suggested to navigate and shape the latent space of models leveraging both sequence and graph-based representations.

Bayesian Optimization. Bayesian Optimization (BO) is a sequential model-based optimization method suitable for blackbox problems. It has two main elements, a probabilistic surrogate model and an acquisition function. The surrogate serves to estimate the objective function given some currently known data, then the acquisition function leverages the model to determine the best point in the objective function to evaluate. These new data are then used to update the surrogate model and the process repeats for a set of iterations, ideally leading to the global maximum of the objective function. Common choices are a Gaussian Process for the surrogate and the Expected Improvement for the acquisition function.

In the *de novo* design, BO is used to optimize the properties of molecules by operating on their latent representation, using the decoder to reconstruct molecules from the suggested points. In this application, approximate inference is often used, in the form of a sparse Gaussian Process, due to the large number of evaluations that are made.

BO has been often used to demonstrate that the latent space of a particular architecture can be effectively navigated. In the context of molecular generation, it was first suggested by Gomez-Bombarelli et al.¹⁰ who, in an earlier version of their work, optimized the log P of molecules penalized by their SA and presence of large rings. This methodology, and objective function, was then adopted in subsequent approaches by Kusner et al.⁹⁵ with GrammarVAE and Dai et al.⁹⁶ with SD-VAE which leveraged SMILES production rules and by Jin et al.⁹⁷ in JT-VAE and Samanta et al.98 in NEVAE which dealt with molecular graphs. A more practical objective was employed by Blaschke et al.⁹⁴ who optimized the predicted DRD2 activity, reporting that BO was capable of effectively navigating the latent space of their uniform AAE and find novel active molecules. Lastly, Griffiths and Herńandez-Lobato93 applied constrained BO as a way to mitigate training set mismatch, where the BO would visit latent points far from the training data that the model struggles to reconstruct. More specifically, the acquisition function was altered to only consider latent points which decoded to valid molecules with a tangible molecular weight. This was reported to lead to the generation of higher quality molecules using three drug-likeness metrics.

Genetic Algorithms and Particle Swarms. Once a continuous latent space is obtained, other optimization algorithms can be employed to optimize for desired properties. For example, Sattarov et al.¹⁰¹ used a Genetic Algorithm (GA)

pubs.acs.org/jcim

to explore the latent space of their SMILES based *seq2seq* AE. Setting as goal optimizing molecules for activity toward the adenosine A2A receptor, they reported the generation of libraries enriched with actives and novel scaffolds. Winter et al.¹⁰² also explored the latent space of a SMILES based *seq2seq* AE using a different meta-heuristic, Particle Swarm Optimization (PSO). They experimented with optimizing different properties, such as QED, penalized log *P*, activity toward EGFR, and activity toward BACE1. Furthermore, multiobjective optimization was also attempted by minimizing and maximizing activity for each of the receptors, reporting molecules with the desired activity profile and favorable absorption, distribution, metabolism, excretion, and toxicity properties.

Gradient-Based Methods. Gradient-based methods are alternative optimization algorithms that additionally require the objective to be differentiable. Although fulfilling the differentiability requirement might not always be possible, training a secondary neural network to predict the desired properties, in parallel with the main model, allows to obtain the gradient of a latent encoding with regard to the so desired property. This process has been used to optimize chemical properties through gradient ascent in some approaches, mainly as a benchmark for the smoothness of latent space.

Jin et al.⁹⁷ optimized the penalized log *P* constrained to a set degree of similarity to the starting molecule. More specifically, the team used a feed-forward network as a predictor and constrained the optimization with the Tanimoto similarity to the original molecule. Liu et al.⁶⁶ employed a similar methodology, optimizing however for the QED based on the gradients of a gated regression network. Bresson and Laurent⁹⁹ used gradient ascent with a single multilayer perceptron to, and following the work of Jin et al.⁹⁷ optimize the penalized log *P* and the log *P* constrained by similarity.

Deep Learning for Latent Space Navigation. Deep Learning models can be trained to operate on molecular encodings of a separate generative model. Prykhodko et al.³⁶ applied such an approach to their LatentGAN, a GAN trained to generate latent points of a separate SMILES heteroencoder. The real data used for training is obtained by passing SMILES through the encoder of the AE, while during the generation the novel latent points are converted into molecules using the decoder part of the AE. By training their model to produce realist encodings of compounds with activity for either EGFR, HTR1A, and S1PR1 (separate model for each), they reported the generation of valid and novel SMILES with a large percentage predicted as active. In a similar vein, Maziarka et al.¹⁰⁰ proposed mol-cycleGAN, a cycleGAN operating on the latent space of the JT-VAE⁹⁷ to optimize molecules, while keeping the results similar to the starting compound. Specifically, the model learns to convert molecules (points in latent space) from one set into another and back again. For example, it can convert from a set with only three aromatic rings into a set with only two, all while keeping transformed points similar to the original. The team reported that the model was capable of removing halogen moieties, replace bioesters, alter the number of rings, and increase the predicted activity toward the DRD2.

More recently, Chenthamarakshan et al.¹⁰³ applied a VAE generating molecules as SMILES and achieved controlled generation with Conditional Latent Attribute Space Sampling (CLaSS).¹¹⁹ Under CLaSS, a Gaussian mixture model is trained to match the posterior of the trained encoder and a binary classifier trained for predicting desired properties from latent encodings.

pubs.acs.org/jcim



Figure 9. Top: In conditioned generation, the desired properties are introduced as explicit inputs to the model. These properties are precomputed for each compound of the training set and used during training to induce a correlation between the two. This correlation is then leveraged during the generation process to target specific property values. Bottom: In the semisupervised case of conditioned generation, only part of the training set has the desired properties available. To overcome this, a predictor network is trained on the labeled instances and used to predict the properties of unlabeled ones.

Then, samples are drawn from the mixture model and filtered using the latent classifiers with only those predicted to have the specified attributes being decoded back to sequences. This method was employed to generate possible leads targeting SARS-CoV-2, using latent attribute predictors for three molecular properties and the binding affinity to relevant protein targets.

Conditioned and Semisupervised Generation. An alternative to both biasing the model and latent space optimization is to include explicit inputs to the model for controlling the properties of the generated molecules. As illustrated in Figure 9 (top) with an AE, this is achieved by introducing a condition vector to the models' input, effectively biasing/conditioning the generation process toward the specified values. During training, the condition vector corresponds to various properties of the encoded molecule, leading the model to infer a correlation between the two. Later, during sampling, this vector can be altered, controlling the properties of the generated molecules.

Li et al.⁸⁴ used a sequential graph generator based on GNNs and added a conditioning vector at each step of the graph building process. Their model was conditioned on molecular scaffolds, QED, SA and, after fine-tuning, the predicted activity toward JN3 and GSK3 β . The team was able to successfully generate inhibitors for either receptor, as well as dual inhibitors. A similar process was employed by Li et al.⁶⁵ to condition a GNN for sequential graph generation by appending the condition vector solely to the initial node states. Conditioning on the number of atoms, bonds and aromatic rings, the model was able to extrapolate and successfully generate molecules when conditioned with values outside the training data.

Kotsias et al.¹⁰⁴ conditioned a SMILES based stacked RNN by setting the conditioning vector as the initial internal state of the network. Two different approaches were compared, either conditioning with molecular fingerprints or building the condition vector with molecular properties such as Topological Polar Surface Area (TPSA), molecular weight, and bioactivity. They reported that while both approaches could generate molecules satisfying the desired properties, the fingerprint-based model generated molecules with scaffolds similar to the seed compound, facilitating the encoding of structural restrictions. Meanwhile, the property-based model generated more dissimilar scaffolds, enabling a more versatile exploration of chemical space. The conditioned generation methodology is also applicable to AEs, where the condition vector is generally appended to the input of both the encoder and decoder. For example,

Simonovsky and Komodakis⁶⁹ demonstrated conditioned generation with their GraphVAE by controlling the number of heavy atoms in generated molecules. Lim et al.⁶¹ used a conditioned VAE to control molecular weight, TPSA and number of H^+ donors and acceptors, reporting independent control of properties, as well as generating molecules with properties beyond those seen during training. Working with a 3D representation of molecules, Skalic et al.⁷² proposed to condition their shape-based VAE with the location of pharmacophores. Specifically, the decoder received a 3D shape constructed by placing property points close to atoms with that property. The team noted that conditioning the reconstruction of random latent points often led to implausible output shapes, but conditioning the decoding of seed molecules improved the reconstruction of pharmacophore features.

When the desired properties are not readily determinable and no large labeled data sets are available, common with bioactivity data, semisupervised-AE can be used, Figure 9 (bottom). A semisupervised-AE consists of an AE with an added predictor network, which receives the molecule as input and outputs its properties. These are then appended to both the input and output of the encoder, ultimately conditioning the decoder. The architecture is termed semisupervised because the data set does not need to be fully labeled. Labeled instances are used for training the predictor, replacing its output, while unlabeled samples have their properties predicted by the predictor network.¹²⁰

Kang and Cho¹¹⁰ employed a semisupervised-VAE conditioned on molecular weight, log *P*, and QED and experimented with various fractions of labeled/unlabeled data used for training. Polykovskiy et al.⁶⁴ evaluated the application of different disentanglement techniques to a semisupervised-AAE. With their most successful method, termed semisupervised entangled AAE, the team was able to generate molecules conditioned on the activity toward the Janus kinase 2 and Janus kinase 3. By setting low activity for JK2 but high activity for JK3, they generated a set of selective inhibitors that were then filtered. A single molecule was synthesized and reported to have *in vitro* activity and selectivity for the Janus kinase 3.

Lim et al.¹¹¹ used a conditioned VAE to extend molecular scaffolds, producing molecules with predetermined scaffolds and desired properties. The model was reported to successfully condition the molecular weight, TPSA and log *P*. Furthermore, a semisupervised extension of the model was used to design EGFR inhibitors, reporting a significant enhancement of the predicted inhibition potency.

More recently, Mendez-Lucio et al.¹⁰⁶ used a VAE in conjunction with a GAN conditioned on gene expression data. More specifically, a two-stage GAN is trained to generate latent points of a GrammarVAE,⁹⁵ with its decoder serving to reconstruct the points generated by the GAN. Here, the conditioning vector is an input to the generator, alongside Gaussian noise. Also, two networks, one per stage, predict whether the generated latent points correspond to the gene expression profiles used for conditioning the generator. Using this method, the team conditioned the model on the gene expression of ten knockouts of pharmacological interest, reporting the generation of molecules similar to known active compounds. Born et al.¹⁰⁷ proposed PaccMannRL, a framework that leverages gene expression data and combines reinforcement learning with conditioned VAEs. This method starts by training two separate VAEs, one to reconstruct molecules, represented as SMILES, and the other to reconstruct gene expression data. The two models are then combined, with the output of both encoders being summed together and used as input to the molecular decoder. This new architecture is then trained through reinforcement learning toward generating molecules targeting the specified gene expression profile. The framework was then employed to generate anticancer compounds, with the team reporting an improvement in predicted efficacy while maintaining similar validity scores.

Also exploiting omics data, Shayakhmetov et al.¹⁰⁵ proposed a SMILES based conditioned AAE to generate molecules capable of inducing a desired transcriptomic change. Particularly, the model is composed of two AAEs, one tasked with reconstructing molecules and the other with reconstructing gene expression profiles, with the conditioning vector being produced by the gene expression encoder as a separate latent vector. As such, this architecture produces a three-part latent space, with one part meant to encode molecule specific information, another one to encode expression specific information, and the last to encode features relevant to both. This specific arrangement was designed to aid the model in ignoring nonrelevant cellular processes that are included in the gene expression profile from the changes induced by the molecule.

Masuda et al.¹⁰⁹ proposed to condition their previously proposed model, generating molecules as 3D structures, with the 3D structure of the intended binding site. Specifically, a separate encoder module was added to encode the binding target structure into a latent representation, which was then concatenated to the output of the molecular encoder, conditioning it. With this approach, the team reported that, by sampling around a seed molecule, they often could generate new compounds with better binding affinities.

A different approach was taken by Jin et al.¹⁰⁸ who adapted JT-VAE to optimize input molecules by adding adversarial training and a conditioning vector to the latent encoding. That is, the model was trained to "translate" molecules missing the desired properties into molecules with those qualities. The adversarial objective ensures that the new molecule has the desired properties, while the conditioning vector directs the generation process. With this method, the team reported success when optimizing for the penalized log *P*, QED and the predicted activity toward DRD2 while constraining by the similarity to the initial molecule.

Synthetic Accessibility. Successfully applying these methods to practical use cases will inevitably depend on synthesizing the novel compounds. However, ensuring the SA of the generated molecules is a major hurdle that often goes unnoticed.

Seeking to help address this, Gao and Coley¹²¹ discussed and compared three approaches capable of guiding generative models toward synthesizable compounds. Specifically, they considered the use of synthesizability scores and retro-synthetic analysis to filter generated molecules, filter the training data set or to modify the objective function used for targeted generation. The team reported that, despite useful, the first two methods often proved insufficient. Furthermore, modifying the objective improved synthesizability, but at the cost of the main goals.

Horwood and Noutahi¹²² directly addressed this issue in their RL based method by iteratively building novel molecules

Review

Tabl	le 3.	Experimental	Validation	of Mo	olecules	Generated	with	Generative 1	DL	,
------	-------	--------------	------------	-------	----------	-----------	------	--------------	----	---

		activity				
target	directed generation	in silico	in vitro	in vivo	ref	
RXR, PPAR	transfer learning	SPiDER	5 synthesized; reported: 4 active		63	
RXR	transfer learning	SPiDER WHALES	4 synthesized; reported: 2 active		62	
JK3 selective	reinforcement learning	docking	1 synthesized; reported: active and selective for JK3		64	
kinase inhibitors	reinforcement learning		50 purchased (similar); reported: 7 active		89	
DRD2, 5-HT1A, 5- HT2A	transfer learning	MT-DNN on ECFP4	1+6 analogues synthesized; reported: active for the 3 receptors	1+6 analogues; 1 active and acceptable safety	127	
VEGFR-2	train on actives	docking	5 synthesized; reported: 3 active and noncytotoxic		17	
DDR1	reinforcement learning	SOM pharmacophore	6 synthesized; reported: 2 active and stable	1 tested; half-life 3.5 h; 10 analogues tested	86	
p300/CBP inhibitors	transfer learning	docking	1+26 analogues synthesized; reported: active, selective, and stable	good bioavailability, efficacy, safety	128	
LXR agonists	transfer learning		25 synthesized, 3 purchased; reported: 12 active		129	

through a series of chemical reactions. With this approach, the team reported the successful optimization of multiple objectives, while maintaining good diversity among generated molecules and also providing a valid synthetic route for every novel molecule.

A similar approach was taken by Gottipati et al.,¹²³ which also applied RL to generate molecules using a sequence of chemical reactions. Like the previous method, the team reported attaining high scoring generated compounds while ensuring SA.

Bradshaw et al.¹²⁴ also proposed to generate novel molecules by recursively combining simple building blocks through a series of chemical reactions. The team proposed two variants of their approach, one where RL is employed to perform targeted generation and, departing from the two previous works, a Wasserstein Autoencoder (WAE) based model. The later one, as discussed in the previous sections, should allow other methods of targeted generation to be used.

CURRENT APPLICATIONS

There are multiple proposed approaches to not only generate molecules, but also do so in a directed manner, controlling and optimizing for desired properties. The adopted objectives have more often been employed as a benchmark of the proposed methodologies than as goals in themselves. While some have been of limited real use, like maximizing log P penalized by SA and large rings, others such as optimizing for specific bioactivity profiles can have a more direct application in fields like drug discovery. For example, the swiftness of these methods has recently found use in the creation of therapeutic leads for SARS-CoV-2.

Drug Development. The large costs associated with drug development has often led to the implementation of various computational tools to assist and accelerate the process. As such, a large focus has been placed on applying deep generative models to various stages of early drug development. The most commonly suggested application has been to the *de novo* drug design. Indeed, the bulk of methods here described are capable of generating broad molecular libraries, with a large part capable of focused generation. These libraries can then be used for virtual screening or high-throughput screening, hopefully exploring previously unseen regions of the chemical space.

A different suggested application has been for molecule optimization. Although the usefulness of some objectives has been questioned, the ability to, for example, optimize solubility or improve SA while maintaining high similarity to an original structure can help inform lead optimization. Furthermore, easily and reliably finding close novel analogues to a molecule or introducing specific modifications like substituting bioesters can also be of practical use. Lastly, some attention has also been given to fragment-based drug development, with methods proposed for growing a molecule from a specified fragment⁵⁹ or linking two fragments together.^{125,126}

Several different biological targets have also deserved attention. Table 3 summarizes these applications. It notes the various instances of experimental validation of *de novo* generated molecules that have already been performed. These are mostly successful *in vitro* activity testing, with two instances of *in vivo* validation. These successful practical realizations encourage further research into this blooming field.

COVID-19. With the recent SARS-CoV-2 pandemic, generative DL methods became an attractive option for the *de novo* design of possible therapeutic leads. Indeed, in the span of a few months, several approaches were proposed and a large number of potential compounds were shared.

Bung et al.⁸³ biased a SMILES based stacked RNN with transfer learning toward generating possible binders of SARS-CoV-2 proteases. Specifically, a set of 1.6 million molecules from ChEMBL were used for pretraining and then around 2500 protease inhibitors were used for fine-tuning the model. Furthermore, reinforcement learning was used to control other molecular properties such as QED, $\log P$, molecular weight and SA. After screening the generated molecules using docking simulations, the team shared 31 potential leads.

Shaker et al.¹³⁰ used their SMILES-based in-house model, named Rosalind, to generate molecules targeting the SARS-CoV-2 main protease M^{pro}. After applying filters for binding affinity predicted by docking, QED, molecular weight, structural alerts and predicted toxicity the team shared a list of 40 compounds.

Chenthamarakshan et al.,¹⁰³ as described earlier, employed CLaSS on a SMILES based VAE to generate compounds with favorable binding to three relevant target proteins of SARS-CoV-2. The controlled sampling leveraged property predictors for QED, SA, log P, and binding affinity to a specified protein. Filters were then applied for toxicity, retrosynthesis prediction and binding affinity to the relevant target with docking and a set of 3.5K potential leads was shared by the team.

Zhavoronkov et al.⁸⁶ used an internal pipeline leveraging 28 different models and various molecular representations to generate molecules targeting the SARS-CoV-2 main protease M^{pro} . Docking was employed to rank the compounds by their affinity and a set of 10 molecules was shared by the team.

Born et al.¹³¹ adapted their previously proposed framework, PaccMannRL, to generate compounds addressing 41 targets of SARS-CoV-2. Specifically, the molecular VAE was adapted to generate SELFIES (instead of SMILES) and the conditioning was performed based on a protein VAE (instead of gene expression data).

Organic Photovoltaics. The design of new organic photovoltaics has great potential to help reduce the dependence on fossil fuels by enabling cheaper and more efficient solar energy.¹³² Deep generative methods can be employed to quickly design new molecules targeting desired properties such as a specific HOMO–LUMO gap and high Power Conversion Efficiency (PCE). Indeed, some attention has already been devoted to this application.

For instance, Sanchez-Lengeling et al.⁸⁸ applied ORGANIC toward generating nonfullerene electron acceptors for use in organic solar panels, reporting an increase in the average predicted PCE. Jørgensen et al.¹³³ used a GrammarVAE⁹⁵ for generating donoracceptor polymers optimized for a specific range of optical gap and LUMO energy. Griffiths and Hernández-Lobato⁹³ employed their proposed constrained BO method toward generating molecules optimized for PCE, reporting that the averaged score of the generated molecules lied above the 90th percentile of the training data. Gebauer et al.⁷³ leveraged transfer learning to fine-tune their model toward generating molecules targeting a specific HOMO–LUMO gap. Similarly, Yuan et al.¹³⁴ biased a RNN with transfer learning to generate donor–acceptor oligomers targeting a specific HOMO–LUMO gap.

■ FUTURE DIRECTIONS OF RESEARCH

Over the last years, the field of deep generative learning for molecular design has seen an explosion of interest, with a great number of different and novel approaches being proposed. Within this blooming field, some trends emerged that have somewhat guided novel research and give clues for the direction of future developments.

Many of the early approaches to generating new molecules with DL borrowed from the NLP field, using RNNs for modeling and generating molecules as sequences. However, in the meantime, the state-of-the-art in language processing has evolved to leverage attention mechanisms within architectures like the transformer, BERT and GPT-3.^{135–137} Indeed, these advances have begun to trickle into the field of generative molecular design with recent works, leveraging the transformer architecture to approach the generation of new bioactive compounds as a translation from the amino acid sequence of a target protein to an active SMILES,¹³⁸ to link fragments together¹²⁶ or to perform scaffold hopping.¹³⁹ Likewise, approaches leveraging these new methods, and others that may emerge from NLP, are likely to receive further attention in the future.

With regard to molecular representations, while early methods dealt mainly with SMILES,^{18,58} several subsequent works sought a more meaningful representation by directly generating molecular graphs.^{87,91} The pursuit of this goal has led to the proposal of novel graph generation procedures and will likely inspire interesting new research.

Nevertheless, graph-based representations still disregard the three-dimensional nature of molecules, possibly missing useful information by, for example, neglecting chirality. As such, recent interest has formed around the generation of 3D molecular structures leading to the proposal of a few methods^{72–74} and, hopefully, motivating future works.

An interesting recent trend has been to leverage omics data to help direct the generation process.^{105–107} Due to the large quantities of omics data already available and the information that it can convey, this strategy has the potential to be extremely useful. A handful of approaches have already been suggested and further work along these lines will likely be developed.

The synthetic accessibility of generated compounds, which is vital for the practical realization of these methods, still poses a significant hurdle to overcome. Some methods have recently emerged taking this into consideration,^{121–124} while it is expected that future approaches will continue to address this issue.

With regard to new applications, methods targeting the biotechnological industry, such as novel artificial flavorings, dyes, catalysts, or pesticides, could be very relevant. Furthermore, employing methods leveraging omics data to aid in metabolic engineering tasks might also be a promising avenue for future work.

Lastly, some evidence has been reported in favor of using multiple generative models in parallel to cover different regions of the chemical space.³⁶ This further motivates new research and development in this area, as different architectures and methods could complement each other, learning specific chemical patterns and enabling a more diverse approach to the exploration of chemical space.

AUTHOR INFORMATION

Corresponding Author

Miguel Rocha — Centre of Biological Engineering, Campus Gualtar, University of Minho, 4710-057 Braga, Portugal; orcid.org/0000-0001-8439-8172; Email: mrocha@ di.uminho.pt

Authors

- Tiago Sousa Centre of Biological Engineering, Campus Gualtar, University of Minho, 4710-057 Braga, Portugal; orcid.org/0000-0003-4013-7012
- João Correia Centre of Biological Engineering, Campus Gualtar, University of Minho, 4710-057 Braga, Portugal
- Vitor Pereira Centre of Biological Engineering, Campus Gualtar, University of Minho, 4710-057 Braga, Portugal

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c01496

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme (Grant Agreement Number 814408).

REFERENCES

(1) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.- Aided Mol. Des.* **2013**, *27*, 675.

pubs.acs.org/jcim

(2) Schneider, G. Automating drug discovery. *Nat. Rev. Drug Discovery* **2018**, *17*, 97–113.

(3) DiMasi, J. A.; Grabowski, H. G.; Hansen, R. W. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics* **2016**, *47*, 20–33.

(4) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. J. Chem. Inf. Model. **2012**, *52*, 2864–2875.

(5) Walters, W. P. Virtual Chemical Libraries: Miniperspective. J. Med. Chem. 2019, 62, 1116–1124.

(6) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: reactiondriven de novo design of bioactive com- pounds. *PLoS Comput. Biol.* **2012**, *8*, No. e1002380.

(7) Spiegel, J.; Durrant, J. AutoGrow4: An open-source genetic algorithm for de novo drug design and lead optimization. *J. Cheminf.* **2020**, *12*, 25.

(8) Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **2019**, *10*, 3567–3572.

(9) Yoshikawa, N.; Terayama, K.; Sumita, M.; Homma, T.; Oono, K.; Tsuda, K. Population-based De Novo Molecule Generation, Using Grammatical Evolution. *Chem. Lett.* **2018**, *47*, 1431–1434.

(10) Gomez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Śanchez- Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276.

(11) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep learning in drug discovery. *Mol. Inf.* **2016**, *35*, 3–14.

(12) Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press, 2016.

(13) Chollet, F. *Deep learning with Python;* Manning Publications Co: Shelter Island, NY, 2018.

(14) Foster, D.; Safari, A. O. M. C. Generative deep learning: teaching machines to paint, write, compose, and play; O'Reilly Media, 2019.

(15) White, D.; Wilson, R. C. Generative models for chemical structures. J. Chem. Inf. Model. 2010, 50, 1257–1274.

(16) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365.

(17) Yuan, W.; Jiang, D.; Nambiar, D. K.; Liew, L. P.; Hay, M. P.; Bloomstein, J.; Lu, P.; Turner, B.; Le, Q.-T.; Tibshirani, R.; Khatri, P.; Moloney, M. G.; Koong, A. C. Chemical Space Mimicry for Drug Discovery. J. Chem. Inf. Model. **2017**, *57*, 875–882.

(18) Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120–131.

(19) Elton, D. C.; Boukouvalas, Z.; Fuge, M. D.; Chung, P. W. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.* **2019**, *4*, 828–849.

(20) Schwalbe-Koda, D.; Gomez-Bombarelli, R. In *Machine Learning Meets Quantum Physics*; Schütt, K. T., Chmiela, S., von Lilienfeld, O. A., Tkatchenko, A., Tsuda, K., Müller, K.-R., Eds.; Springer International Publishing: Cham, 2020; pp 445–467.

(21) Zhavoronkov, A.; Vanhaelen, Q.; Oprea, T. I. Will Artificial Intelligence for Drug Discovery Impact Clinical Pharmacology? *Clin. Pharmacol. Ther.* (*N. Y., NY, U. S.*) **2020**, *107*, 780–785.

(22) Bian, Y.; Xie, X.-Q. Generative chemistry: drug discovery with deep learning gener- ative models. *J. Mol. Model.* **2021**, *27*, 71.

(23) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discovery Today* **2018**, 23, 1241–1250.

(24) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discovery* **2019**, *18*, 463–477.

(25) Engel, T., Gasteiger, J., Eds. Chemoinformatics: basic concepts and methods; Wiley-VCH: Weinheim, 2018; OCLC: 1012130305.

(26) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **2019**, *47*, D1102–D1109.

(27) Ash, S.; Cline, M.; Homer, R. W.; Hurst, T.; Smith, G. SYBYL Line Notation (SLN): A Versatile Language for Chemical Structure Representation. J. Chem. Inf. Comput. Sci. **1997**, 37, 71–79.

(28) Koniver, D. A.; Wiswesser, W. J.; Usdin, E. Wiswesser Line Notation: Simplified Techniques for Converting Chemical Structures to WLN. *Science* **1972**, *176*, 1437–1439.

(29) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, 28, 31–36.

(30) O'Boyle, N. M. Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *J. Cheminf.* **2012**, *4*, 22.

(31) Bjerrum, E. J. SMILES enumeration as data augmentation for neural network modeling of molecules. *arXiv* (*Machine Learning*), May 17, **201**7, 703.07076, ver. 2.

(32) Bjerrum, E. J.; Sattarov, B. Improving chemical autoencoder latent space and molec- ular de novo generation diversity with heteroencoders. *Biomolecules* **2018**, *8*, 131.

(33) Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Rey- mond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminf.* **2019**, *11*, 71.

(34) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G. Generative molecular design in low data regimes. *Nature Machine Intelligence* **2020**, *2*, 171–180.

(35) van Deursen, R.; Ertl, P.; Tetko, I. V.; Godin, G. GEN: highly efficient SMILES explorer using autodidactic generative examination networks. *J. Cheminf.* **2020**, *12*, 22.

(36) Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; En- gkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminf.* **2019**, *11*, 74.

(37) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminf.* **2015**, *7*, 23.

(38) Winter, R.; Montanari, F.; Nóe, F.; Clevert, D.-A. Learning continuous and data- driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **2019**, *10*, 1692–1701.

(39) O'Boyle, N.; Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine- Learning of Chemical Structures; preprint, *ChemRxiv*, September 19, **2018**, ver. 1. DOI: 10.26434/chemrxiv.7097960.v1.

(40) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing em- bedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* **2020**, *1*, 045024.

(41) Faulon, J.-L., Bender, A., Eds. *Handbook of chemoinformatics algorithms*; Chapman & Hall/CRC mathematical and computational biology series; Chapman & Hall/CRC: Boca Raton, FL, 2010; Chapter 1. OCLC: ocn226357322.

(42) Wishart, D. S.; et al. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 2018, 46, D1074–D1082.

(43) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large- scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.

(44) Landrum, G. RDKit: open-source cheminformatics software, 2016. (45) Steinbeck, C.; Hoppe, C.; Kuhn, S.; Floris, M.; Guha, R.; Willighagen, E. Recent De-velopments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics. *Curr. Pharm. Des.* **2006**, *12*, 2111–2120.

(46) Sun, J.; Jeliazkova, N.; Chupakhin, V.; Golib-Dzib, J.-F.; Engkvist, O.; Carlsson, L.; Wegner, J.; Ceulemans, H.; Georgiev, I.; Jeliazkov, V.; Kochev, N.; Ashby, T. J.; Chen, H. ExCAPE-DB: an integrated large

scale dataset facilitating Big Data analysis in chemogenomics. J. Cheminf. 2017, 9, 17.

(47) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.

(48) Shivanyuk, A.; Ryabukhin, S.; Tolmachev, A.; Bogolyubsky, A.; Mykytenko, D.; Chupryna, A.; Heilman, W.; Kostyuk, A. Enamine real database: Making chemical diversity real. *Chem. Today* **2007**, *25*, 58– 59.

(49) Huang, R.; Xia, M.; Nguyen, D.-T.; Zhao, T.; Sakamuru, S.; Zhao, J.; Shahane, S. A.; Rossoshek, A.; Simeonov, A. Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Front. Environ. Sci.* **2016**, *3*, 85.

(50) Ramakrishnan, R.; Hartmann, M.; Tapavicza, E.; Von Lilienfeld, O. A. Electronic spectra from TDDFT and machine learning in chemical space. *J. Chem. Phys.* **2015**, *143*, 084111.

(51) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.

(52) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *J. Med. Chem.* **2004**, *47*, 2977– 2980.

(53) Cho, K.; Van Merrïenboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* (*Computation and Language*), October 7, **2014**, 1409.1259, ver. 2.

(54) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.

(55) Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *arXiv (Machine Learning)*, June 10, **2014**, 1406.2661, ver. 1.

(56) Kingma, D. P.; Welling, M. Auto-encoding variational bayes. arXiv (Machine Learning), May 1, **2014**, 1312.6114, ver. 10.

(57) Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv (Machine Learning)*, May 25, **2016**, 1511.05644, ver. 2..

(58) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **2017**, *9*, 48.

(59) Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for *De Novo* Drug Design. *Mol. Inf.* **2018**, *37*, 1700111.

(60) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru- Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv* (*Machine Learning*), February 7, **2018**, 1705.10843, ver. 3.

(61) Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Cheminf.* **2018**, *10*, 31.

(62) Merk, D.; Grisoni, F.; Friedrich, L.; Schneider, G. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid X receptor modulators. *Commun. Chem.* **2018**, *1*, 68.

(63) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. *De Novo* Design of Bioactive Small Molecules by Artificial Intelligence. *Mol. Inf.* **2018**, *37*, 1700153.

(64) Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery. *Mol. Pharmaceutics* **2018**, *15*, 4398–4405.

(65) Li, Y.; Vinyals, O.; Dyer, C.; Pascanu, R.; Battaglia, P. Learning Deep Generative Models of Graphs. *arXiv* (*Machine Learning*), March 8, **2018**, 1803.03324, ver. 1.

(66) Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A. Constrained graph variational autoencoders for molecule design. *Adv. Neural Inf. Process. Syst.* **2018**, 7795–7804.

(67) Mercado, R.; Rastemo, T.; Lindelof, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Bjerrum, E. J. Graph networks for molecular design. *Mach. Learn.: Sci. Technol.* **2021**, *2*, 025023.

(68) De Cao, N.; Kipf, T. MolGAN: An implicit generative model for small molecular graphs. *arXiv* (*Machine Learning*), May 30, **2018**, 1805.11973, ver. 1.

(69) Simonovsky, M.; Komodakis, N. Graphvae: Towards generation of small graphs us- ing variational autoencoders. *International Conference on Artificial Neural Networks.* **2018**, *11139*, 412–422.

(70) Ma, T.; Chen, J.; Xiao, C. Constrained Generation of Semantically Valid Graphs via Regularizing Variational Autoencoders. *Adv. Neural Inf. Process. Syst.* **2018**, 7113–7124.

(71) Hawkins, P. C. D. Conformation Generation: The State of the Art. J. Chem. Inf. Model. 2017, 57, 1747–1756.

(72) Skalic, M.; Jimenez, J.; Sabbadin, D.; De Fabritiis, G. Shape-Based Generative Mod- eling for de Novo Drug Design. *J. Chem. Inf. Model.* **2019**, *59*, 1205–1214.

(73) Gebauer, N.; Gastegger, M.; Schütt, K. T. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *NeurIPS*. **2019**.

(74) Ragoza, M.; Masuda, T.; Koes, D. R. Learning a Continuous Representation of 3D Molecular Structures with Deep Generative Models. *arXiv* (*Quantitative Methods*), November 15, **2020**, 2010.08687, ver. 3.

(75) Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. Frechet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. J. Chem. Inf. Model. **2018**, *58*, 1736–1741.

(76) Arús-Pous, J.; Blaschke, T.; Ulander, S.; Reymond, J.-L.; Chen, H.; Engkvist, O. Exploring the GDB-13 chemical space using deep generative models. J. Cheminf. **2019**, *11*, 1–14.

(77) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108.

(78) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Front. Pharmacol.* **2020**, *11*, 11.

(79) Renz, P.; Van Rompaey, D.; Wegner, J. K.; Hochreiter, S.; Klambauer, G. On fail- ure modes in molecule generation and optimization. *Drug Discovery Today: Technol.* **2019**, *32–33*, 55–63.

(80) Cieplinski, T.; Danel, T.; Podlewska, S.; Jastrzebski, S. We should at least be able to Design Molecules that Dock Well. *arXiv* (*Biomolecules*) December 28, **2020**, 2006.16955, ver. 3.

(81) Zhang, J.; Mercado, R.; Engkvist, O.; Chen, H. Comparative study of deep generative models on chemical space coverage. *ChemRxiv*, May 2, 2021, ver. 3. DOI: 10.26434/chemrxiv.13234289.v3.

(82) Blaschke, T.; Arús-Pous, J.; Chen, H.; Margreitter, C.; Tyrchan, C.; Engkvist, O.; Papadopoulos, K.; Patronov, A. REINVENT 2.0: An AI Tool for De Novo Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 5918.

(83) Bung, N.; Krishnan, S. R.; Bulusu, G.; Roy, A. De novo design of new chemical entities for SARS-CoV-2 using artificial intelligence. *Future Med. Chem.* **2021**, *13*, 575.

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

(85) Blaschke, T.; Engkvist, O.; Bajorath, J.; Chen, H. Memoryassisted reinforcement learning for diverse molecular de novo design. *J. Cheminf.* **2020**, *12*, 1–17.

(86) Zhavoronkov, A.; et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038–1040.

(87) Popova, M.; Shvets, M.; Oliva, J.; Isayev, O. MolecularRNN: Generating real- istic molecular graphs with optimized properties. *arXiv* (*Machine Learning*), May 31, **2019**, 1905.13372, ver. 1.

(88) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry

(ORGANIC). ChemRxiv, August 18, 2017, ver. 3. DOI: 10.26434/ chemrxiv.5309668.v3.

(89) Putin, E.; Asadulaev, A.; Vanhaelen, Q.; Ivanenkov, Y.; Aladinskaya, A. V.; Aliper, A.; Zhavoronkov, A. Adversarial Threshold Neural Computer for Molecular *de Novo* De- sign. *Mol. Pharmaceutics* **2018**, *15*, 4386–4397.

(90) Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced Adversarial Neural Computer for *de Novo* Molecular Design. *J. Chem. Inf. Model.* **2018**, *58*, 1194–1204.

(91) You, J.; Liu, B.; Ying, Z.; Pande, V.; Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6410–6421.

(92) Karimi, M.; Hasanzadeh, A.; Shen, Y. Network-principled deep generative models for designing drug combinations as graph sets. *Bioinformatics* **2020**, *36*, i445–i454.

(93) Griffiths, R.-R.; Hernández-Lobato, J. M. Constrained Bayesian optimization for auto- matic chemical design using variational autoencoders. *Chem. Sci.* **2020**, *11*, 577–586.

(94) Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of Generative Autoencoder in *De Novo* Molecular Design. *Mol. Inf.* **2018**, *37*, 1700123.

(95) Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar variational autoencoder. *Proc. 34th Int. Conf. Mach. Learn.* **2017**, *70*, 1945–1954.

(96) Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; Song, L. Syntax-directed variational autoencoder for structured data. *arXiv* (*Machine Learning*), February 24, **2018**, 1802.08786, ver 1.

(97) Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *Proc. 35th Int. Conf. Mach. Learn.* **2018**, *50*, 2323–2332.

(98) Samanta, B.; De, A.; Jana, G.; Chattaraj, P. K.; Ganguly, N.; Rodriguez, M. G. NeVAE: A Deep Generative Model for Molecular Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence* **2019**, 33, 1110–1117.

(99) Bresson, X.; Laurent, T. A Two-Step Graph Convolutional Decoder for Molecule Generation. *arXiv* (*Machine Learning*), June 15, **2019**, 1906.03412, ver 2.

(100) Maziarka, L.; Pocha, A.; Kaczmarczyk, J.; Rataj, K.; Danel, T.; Warcho-l, M. Mol- CycleGAN: a generative model for molecular optimization. J. Cheminf. **2020**, *12*, 2.

(101) Sattarov, B.; Baskin, I. I.; Horvath, D.; Marcou, G.; Bjerrum, E. J.; Varnek, A. De Novo Molecular Design by Combining Deep Deep Autoencoder Recurrent Neural Networks with Generative Topographic Mapping. *J. Chem. Inf. Model.* **2019**, *59*, 1182–1196.

(102) Winter, R.; Montanari, F.; Steffen, A.; Briem, H.; Nóe, F.; Clevert, D.-A. Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci.* **2019**, *10*, 8016–8024.

(103) Chenthamarakshan, V.; Das, P.; Hoffman, C. S.; Strobelt, H.; Padhi, I.; Lim, W. K.; Hoover, B.; Manica, M.; Born, J.; Laino, T.; Mojsilovic, A. CogMol: Target-Specific and Selective Drug Design for COVID-19 Using Deep Generative Models. *NeurIPS 2020* 2020.

(104) Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence* **2020**, *2*, 254–265.

(105) Shayakhmetov, R.; Kuznetsov, M.; Zhebrak, A.; Kadurin, A.; Nikolenko, S.; Aliper, A.; Polykovskiy, D. Molecular Generation for Desired Transcriptome Changes With Ad- versarial Autoencoders. *Front. Pharmacol.* **2020**, *11*, 269.

(106) Mendez-Lucio, O.; Baillif, B.; Clevert, D.-A.; Rouquíe, D.; Wichard, J. De novo gener- ation of hit-like molecules from gene expression signatures using artificial intelligence. *Nat. Commun.* **2020**, 11, 1-10.

(107) Born, J.; Manica, M.; Oskooei, A.; Cadow, J.; Rodŕiguez Marítnez, M. PaccMannRL: Designing Anticancer Drugs From Transcriptomic Data via Reinforcement Learning. In *Research in Computational Molecular Biology*; Springer: Cham, 2020; pp 231–233. (108) Jin, W.; Yang, K.; Barzilay, R.; Jaakkola, T. Learning Multimodal Graph-to-Graph Translation for Molecular Optimization. *arXiv* (*Machine Learning*), January 28, **2019**, 1812.01070, ver. 3.

(109) Masuda, T.; Ragoza, M.; Koes, D. R. Generating 3D Molecular Structures Conditional on a Receptor Binding Site with Deep Generative Models. *arXiv* (*Chemical Physics*), November 23, 2020, 2010.14442, ver. 3.

(110) Kang, S.; Cho, K. Conditional Molecular Design with Deep Generative Models. J. Chem. Inf. Model. 2019, 59, 43–52.

(111) Lim, J.; Hwang, S.-Y.; Moon, S.; Kim, S.; Kim, W. Y. Scaffoldbased molecular design with a graph generative model. *Chem. Sci.* 2020, *11*, 1153–1164.

(112) Varnek, A., Ed. Tutorials in chemoinformatics; John Wiley & Sons, Inc: Hoboken, NJ, 2017.

(113) Engel, T., Gasteiger, J., Eds. *Applied chemoinformatics:* achievements and future opportunities; Wiley-VCH: Weinheim, 2018; OCLC: 1034693178.

(114) Kadurin, A.; Aliper, A.; Kazennov, A.; Mamoshina, P.; Vanhaelen, Q.; Khrabrov, K.; Zhavoronkov, A. The cornucopia of meaningful leads: Applying deep adversarial au- toencoders for new molecule development in oncology. *Oncotarget* **2017**, *8*, 10883–10890.

(115) Alpaydin, E. Introduction to machine learning, 2nd ed.; *Adaptive computation and machine learning*; MIT Press: Cambridge, Mass, 2010; OCLC: ocn317698631.

(116) Raschka, S. Python machine learning: unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics; Community experience distilled; Packt Publishing Open Source: Birmingham, UK; Mumbai, 2016.

(117) Frazier, P. I. A Tutorial on Bayesian Optimization. arXiv (Machine Learning), July 8, 2018, 1807.02811, ver. 1.

(118) Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2016**, *104*, 148–175.

(119) Das, P.; Sercu, T.; Wadhawan, K.; Padhi, I.; Gehrmann, S.; Cipcigan, F.; Chen- thamarakshan, V.; Strobelt, H.; Santos, C. D.; Chen, P.-Y.; Yang, Y. Y.; Tan, J.; Hedrick, J.; Crain, J.; Mojsilovic, A. Accelerating antimicrobial discovery with controllable deep generative models and molecular dynamics. *arXiv* (*Machine Learning*), February 26, **2020**, 2005.11248, ver. 2.

(120) Kingma, D. P.; Mohamed, S.; Rezende, D. J.; Welling, M. Semisupervised learning with deep generative models. *Adv. Neural Inf. Process. Syst.* **2014**, 3581–3589.

(121) Gao, W.; Coley, C. W. The synthesizability of molecules proposed by generative mod- els. *J. Chem. Inf. Model.* **2020**, *60*, 5714–5723.

(122) Horwood, J.; Noutahi, E. Molecular Design in Synthetically Accessible Chemical Space via Deep Reinforcement Learning. ACS Omega 2020, 5, 32984–32994.

(123) Gottipati, S. K.; Sattarov, B.; Niu, S.; Pathak, Y.; Wei, H.; Liu, S.; Blackburn, S.; Thomas, K.; Coley, C.; Tang, J.; et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. *Int. Conf. Mach. Learn.* **2020**, 3668–3679.

(124) Bradshaw, J.; Paige, B.; Kusner, M. J.; Segler, M.; Hernández-Lobato, J. M. Barking up the right tree: an approach to search over molecule synthesis DAGs. *Adv. Neural Inf. Process. Syst.* **2020**, 6852–6866.

(125) Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Deep generative models for 3d linker design. *J. Chem. Inf. Model.* **2020**, *60*, 1983–1995.

(126) Yang, Y.; Zheng, S.; Su, S.; Zhao, C.; Xu, J.; Chen, H. SyntaLinker: automatic fragment linking with deep conditional transformer neural networks. *Chem. Sci.* **2020**, *11*, 8312–8322.

(127) Tan, X.; et al. Automated design and optimization of multitarget schizophrenia drug candidates by deep learning. *Eur. J. Med. Chem.* **2020**, *204*, 112572.

(128) Yang, Y.; Zhang, R.; Li, Z.; Mei, L.; Wan, S.; Ding, H.; Chen, Z.; Xing, J.; Feng, H.; Han, J.; Jiang, H.; Zheng, M.; Luo, C.; Zhou, B. Discovery of Highly Potent, Selec- tive, and Orally Efficacious p300/ CBP Histone Acetyltransferases Inhibitors. J. Med. Chem. 2020, 63, 1337–1360.

(129) Grisoni, F.; Huisman, B.; Button, A.; Moret, M.; Atz, K.; Merk, D.; Schneider, G. Combining generative artificial intelligence and onchip synthesis for de novo drug design. *ChemRxiv*, December 30, **2020**, ver. 1. DOI: 10.26434/chemrxiv.13498587.v1.

(130) Shaker, N.; Abou-Zleikha, M.; AlAmri, M.; Mehellou, Y. A Generative Deep Learning Approach for the Discovery of SARS CoV2 Protease Inhibitors. *ChemRxiv*, April 23, **2020**, ver. 1. DOI: 10.26434/ chemrxiv.12170337.vl.

(131) Born, J.; Manica, M.; Cadow, J.; Markert, G.; Mill, N. A.; Filipavicius, M.; Martínez, M. R. PaccMannRL on SARS-CoV-2: Designing antiviral candidates with conditional generative models. *arXiv* (*Quantitative Methods*), July 6, **2020**, 2005.13285, ver. 3.

(132) Hachmann, J.; Olivares-Amaya, R.; Atahan-Evrenk, S.; Amador-Bedolla, C.; Śanchez- Carrera, R. S.; Gold-Parker, A.; Vogt, L.; Brockway, A. M.; Aspuru-Guzik, A. The Harvard clean energy project: large-scale computational screening and design of or- ganic photo-voltaics on the world community grid. *J. Phys. Chem. Lett.* **2011**, *2*, 2241–2251.

(133) Jørgensen, P. B.; Mesta, M.; Shil, S.; Lastra, J. M. G.; Wedel, K.; Thygesen, K. S.; Schmidt, M. N. Machine learning-based screening of complex molecules for polymer solar cells. *J. Chem. Phys.* **2018**, *148*, 241735.

(134) Yuan, Q.; Santana-Bonilla, A.; Zwijnenburg, M. A.; Jelfs, K. E. Molecular generation targeting desired electronic properties via deep generative models. *Nanoscale* **2020**, *12*, 6744–6758.

(135) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* (*Computation and Language*), December 6, **2017**, 1706.03762, ver. 5.

(136) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pretraining of deep bidirectional transformers for language understanding. *arXiv* (*Computation and Language*), May 24, **2019**, 1810.04805, ver. 2.

(137) Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Nee- lakantan, A.; Shyam, P.; Sastry, G.; Askell, A., et al. Language models are few-shot learners. *arXiv* (*Computation and Language*), July 22, **2020**, 2005.14165, ver. 4.

(138) Grechishnikova, D. Transformer neural network for proteinspecific de novo drug gen- eration as a machine translation problem. *Sci. Rep.* **2021**, *11*, 1–13.

(139) Zheng, S.; Lei, Z.; Ai, H.; Chen, H.; Deng, D.; Yang, Y. Deep Scaffold Hopping with Multi-modal Transformer Neural Networks. *ChemRxiv*, September 28, **2020**, ver. 1. DOI: 10.26434/chemrxiv.13011767.v1.