Received: 1 October 2021 | Revised: 24 January 2022 | Accepted: 13 February 2022

DOI: 10.1002/wcms.1608

ADVANCED REVIEW

WIREs COMPUTATIONAL MOLECULAR SCIENCE WILEY

Generative models for molecular discovery: Recent advances and challenges

Camille Bilodeau[1] | **Wengong Jin**[2] | **Tommi Jaakkola**[2] | **Regina Barzilay**[2] | **Klavs F. Jensen**[1]

[1]Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

[2]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Correspondence
Klavs F. Jensen, Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA.
Email: kfjensen@mit.edu

Funding information
This work was supported by Dow Chemical Company, the MIT consortium for Pharmaceutical Discovery and Synthesis, and the DARPA Accelerated Molecular Discovery program HR00111920025.

Edited by: Raghavan Sunoj, Associate Editor

Abstract

Development of new products often relies on the discovery of novel molecules. While conventional molecular design involves using human expertise to propose, synthesize, and test new molecules, this process can be cost and time intensive, limiting the number of molecules that can be reasonably tested. Generative modeling provides an alternative approach to molecular discovery by reformulating molecular design as an inverse design problem. Here, we review the recent advances in the state-of-the-art of generative molecular design and discusses the considerations for integrating these models into real molecular discovery campaigns. We first review the model design choices required to develop and train a generative model including common 1D, 2D, and 3D representations of molecules and typical generative modeling neural network architectures. We then describe different problem statements for molecular discovery applications and explore the benchmarks used to evaluate models based on those problem statements. Finally, we discuss the important factors that play a role in integrating generative models into experimental workflows. Our aim is that this review will equip the reader with the information and context necessary to utilize generative modeling within their domain.

This article is categorized under:

Data Science > Artificial Intelligence/Machine Learning

KEYWORDS

generative adversarial networks, generative models, molecular representation, normalizing flow models, variational autoencoders

1 | INTRODUCTION

The discovery of new molecules and materials is needed to address important challenges in chemistry ranging from treating complex diseases to tackling climate change. Conventionally, this process relies on human expertise to propose, synthesize, and test new molecules. While many of the most important discoveries in scientific history have been made in this fashion, conventional discovery is inherently cost and time intensive, limiting the number and diversity of molecules that can be reasonably explored. It has been estimated that the number of compounds that have ever been synthesized lies around 10^8

while the total number of theoretically feasible compounds lies between 10^{23} and 10^{60}.[1,2] Thus, conventional discovery methods are only capable of exploring a small fraction of chemical space. There is a critical need to develop methods that can efficiently explore chemical space to identify molecules capable of solving important problems in chemistry and engineering.

Generative models offer a promising solution. Instead of designing molecules using human expertise, generative models leverage recent advances in deep learning to address the inverse molecular design problem: given a desired set of properties, what is the set of molecules that will satisfy those properties? By identifying a function that maps a set of properties to a set of structures, generative models can rapidly identify diverse sets of molecules that are highly optimized for specific applications. Since their recent introduction, there has been explosion in the number and variety of generative models that have been applied to molecular design. These models vary in their molecular representation, architecture, and the type of molecular design problems they address. Further, to facilitate comparisons between the growing number of models, a number of benchmarks have been recently proposed that evaluate models based on factors such as distribution learning and chemical diversity and novelty.[3,4] A number of excellent reviews have been written to summarize the development of this field.[5–10]

Despite these remarkable advances, there are relatively few examples of applying generative models to discover molecules for concrete applications. Most studies focus on optimizing molecules for computational metrics such as logP (logarithm of the partition coefficient) or QED[11] (quantitative estimate of drug-likeliness) and few published studies involve experimentally testing the identified lead molecules. Thus, there are clearly gaps that prevent generative models from reaching their full potential in molecular discovery.

In this review, we provide a concise summary of the recent advances in the state of the art of generative molecular design, describing the considerations for integrating these models into real molecular discovery campaigns, and discussing the remaining challenges that must be solved for their promise to be fully realized. Our aim is that this review will equip the reader with the information necessary to fully leverage generative modeling in their field of expertise.

2 | BACKGROUND

2.1 | Representation of molecules

The strength of neural networks lies in their ability to take in a complex input representation transform it into a latent representation needed to solve a particular task. In this way, the choice of input representation plays a key role in governing how the model learns information about the molecule. Input representations often fall into one of three categories: (1) one-dimensional (e.g., string-based representations), (2) two-dimensional (e.g., molecular graphs), and (3) three-dimensional representations (e.g., coordinate-based).

2.1.1 | One-dimensional representations

The most common type of one-dimensional representation is referred to as SMILES (Simplified Molecular Input Line Entry System),[12] a simple string-based representation that transforms a molecule into a sequence of characters based on predefined atom ordering rules. Representing a molecule as a sequence of characters has proven advantageous because it enables the reapplication of neural network architectures that have been previously developed for language processing. In particular, by representing molecules as sequences, prior work[13–20] trained recurrent neural networks as generative models[21,22] that generate the SMILES strings of molecules. Unfortunately, these methods are prone to generate invalid SMILES that cannot be converted to molecular structures because they ignore the complex grammar of the SMILES notation. To remedy this issue, Kusner et al. and Dai et al. enhanced the recurrent neural network with syntactic constraints of the SMILES grammar. However, these methods still fail to capture chemical validity and often generate invalid SMILES strings.[23,24] Given the complication of SMILES notation, Krenn et al. designed an improved string representation called SELFIES (Self-Referencing Embedded Strings).[25] Based on the SELFIES representation, a recurrent neural network model can be trained generate molecules that are 100% valid. It is important to note, however, that validity is defined with respect to valency rules. As a result, molecules that are valid may not necessarily be stable.

2.1.2 | Two-dimensional representations

Molecules can alternatively be represented in neural networks as graphs, with nodes and edges corresponding to atoms and bonds, respectively. Graphical representations are powerful in that they directly capture the connectivity between atoms, while in one-dimensional representations this information must inferred by the model. Unfortunately, graphs have also proven more difficult generate than sequences and, as a result, there have been a number of efforts aimed at developing neural network architectures for generating realistic molecular graphs. One strategy is to generate molecular graphs by outputting the atoms and the adjacency matrix of the graphs simultaneously.[26–28] In contrast, You et al.,[29] Li et al.,[30] Samanta et al.,[31] and Liu et al.[32] developed generative models that decode molecules sequentially, atom-by-atom. Jin et al.[33] took a related approach, grouping atoms into substructures and developing a model that generates molecules substructure-by-substructure (also sequentially). These substructures consisted of either two atoms connected by a bond or all of the atoms in a ring (e.g., a phenyl ring). Their model first generated a junction tree with substructures as nodes and then predicted how the substructures should be attached to each other. Jin et al.[34] later extended this approach to a hierarchical model, allowing the use of larger substructures. Their model outperformed atom-by-atom approaches on multiple molecule generation tasks, including polymer generation, drug-likeness optimization, and Dopamine Receptor D2 binding affinity optimization.

2.1.3 | Three-dimensional representations

Lastly, molecules can be represented in three dimensions using point clouds—where each atom corresponds to a point in space—in order to capture not only covalent atom connectivity, but also information about the conformational preferences of the molecule. For example, Gebauer et al.[35] generated molecules sequentially by placing atoms in Cartesian coordinates. Simm et al.[36] adopted a similar approach that generated molecules atom-by-atom based on internal coordinates. One disadvantage to using these methods for molecular discovery applications is that to accurately capture physical properties it is necessary to consider more than one conformer of a given molecule. A second disadvantage to these approaches is that they typically use energy minimization with force fields to generate conformations for molecules in the training set, which can be time-consuming, especially for large, flexible molecules. Recently, researchers have explored replacing/supplementing traditional conformer generation methods[37,38] with three-dimensional generative models[39–41] trained on large datasets of conformer ensembles,[42,43] which can be used to reduce the computational expense of obtaining three-dimensional training data.

2.2 | Model architectures

The success of a given deep neural network depends greatly on its architecture—the types of layers that comprise the network and the way in which those layers are arranged. Deep generative models for molecular discovery can be broken up into three classes of neural network architectures: variational autoencoders (VAEs), generative adversarial networks (GANs), and normalizing flow models, illustrated in Figure 1. Each architecture differs in the strategy it uses to learn a latent representation of the molecule. Both VAEs and normalizing flow models aim to maximize the likelihood of the training data. VAEs approximately maximize likelihood using variational inference techniques,[44] while normalizing flow models maximize the likelihood exactly by requiring the model to be invertible. In contrast, GAN-based approaches formulate molecule generation as a minimax game, where a discriminator model learns to distinguish the real data from the fake samples that are produced by a generator model. Here we will briefly describe how these methods work and the ways that they have been used for molecular discovery applications. For greater detail on these methods (including mathematical formulations), we refer the reader to previous reviews and resources.[5,45]

2.2.1 | Variational autoencoders

VAEs are generative models that consist of an encoder, which learns to map a molecule into a continuous embedding, followed by a decoder, which learns to reconstruct a molecule from its learned embedding.[46] VAEs are trained using a loss

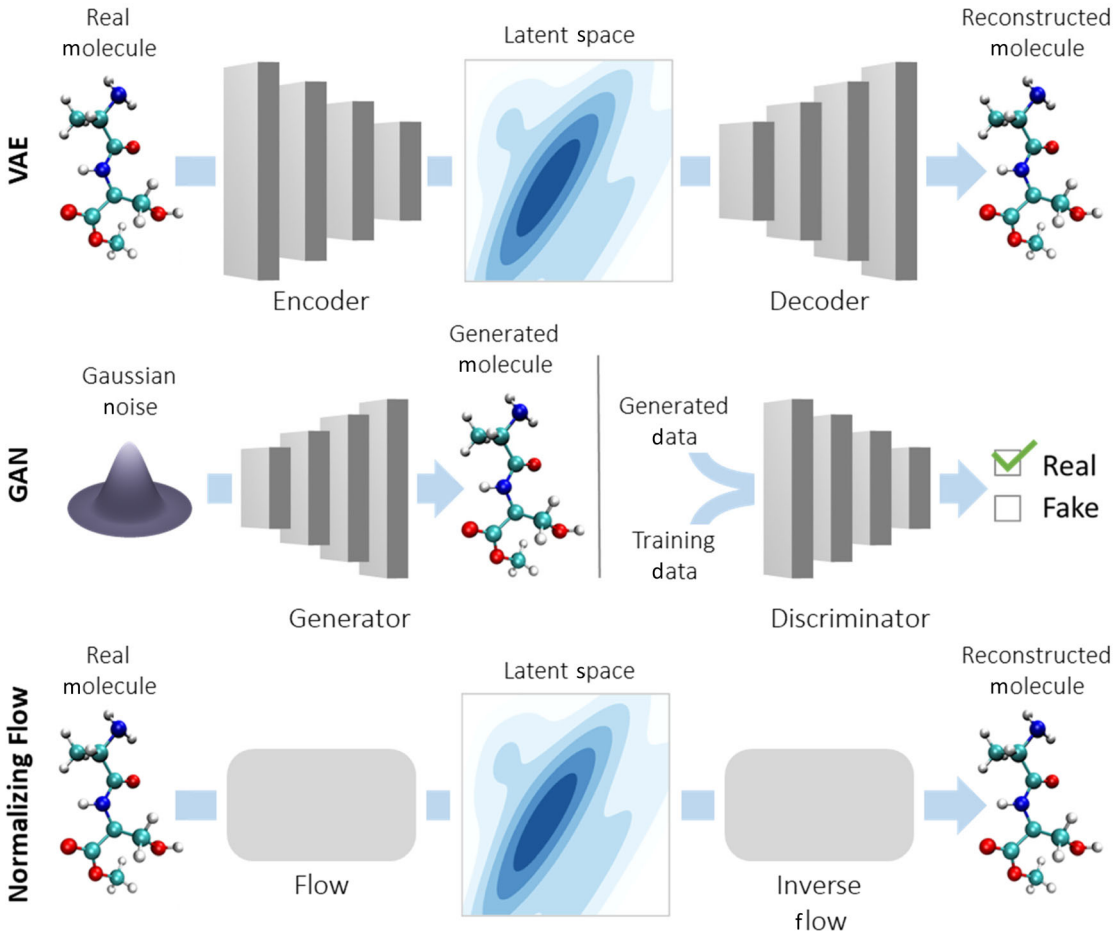

FIGURE 1 Comparison of the major neural network architectures used in generative modeling: Variational autoencoders (VAEs, top), generative adversarial networks (GANs, middle), and normalizing flow models (bottom)

function consisting of two terms: (1) a reconstruction loss that forces the decoder to recover the correct molecule from its embedding and (2) a Kullback–Leibler (KL) divergence term that regularizes the distribution of learned molecular embeddings such that the distribution of generated molecules closely resembles the training distribution. In the context of molecule generation, VAEs have been used to generate SMILES strings[13,19,23,24] and molecular graphs.[32,33,47,48]

2.2.2 | Generative adversarial networks

GANs are generative models that consist of a generator, which learns to generate a molecule from Gaussian noise, and a discriminator, which learns to identify molecules as either real (belonging to a training dataset) or fake (constructed by the generator). These two networks are trained to compete against one another, with the generator learning to generate sufficiently realistic molecules to trick the discriminator, and the discriminator learning to tell the difference. GANs have been successful at generating highly realistic images,[49] in part because adversarial training enables the model to learn a more nuanced definition of what makes an example realistic than is accessible to VAEs through their loss functions.

Recently, researchers have adapted this approach to generating molecules, represented as sequences or graph.[50–52] For example, Guimaraes et al.[53] trained a GAN to generate SMILES strings and, as described earlier, De Cao and Kipf[26] and Ma et al.[27] trained GANs to generate the adjacency matrix of a molecular graph. Nevertheless, generating sequences and graphs using GANs remains challenging because constructing sequences and graphs requires backpropagating the gradient through discrete choices.

2.2.3 | Normalizing flow models

Normalizing flow models generate molecules by learning a series of invertible transformations between a prior distribution (e.g., a Gaussian distribution) and real-world high-dimensional data such as molecules. The major advantage of flow-based models over VAEs is that the invertible mapping allows the calculation of the exact data likelihood. This advantage motivated Zang et al.,[54] Shi et al.,[55] and Madhawa et al.[56] to apply flow-based models to molecule generation. For example, Shi et al. developed an autoregressive flow-based model called GraphAF, which outperformed variational autoencoder based methods on logP optimization tasks.

2.2.4 | Miscellaneous

Beyond the three major generative model classes, researchers have explored other types of models for tasks that are closely related to molecular design. For instance, diffusion-based models have been applied to molecular conformation generation,[57] where a model learns to iteratively modify pairwise atomic distances in order to generate stable conformations. While the full combination of a graph and a conformer diffusion model is not yet available, they can be readily combined. In addition, Bradshaw et al.[58,59] has proposed reaction-based models for molecular design in order to generate molecules that are easy to synthesize. We will further discuss this type of approach in Section 3.

2.3 | Formulating the molecular generation problem

There are a wide range of reasons one might be interested in discovering new molecules. A first step in applying generative modeling to molecular generation is to formulate these distinct applications as concrete problem statements, for example, we are interested in discovering molecules with X properties, subject to Y constraints. Broadly, molecular generation problem statements fall into three classes: (1) unconstrained molecular generation, (2) property-constrained molecular generation, and (3) structure-constrained molecular generation.

2.3.1 | Unconstrained molecular generation

The goal of unconstrained molecular generation is to generate diverse and novel molecules without any property constraints (except chemical validity). This is valuable for exploratory molecular generation campaigns, which focus on identifying interesting and unusual chemistries. For this class of problems, the generative model aims to learn the general distribution of molecules in chemical space (e.g., what do molecules typically look like?). To learn this broad distribution, the generative model is typically trained on a large databases of compounds such as ChEMBL[60] and ZINC.[61] The distribution is then learned by training the model to maximize the likelihood of all the compounds in the training set. Unconstrained generative models are typically evaluated based on the chemical validity, novelty, and uniqueness of generated compounds. For example, Segler et al.[17] trained a recurrent neural network to generate molecules without any property constraints. Their goal was to use generated molecules to enrich existing molecular libraries for virtual screening.

2.3.2 | Property-constrained molecular generation

Property-constrained molecular generation extends the previous formulation by adding constraints on the generated molecules. In this case, a model must generate compounds that are both chemically valid and have specific, desirable properties such as good solubility, low toxicity, or high potency. Since it is infeasible to experimentally validate every generated compound, it is necessary to train a property predictor to assess compound properties in silico, also known as quantitative structure–activity relationship (QSAR) model. The property predictor is trained on a separate dataset of molecules labeled with their properties (e.g., IC50/EC50 for potency). After training, the property predictor is used to estimate whether a generated molecule satisfies the given constraints.

In this way, the generative model learns to generate compounds that are predicted to satisfy the constraints via the property predictor. This task is often cast as a discrete optimization problem and can be solved by reinforcement learning, Bayesian optimization, or genetic algorithms. In reinforcement learning, a model is trained to maximize the expected reward defined based on the output of the property predictor. For instance, Olivecrona et al.[14] first pretrained a recurrent neural network on over 1 million drug-like molecules from ChEMBL to generate SMILES strings. They subsequently trained a property predictor to assess Dopamine Receptor D2 (DRD2) binding affinity of new compounds. Finally, they used reinforcement learning to fine-tune their generative model to maximize the predicted DRD2 binding affinity. Popova et al.[15] adopted a similar strategy to train a generative model that maximizes the inhibitory activity against Janus protein kinase 2. You et al.[29] and Zhou et al.[62] also used reinforcement learning to generate drug-like compounds with high QED scores. Their method operated on molecular graphs instead of SMILES strings and learned to iteratively add or remove atoms and bonds from a molecule in order to maximize the expected reward.

Alternatively, Bayesian optimization methods[13,23,33,63] can be used to cast the discrete optimization problem into a continuous optimization problem by learning a continuous embedding of molecules. These methods involve first training a variational autoencoder to map discrete molecules into a continuous embedding space and then training another neural network to predict the chemical property of the original molecule from its continuous embedding vector. They then apply Bayesian optimization in the continuous embedding space to find an embedding with the best associated property score. The discovered embedding is decoded into a discrete molecule by a decoder network.

Finally, genetic algorithms solve the discrete optimization problem by searching for favorable compounds via mutation of molecules. A genetic algorithm consists of two components: a set of mutation rules and a fitness function. The fitness function is a weighted interpolation of predicted property scores and penalty scores that penalizes long-surviving molecules. The additional penalty term encourages the model to explore a diverse set of molecules. New compounds are derived by applying mutation rules to existing molecules. Old molecules with low fitness scores are removed in each iteration. Nigam et al. applied genetic algorithms to design molecules with high logP scores.[64] Their fitness function was parameterized as a neural network. Ahn et al.[65] extended this approach by learning a policy function to perform mutations over molecules rather than using a fixed set of mutation rules. It is important to note that while many studies use logP as convenient metric to use for method development, it is an artificial task that cannot be directly tied to any real application.

2.3.3 | Structure-constrained molecular generation

The goal of structure-constrained molecular generation is to modify the structure of a candidate molecule in order to improve its properties. This approach is useful for molecular generation campaigns where lead candidates with desirable properties have already been identified and the goal is to explore closely related molecules. In the pharmaceutical industry, this process is analogous to lead optimization.[66] An example of structure-constrained optimization is the work published by Jin et al., 2019[67] and 2020[34] which formulated lead optimization as a graph-to-graph translation problem, where the model learned to translate input molecules into improved molecules. The model was trained on a dataset of molecular pairs, with each pair containing two similar molecules, one less optimal than the other. At test time, the translation model learned to generate analogs of a given molecule with better properties.

Another strategy for structure-constrained molecular generation involves constraining the output molecule to contain a specific scaffold or fragment. Langevin et al.[68] and Li et al.[69] built generative models that output drug molecules with a specific scaffold. The scaffolds were usually extracted from existing drugs with good biological properties. Jin et al.,[70] Podda et al.,[71] Imrie et al.,[72] and Green et al.[73] similarly developed models that learned to generate molecules with specific fragments. In this approach, models were first initialized with certain fragments. Models then learned to connect the given fragments into a coherent molecule by a series of edge selection and bond-type prediction steps.

2.4 | Benchmarks for generative models for molecular design

As has been seen in the previous sections, a large number of generative models with a variety of network architectures have been published for molecular generation. In order to rigorously determine whether one model is better or worse than another it is necessary to develop benchmarking metrics that can be evaluated for models trained on publicly available datasets. As evidenced by the different types of molecular generation problem statements, there are a variety

TABLE 1 Common generative modeling benchmarks and their definitions

Benchmark	Definition	Goal type	Benchmark set
Validity	Percent of valid molecules with respect to bonds/valency, evaluated using RDKit's molecular structure parser	Unconstrained	MOSES, Guacamol
Filters	Percent of molecules passing a set of rule-based filters	Unconstrained	MOSES, Guacamol
Uniqueness	Percent of unique molecules	Unconstrained	MOSES, Guacamol
KL divergence	Statistic capturing the difference between the generated molecular distribution and the training set distribution	Unconstrained	Guacamol
Molecular properties distributions	Comparison between generated and training set property distributions such as molecular weight or logP	Unconstrained	MOSES
Novelty	Percent of molecules not present in training set	Unconstrained	MOSES, Guacamol
Fragment similarity	Metric comparing distributions of BRICS fragments in the generated and training sets	Unconstrained	MOSES
Scaffold similarity	Metric comparing distributions of Bemis-Murcko scaffolds in the generated and training sets	Unconstrained	MOSES
Chemical space coverage	Extent to which generated molecules cover the chemical space of the GDB13 dataset	Unconstrained	Zhang et al.[74]
Similarity	Generate molecules that are similar to a target molecule removed from the training data	Goal-oriented	Guacamol
Rediscovery	Rediscover molecules removed from training data	Goal-oriented	Guacamol
Isomer identification	Generated molecules that follow a simple (unknown) pattern	Goal-oriented	Guacamol
Median molecule discovery	Generate molecules that maximize similarity to multiple molecules	Goal-oriented	Guacamol
Calculated property optimization	Generate molecules with optimum calculated property values such as LogP or QED	Goal-oriented	Guacamol

of goals for generative models and, as such, it is necessary to use benchmarking metrics that are representative of these goals. Two recent benchmarking sets, MOSES[3] and Guacamol,[4] define such metrics for both unconstrained molecular generation and for goal-oriented molecular generation (including both property constrained and structure constrained molecule generation, shown in Table 1). It is important to note that while these benchmarks are useful for comparing generative modeling approaches, they do not cover all of the necessary aspects of molecular discovery.[75]

2.4.1 | Unconstrained molecular generation benchmarks

The goals for unconstrained molecular generation are to generate molecules that are (1) valid and unique, (2) based on a chemical distribution that matches the training set, and (3) novel and diverse. Molecule validity is typically measured with respect to valency and bonds using RDKit's molecular structure parser. A more stringent metric for validity is defined as the percent of molecules that pass a given set of rule-based filters such as Walters's rd_filters implementation,[76] a filter set that includes heuristic rules such as a maximum ring size. One caveat to using rule-based filters is that they are typically defined based on realistic molecules and can miss unusual functional groups produced by generative models. A model's generated chemical distribution can be measured based on KL divergence, a metric that measures how well a given probability distribution approximates a second distribution. Chemical distribution can also be evaluated by comparing the distributions of common molecular properties such as molecular weight or logP in both the training and the test sets. The diversity of generated molecules is can be measured via a novelty metric, Fragment and Scaffold similarity (which use BRICS fragments[77] and Bemis-Murcko scaffolds[78] respectively), or Frechet

ChemNet Distance.[79] Finally, Zhang et al.[74] recently proposed a metric that compares models based on chemical space coverage of a large reference dataset, GDB13.

2.4.2 | Goal-directed molecular generation benchmarks

With goal-directed generative models, the aim is to discover molecules with specific properties. To test a model's ability to generate molecules with specific properties in an easy and reproducible fashion, Guacamol proposes benchmarks based on similarity, rediscovery, isomer identification, and median molecule generation.[4] The goal of the similarity benchmark is to generate molecules that are similar to a target molecule that was removed from the training set. The rediscovery benchmark is related to similarity, with the goal of rediscovering molecule(s) that have been removed from the training set. The isomer benchmark involves generating molecules that follow a simple pattern (which is a priori unknown). Finally, the goal of the median molecule discovery benchmark is to generate molecules to maximize similarity to multiple molecules. In addition, properties that can be easily measured computationally, such as logP or QED are commonly used as additional, goal-oriented benchmarks. It is important to note that many of these benchmarks have been developed or selected in order to measure model performance quickly and reproducibly. As a consequence, objectives (most notably the maximization of logP) are highly artificial and are not closely related to any real application.

3 | PRACTICAL CONSIDERATIONS FOR GENERATING MOLECULES FOR SPECIFIC APPLICATIONS

As described in the previous section, a variety of generative models have been developed and explored theoretically and computationally. Despite this, there remain relatively few examples of using generative models in real molecular discovery campaigns. This is because these campaigns are often characterized by a series of additional hurdles, making it difficult to directly deploy generative models for concrete applications. In this section, we will discuss these hurdles, in particular focusing on (1) the multi-objective nature of real molecular design problems, (2) the requirement that discovered molecules must be synthesizable, and (3) the challenges associated with error-prone predictive models.

3.1 | Real molecular design problems are generally multi-objective in nature

In applied settings, we are often interested in discovering molecules that are more optimal than any previously known molecule with respect to one or more properties. In addition to these properties, there are usually additional objectives or constraints that are secondary to the main design goal but are equally necessary for the molecule to be useful for a given application. To illustrate this, consider the problem of designing a drug molecule: the primary objective may be to discover a lead molecule that binds to a target with high affinity, but at the same time it is necessary for that molecule be soluble, have limited off-target effects, and be synthetically accessible. While an expert medicinal chemist might intuitively consider each of these additional constraints when designing new molecules, generative models, which aim to discover diverse and novel sets of molecules, will attempt to explore chemical space thoroughly. This means that any secondary objectives or constraints that have not been captured in the modeling approach will be ignored, reducing the model's usefulness in discovering new molecules. Therefore, it is critical to consider how the relevant secondary objectives will be captured when applying generative models to real molecular discovery scenarios. Additional common secondary objectives include toxicity, stability (often with respect to temperature, light, and/or time), phase behavior, solubility, or corrosivity.

There are two types of approaches that one can use to account for multiple objectives in generative molecular design: explicit multi-objective optimization and structure-guided multi-objective optimization, shown in Figure 2.

3.1.1 | Explicit multi-objective optimization

Explicit multi-objective optimization involves explicitly defining and optimizing over every property that is relevant to the application. This is most often accomplished by considering each property using a separate predictive

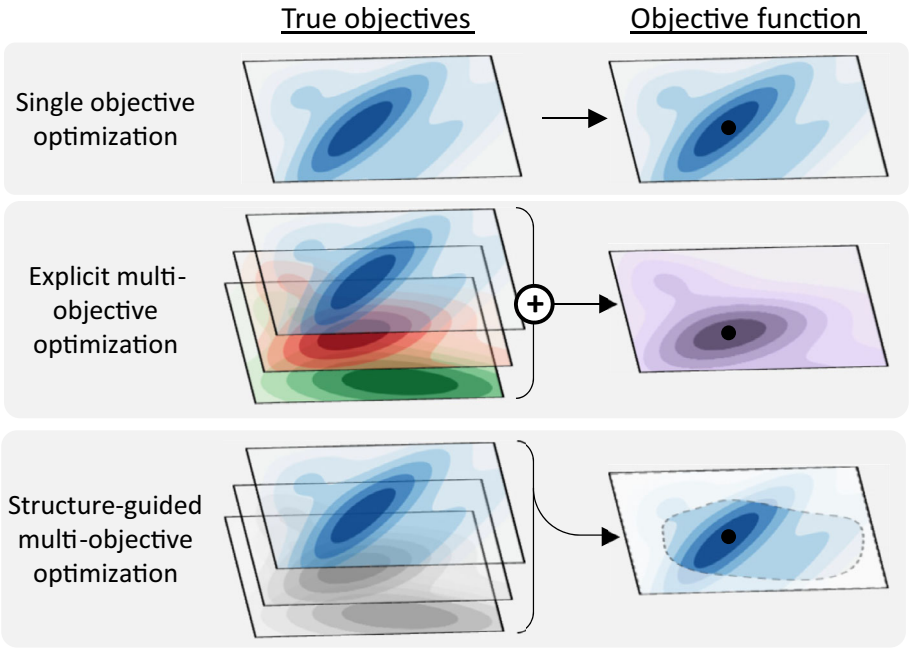

FIGURE 2 Comparison of single objective, explicit multi-objective, and structure-guided multi-objective optimization. Explicit multi-objective optimization involves combining multiple objectives (colored) to obtain an objective function, while structure-guided optimization involves implicitly considering secondary objectives (gray) by limiting the search space

model or evaluator[80–82] and optimizing over an objective function that combines these properties. Alternatively, if the property can be attributed to a specific scaffold within the molecules, the molecule can be optimized by incorporating the relevant scaffolds into the generative procedure.[70,83] By explicitly defining each objective, the user can directly control the relative importance of each objective, allowing this approach to be easily tailored to different applications.

One challenge with this approach, however, is that it requires an automated way for predicting or obtaining every relevant property without human intervention. This can become problematic for properties with smaller public datasets such as toxicity or phase behavior, for which it might not be feasible to train high accuracy predictive models. Additionally, it is difficult to train predictive models for properties that are hard to define. To illustrate this, consider the case where we would like to evaluate whether a molecule is synthetically feasible in an automated fashion. While it might seem like a good approach to simply determine whether or not a synthetic pathway can be found for that molecule using computer aided synthesis planning (CASP), this strategy does not consider how many byproducts might form or how difficult it might be to characterize or separate those byproducts. In this way, fully evaluating synthetic feasibility requires considering a complex combination of a number of factors, making it hard to predict in an automated fashion.

3.1.2 | Structure-guided multi-objective optimization

Structure-guided multi-objective optimization *implicitly* holds some properties constant using structural constraints. As described earlier, this can be accomplished via candidate-based[67,84] or scaffold-based optimization,[68–71,73,85–88] in which previously identified starting candidate molecules or scaffolds with desirable properties are improved with respect to one or more additional properties. In the case of the previously described drug design problem, we might begin with a candidate that is soluble and can be easily synthesized but is only moderately efficacious. In this case, one might train the generative model to improve drug efficacy with the goal of maintaining each of the other desirable properties. Unlike explicit multi-objective optimization, the user does not need to define each of the secondary objectives, making it straightforward to consider difficult-to-define objectives. Such implicit multi-objective optimization can also be accomplished by simply limiting exploration in chemical space to molecules that are similar to molecules used in training,[16,75] though this does limit the diversity of the generated molecules.

3.2 | Molecules designed using generative models must be synthesizable

One of the most universal requirements of molecules discovered for real applications is that they must be synthesizable. The most straightforward way to consider synthesizability in a generative modeling problem is to treat it explicitly as an additional objective. As described earlier, this requires an automated approach for evaluating synthesizability without human intervention. While CASP tools (which predict synthetic pathways for a given molecule) can be used to approximate this, they are often too computationally expensive to be used directly for explicit optimization. Instead, a number of scores have been proposed to quickly evaluate synthesizability in an automated fashion.

One of the earliest examples of such a score is called the synthetic accessibility (SA) score proposed by Ertl and Schuffenhauer,[89] which estimates synthesizability by penalizing molecules containing nonstandard structural features and prioritizing molecules with features that resemble representative molecules from PubChem. Later, Fukunishi et al.[90] further developed this score by expanding the set of representative molecules to include ChEMBl, ZINC, and LigandBOX. More recently, Coley et al.[91] proposed the synthetic complexity (SC) score, which estimated with the expected number of reaction steps required for synthesis. Finally, Thakkar et al.[92] recently presented the retrosynthetic accessibility (RA) score, which estimates synthesizability by using a machine learning model trained based on CASP predictions, thereby circumventing the computational expense of directly using a CASP model.

Importantly, each of these scores is only an approximate measure of synthetic feasibility and, therefore, each measurement has advantages and disadvantages in the context of molecular discovery. The SA score relies on identifying commonly occurring fragments within a given molecule and can therefore be easily "tricked" by complex molecules containing many reasonable looking fragments. The SC score focuses on measuring synthetic complexity, rather than feasibility and so it has been shown to correlate better with the length of the predicted synthesis pathway, but poorly with feasibility (when benchmarked against CASP models).[93] The RA score directly predicts synthetic feasibility with respect to a CASP model as a function of the full molecular structure, and therefore performs better than the other scores when CASP models are used as a reference. It is important to note, however, because the RA score directly relies on a given CASP model, any shortcomings of that model will also be present in the RA score.[92] In this way, each of the currently developed scores capture some (but not all) aspects of synthetic feasibility. They can be useful for guiding molecular discovery using generative modeling but will likely still lead to some synthetically infeasible molecules.

One alternative to treating synthesizability as a secondary objective is to use a generative model that directly integrates CASP tools into the generation process.[58,59,94–97] For example, instead of simply generating individual molecular graphs, Bradshaw et al.[58] generates synthetic pathways and then optimizes the final product for specific properties using a hill-climbing algorithm. A related approach proposed by Gottipati et al.[97] uses Policy Gradient for Forward Synthesis (PGFS) wherein an agent learns to apply chemical reactions to commercially available molecules to explore chemical space. Reinforcement learning (RL) can then be used to guide this agent toward molecules that are optimized for specific properties. These approaches inherently generate a synthetic pathway concurrently with the optimized molecule making them well suited to enforcing synthesizability.

One caveat to this class of models is that they are closely integrated with CASP models, which can only approximately predict retrosynthetic pathways and, consequently, any limitations in the CASP model will also be present in the generative model. For example, if a certain class of reactions necessary for synthesizing a given class of molecules is missing from the CASP training data, it may not be possible for the generative model to find a pathway to any molecules in that class. In this instance, the model will deem that class of molecules synthetically infeasible, when they may, in fact, be important for the application. In this way, requiring that molecules must be discoverable via a CASP model may limit the diversity of chemistries accessible during generation.

3.3 | Generative modeling workflows rely on predictive models with limited certainty

As described previously, generative models rely on predictive models to optimize molecules for specific properties. The accuracy of these predictive models is limited by the size and accuracy of the training datasets. For properties for which the training data comes from simulations or experiments, the training data is likely to contain noise and/or bias, preventing the model from learning the true structure–property relationship. Additionally, for properties for which there is limited training data, the available data may not be sufficient to constrain the model close to the target function. In this way, it is important

to consider how generative models behave when paired with error-prone predictive models. For a more detailed discussion of uncertainty quantification in deep learning we refer the reader to other resources.[98–104]

The most straightforward way deal with error-prone predictive models in generative molecular discovery is to use Bayesian optimization methods.[13,23,33,63] Bayesian optimization methods replace discrete predictions with continuous probability distributions allowing the use of sampling functions that balance exploration and exploitation. For generative models that use discrete optimization, uncertainty must be accounted for through the use of external uncertainty quantification metrics such as ensemble variance[105] or Monte Carlo (MC) dropout.[106] In the absence of uncertainty handling, error-prone predictions can mislead generative models—which aim to discover novel and diverse molecules—to explore regions of chemical space that are poorly captured by the predictive model, leading to workflows that propose molecules that are predicted to be optimal but underperform when tested experimentally.[75]

In contrast, there may be cases where it is useful to deliberately explore regions of chemical space that are poorly understood by the predictive model.[107,108] This may be valuable for applications where it is more important to discover new molecules than to discover molecules that have a target set of properties. Additionally, this can be particularly valuable for active learning applications. Active learning is an experimental strategy that involves iteratively selecting new experiments to perform with the goal of minimizing the cost of acquiring training data while simultaneously maximizing the accuracy of the machine learned model. In an active learning setting, it would be valuable to develop a generative model that deliberately proposes poorly predicted molecules that can be tested experimentally and used as training data to improve predictor in additional training iterations. In the continuous optimization case, this can easily be accomplished by adjusting the Bayesian optimization sampling function to emphasize exploration. This can also be accomplished in either continuous or discrete optimization by treating uncertainty as an explicit optimization objective.

3.4 | Common applied workflows for generative molecular design

As described earlier, there are a few examples of using generative models in real molecular discovery campaigns. The most straightforward way to use generative models in a real application is to train a generator and predictor model on an initial dataset, to discover a molecule or set of molecules, and to select a small number of molecules for experimental testing (Figure 3, top). This approach has been powerful for designing dyes that absorb light at target wavelengths[109] as well as inhibitors for Janus kinase 3,[110] the discoidin domain receptor,[111] and a series of receptors associated with neuropsychiatric function.[112]

One caveat to this approach, however, is that the low throughput nature of the experiments prevents extensive experimental testing (each of the described studies only includes experimentally testing four or fewer generated molecules for a given design target). Each generative model can often propose hundreds or thousands of lead molecules and the molecules explored experimentally are often manually selected based on additional secondary properties. This manual selection process can lead to final compounds that closely resemble the training set.[113] In this way, using a generative model in a discovery campaign does not guarantee that a novel lead candidate can be straightforwardly identified in an automated fashion. Rather, generative models can be used to propose large sets of candidate molecules that can be manually down-selected to promising leads.

In addition to using generative models to simply discover molecules, one can use the data generated from promising leads to retrain the predictor and generator models, thus "closing the loop" on molecular discovery (Figure 3, bottom). This approach can be valuable for applications where the predictive model has been trained on a limited dataset. While there are no examples of using generative modeling for experimental closed-loop discovery of molecules, Chen and Gu[114] recently illustrated the value of this approach for computationally designing materials with desirable mechanical properties. For this application, they circumvented the need to synthesize materials by measuring properties using the finite element method (FEM) a relatively inexpensive computational approach. While such computational methods are often less accurate than experimental methods, they provide opportunities to develop closed-loop generative modeling approaches on model computational systems that can be later translated to experimental systems.

A key advance required to achieve closed loop experimental molecular discovery is the development of high throughput experimental platforms capable of synthesizing and testing molecules. To this end, a number of platforms have been developed that vary in terms of the flexibility of the synthesis and testing operations that they can perform. Ultimately, the aim is for these platforms to be coupled with machine learning approaches to yield self-driving

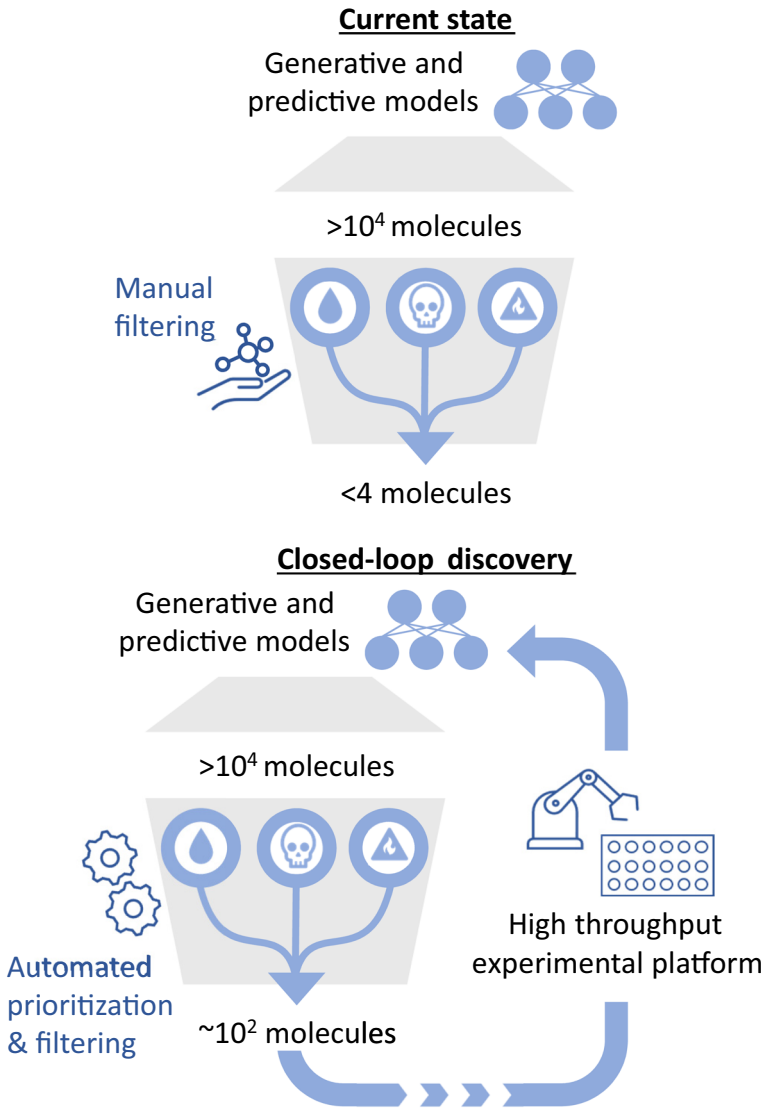

FIGURE 3 Currently, in experimental applications, generative models have been used in single-pass workflows with manual filtering. In the future, we are likely to see the integration generative models into closed-loop, autonomous discovery pipelines

laboratories for molecular discovery. This advance will require integrating (1) generative models capable of molecular discovery, (2) retrosynthesis models capable of predicting viable synthetic pathways, and (3) fully automated experimental platforms capable of flexibly carrying out synthesis and testing operations. While such a system has not been achieved, such a vision has been described elsewhere, and will likely come to fruition in the near future. For more information about autonomous experimental platforms, we refer the reader to these excellent reviews articles.[115,116]

4 | FUTURE DIRECTIONS

4.1 | The future of benchmarks

As with many areas of machine learning, the trajectory of generative modeling has been historically governed by the key benchmarks in the field. The existing set of benchmarks have been critical in guiding a wave of increasingly powerful generative models for molecular discovery, facilitating improvements along metrics such as validity, uniqueness,

and diversity of generated molecules. Despite this, as others have suggested,[117] many of the newest generative models perform very well across these benchmarks, indicating that these benchmarks are no longer sufficient to evaluate which models perform best. Importantly, these high-performing models do not necessarily have all of the attributes needed to be directly employed in a real molecular discovery application, an observation that is supported by the relatively few examples of real applications in the literature. Thus, there is a need to develop more meaningful benchmarks to facilitate the development of generative models that are more ideally designed to solve real world problems.

One of the challenges associated with developing such a set of benchmarks is that generative modeling criteria are likely to vary from one application to another. An ideal set of benchmarks is one that contains metrics relevant to a diverse range of applications while also encompassing most of the hurdles associated with using generative models for molecular discovery. We expect that this set of benchmarks will include factors such as synthetic feasibility, safety and handling, uncertainty quantification, and other considerations relevant to deploying generative models for real applications.

4.2 | The future of representations

The past few years have seen many developments in generative models capable of learning from 2D molecular graphs and 3D point clouds. These advances are valuable because they enable generative models to learn on more information-rich representations that contain more information about the underlying physics of the molecule.[118] But there are still technical challenges associated with using these frameworks. Most 2D and 3D generative models are auto regressive, meaning that they generate the graph or point cloud in an ordered fashion. Because molecules are unordered by nature, models that impose a canonical ordering among atoms introduce additional bias into generative models, potentially limiting their performance. In addition, most representations have been developed for small molecules and cannot be easily used to describe large or chiral molecules. Finally, generative models that leverage 2D and 3D representations are more computationally expensive to train than simpler models, making them more difficult to scale to larger molecules and/or larger datasets.

4.3 | Final thoughts

Within the last decade, deep generative modeling for molecular discovery has grown into a well-developed field. Generative modeling holds great promise as an automated approach for discovering novel chemistries capable of solving many global crises. While many advances are still needed for generative modeling to achieve its full potential, we anticipate that the current challenges will be solved in the coming years.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

Camille Bilodeau: Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); validation (equal); writing – original draft (equal); writing – review and editing (equal). **Wengong Jin:** Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); writing – original draft (equal); writing – review and editing (equal). **Tommi Jaakkola:** Conceptualization (equal); funding acquisition (equal); methodology (equal); project administration (equal); resources (equal); validation (equal); writing – review and editing (equal). **Regina Barzilay:** Conceptualization (equal); funding acquisition (equal); methodology (equal); project administration (equal); resources (equal); supervision (equal); validation (equal); writing – review and editing (equal). **Klavs Jensen:** Conceptualization (equal); funding acquisition (lead); methodology (equal); project administration (equal); resources (equal); supervision (equal); writing – review and editing (equal).

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Klavs F. Jensen https://orcid.org/0000-0001-7192-580X

RELATED WIRES ARTICLES

Advances and challenges in deep generative models for de novo molecule generation

REFERENCES

1. Polishchuk PG, Madzhidov TI, Varnek A. Estimation of the size of drug-like chemical space based on GDB-17 data. J. Comput. Aided. Mol. Des. 2013;27(8):675–679.
2. Reymond J-L, Awale M. Exploring chemical space for drug discovery using the chemical universe database. ACS Chem Nerosci. 2012; 3:649–57.
3. Polykovskiy D, Zhebrak A, Sanchez-Lengeling B, Golovanov S, Tatanov O, Belyaev S et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. Front Pharmacol. 2020;11:565644.
4. Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: benchmarking models for de novo molecular design. J Chem Inf Model. 2019;59(3):1096–1108.
5. Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. Science. 2018;361:360–5.
6. Xue D, Gong Y, Yang Z, Chuai G, Qu S, Shen A et al. Advances and challenges in deep generative models for de novo molecule generation. WIREs Comput Mol Sci. 2019;9:e1395.
7. Elton DC, Boukouvalas Z, Fuge MD, Chung PW. Deep learning for molecular design—a review of the state of the art. Mol Syst Des Eng. 2019;4:828–49.
8. Schwalbe-Koda D, Gómez-Bombarelli R. Generative models for automatic chemical design. In: Schutt KT, Chmiela S, von Lilienfeld O, Tkatchenko A, Tsuda K, Müller K-R, editors. Machine learning meets quantum physics. Cham: Springer International Publishing; 2020. p. 445–67.
9. Vanhaelen Q, Lin Y-C, Zhavoronkov A. The advent of generative chemistry. ACS Med Chem Lett. 2020;11:1496–505.
10. Alshehri AS, Gani R, You F. Deep learning and knowledge-based methods for computer-aided molecular design—toward a unified approach: state-of-the-art and future directions. Comput Chem Eng. 2020;141:107005.
11. Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL. Quantifying the chemical beauty of drugs. Nat Chem. 2012;4:90–8.
12. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Model. 1988;28:31–6.
13. Gómez-Bombarelli R, Wei J, Duvenaud D, Hernandez-Lobato J-M, Sanchez-Lengeling B, Sheberla D et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent Sci. 2018;4:268–76.
14. Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular de-novo design through deep reinforcement learning. J Chem. 2017;9:48.
15. Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. Sci Adv. 2018;4:eaap7885.
16. Kotsias P-C, Arus-Pous J, Chen H, Engkvist O, Tyrchan C, Bjerrum EJ. Direct steering of de novo molecular generation using descriptor conditional recurrent neural networks (cRNNs). Nat Mach. 2020;2:254–65.
17. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent Sci. 2018;4:120–31.
18. Kang S, Cho K. Conditional molecular design with deep generative models. J Chem Inf Model. 2019;59:43–52.
19. Alperstein Z, Cherkasov A, Rolfe JT. All SMILES variational autoencoder. 2019. arXiv preprint arXiv:1905.13343.
20. Arús-Pous J, Patronov A, Bjerrum EJ, Tyrchan C, Reymond J-L, Chen H et al. SMILES-based deep generative scaffold decorator for de-novo drug design. J Chem. 2020;12:38.
21. Sepp H, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80.
22. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. arXiv preprint arXiv:1412.3555.
23. Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. ICML. 2017;70:1945–1954.
24. Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for structured data. 2018. arXiv preprint arXiv: 1802.08786.
25. Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. Mach Learn: Sci Technol. 2020;1:045024.
26. De Cao N, Kipf T. MolGAN: an implicit generative model for small molecular graphs. 2018. arXiv preprint arXiv:1805.11973.
27. Ma T, Chen J, Xiao C. Constrained generation of semantically valid graphs via regularizing Variational autoencoders. 2018. arXiv preprint arXiv:1809.02630.
28. Simonovsky M, Komodakis N. GraphVAE: towards generation of small graphs using variational autoencoders. ICANN. 2018;11139: 412–422.
29. You J, Liu B, Ying R, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. 2018. arXiv preprint arXiv:1806.02473.
30. Li Y, Vinyals O, Dyer C, Pascanu R, Battaglia P. Learning deep generative models of graphs. arXiv:1803.03324 [cs.LG]. 2018.

31. Samanta B, De A, Jana G, Chattaraj PK, Ganguly N, Rodriguez MG. NeVAE: a deep generative model for molecular graphs. AAAI. 2019;33:1110–7.
32. Liu Q, Allamanis M, Brockschmidt M, Gaunt A. Constrained graph variational autoencoders for molecule design. NeurIPS. 2018;31: 7632–7642.
33. Jin W, Barzilay R, Jaakkola T. Junction tree Variational autoencoder for molecular graph generation. ICML. 2018;80:2323–2332.
34. Jin W, Barzilay R, Jaakkola T. Hierarchical generation of molecular graphs using structural motifs. ICML. 2020;119:4839–4848.
35. Gebauer NWA, Gastegger M, Schütt KT. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. NeurIPS. 2019;32:8600–8610.
36. Simm GNC, Pinsler R, Hernández-Lobato JM. Reinforcement learning for molecular design guided by quantum mechanics. ICML. 2020;119:8959–8969.
37. Hawkins PCD, Skillman AG, Warren GL, Ellingson BA, Stahl MT. Conformer generation with OMEGA: algorithm and validation using high quality structures from the protein databank and Cambridge structural database. J Chem Inf Model. 2010;50:572–84.
38. Riniker S. Molecular dynamics fingerprints (MDFP): machine learning from MD data to predict free-energy differences. J Chem Inf Model. 2017;57:726–41.
39. Simm GNC, Hernández-Lobato JM. A generative model for molecular distance geometry. ICML. 2020;119:8949–8958.
40. Xu M, Luo S, Bengio Y, Peng J, Tang J. Learning neural generative dynamics for molecular conformation generation. 2021. arXiv preprint arXiv:2102.10240.
41. Ganea O-E, Pattanaik L, Coley CW, Barzilay R, Jensen KF, Green WH et al. GeoMol: torsional geometric generation of molecular 3D conformer ensembles. 2021. arXiv preprint arXiv:2106.07802.
42. Axelrod S & Gomez-Bombarelli R GEOM: energy-annotated molecular conformations for property prediction and molecular generation; 2020. arXiv:2006.05531 [physics].
43. Ramakrishnan R, Dral PO, Rupp M, von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. Sci Data. 2014;1:140022.
44. Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. J Am Stat Assoc. 2017;112:859–77.
45. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S et al. Generative adversarial networks. NeurIPS. 2014;27: 3276–3285.
46. Kingma DP, Welling M. Auto-encoding variational bayes. 2014. arXiv preprint arXiv:1312.6114.
47. Kwon Y, Yoo J, Choi Y-S, Son W-J, Lee D, Kang S. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. J Chem. 2019;11:70.
48. Assouel R, Ahmed M, Segler MH, Saffari A, Bengio Y DEFactor: differentiable edge factorization-based probabilistic graph generation; 2018. arXiv:1811.09766 [cs].
49. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. 2018. arXiv preprint arXiv: 1809.11096.
50. Pölsterl S, Wachinger C. Adversarial learned molecular graph inference and generation. ECML. 2019;33(5):173–189.
51. Prykhodko O, Johansson S. V, Kotsias P.-C, Arus-Pous J, Bjerrum E. J, Engkvist O et al. A de novo molecular generation method using latent vector based generative adversarial network. J Chem. 2019;11:74.
52. Rivas P, Guarino M, Shah A. DiPol-GAN: Generating Molecular Graphs Adversarially with Relational Differentiable Pooling. https://www.rivas.ai/pdfs/guarino2019dipol.pdf
53. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models; 2018. arXiv:1705.10843 [cs, stat].
54. Zang C, Wang F. MoFlow: an invertible flow model for generating molecular graphs. KDD. 2020;22:617–626.
55. Shi C, Xu M, Zhu Z, Zhang W, Zhang M, Tang J. GraphAF: a flow-based autoregressive model for molecular graph generation. 2020. arXiv preprint arXiv:2001.09382.
56. Madhawa K, Ishiguro K, Nakago K, Abe M GraphNVP: an invertible flow model for generating molecular graphs; 2019. arXiv: 1905.11600 [cs, stat].
57. Shi C, Luo S, Xu M, Tang J. Learning gradient fields for molecular conformation generation. ICML. 2021;139:9558–9568.
58. Bradshaw J, Paige B, Kusner MJ, Segler MHS, Hernández-Lobato JM. Barking up the right tree: an approach to search over molecule synthesis DAGs. NeurIPS. 2020;33:6852–6866.
59. Bradshaw J, Paige B, Kusner MJ, Segler MHS, Hernández-Lobato JM. A model to search for synthesizable molecules. NeurIPS. 2019; 32:12000–12010.
60. Mendez D, Gaulton A, Bento A. P, Chambers J, De Veji M, Felix E et al. ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Res. 2019;47:D930–40.
61. Irwin JJ, Shoichet BK. ZINC—a free database of commercially available compounds for virtual screening. J Chem Inf Model. 2006; 45(1):177–82.
62. Zhou Z, Kearnes S, Li L, Zare RN, Riley P. Optimization of molecules via deep reinforcement learning. Sci Rep. 2019;9:10752.
63. Griffiths R-R, Hernández-Lobato JM. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. Chem Sci. 2020;11:577–86.
64. Nigam A, Friederich P, Krenn M, Aspuru-Guzik A. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. 2020. arXiv preprint arXiv:1909.11655.

65. Ahn S, Kim J, Lee H, Shin J. Guiding deep molecular optimization with genetic exploration. NeurIPS. 2020;33:12008–12021.
66. Besnard J, Ruda GF, Setola V, Abecassis K, Huang X-P, Norval S et al. Automated design of ligands to polypharmacological profiles. Nature. 2013;492(7428):215–220.
67. Jin W, Yang K, Barzilay R, Jaakkola T. Learning multimodal graph-to-graph translation for molecular optimization. 2019. arXiv preprint arXiv:1812.01070.
68. Langevin M, Minoux H, Levesque M, Bianciotto M. Scaffold-constrained molecular generation. J Chem Inf Model. 2020;60(12): 5637–46.
69. Li Y, Hu J, Wang Y, Zhou J, Zhang L, Liu Z. DeepScaffold: a comprehensive tool for scaffold-based de novo drug discovery using deep learning. J Chem Inf Model. 2020;60:77–91.
70. Jin W, Barzilay R, Jaakkola T. Multi-objective molecule generation using interpretable substructures. ICML. 2020;119:4849–4859.
71. Podda M, Bacciu D, Micheli A. A deep generative model for fragment-based molecule generation. AISTATS. 2020;108:2240–2250.
72. Imrie F, Bradley AR, van der Schaar M, Deane CM. Deep generative models for 3D linker design. J Chem Inf Model. 2020;60:1983–95.
73. Green H, Koes DR, Durrant JD. DeepFrag: a deep convolutional neural network for fragment-based lead optimization. Chem Sci. 2021; 12:8036–47.
74. Zhang J, Mercado R, Engkvist O, Chen H. Comparative study of deep generative models on chemical space coverage. J Chem Inf Model. 2021;61(6):2572–81.
75. Renz P, Van Rompaey D, Wegner JK, Hochreiter S, Klambauer G. On failure modes in molecule generation and optimization. Drug Discov Today Technol. 2019;32–33:55–63.
76. Walters P rd_filters. https://github.com/PatWalters/rd_filters. Accessed 10 Sep 2021.
77. Degen J, Wegscheid-Gerlach C, Zaliani A, Rarey M. On the art of compiling and using 'drug-like' chemical fragment spaces. Chem Med Chem. 2008;3:1503–7.
78. Bemis GW, Murcko MA. The properties of known drugs. 1. Molecular frameworks. J Med Chem. 1996;39:2887–93.
79. Preuer K, Renz P, Unterthiner T, Hochreiter S, Klambauer G. Fréchet ChemNet distance: a metric for generative models for molecules in drug discovery. J Chem Inf Model. 2018;58:1736–41.
80. Fu T, Xiao C, Li X, Glass LM, Sun J. MIMOSA: multi-constraint molecule sampling for molecule optimization. 2021. arXiv preprint arXiv:2010.02318.
81. He C, Huang S, Cheng R, Tan KC, Jin Y. Evolutionary multiobjective optimization driven by generative adversarial networks (GANs). IEEE Trans Cybern. 2021;51:3129–42.
82. Domenico A, Nicola G, Daniela T, Fulvio C, Nicola A, Orazio N. De novo drug design of targeted chemical libraries based on artificial intelligence and pair-based multiobjective optimization. J Chem Inf Model. 2020;60:4582–93.
83. Xie Y, Shi C, Zhou H, Yang Y, Zhang W, Yu Y et al. MARS: Markov molecular sampling for multi-objective drug discovery. 2021. arXiv preprint arXiv:2103.10432.
84. Shin B, Park S, Bak J, Ho JC. Controlled molecule generator for optimizing multiple chemical properties. CHIL. 2021;1:146–153.
85. Lim J, Hwang S-Y, Kim S, Moon S, Kim WY. Scaffold-based molecular design using graph generative model. Chem Sci. 2020;11: 1153–64.
86. 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;11:8312–22.
87. Polishchuk P. CReM: chemically reasonable mutations framework for structure generation. J Chem. 2020;12:28.
88. Chen Z, Min MR, Parthasarathy S, Ning X. Molecule optimization via fragment-based generative models. Nat Mach. 2021;3:1040–9.
89. Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. J Chem. 2009;1:8.
90. Fukunishi Y, Kurosawa T, Mikami Y, Nakamura H. Prediction of synthetic accessibility based on commercially available compound databases. J Chem Inf Model. 2014;54:3259–67.
91. Coley CW, Rogers L, Green WH, Jensen KF. SCScore: synthetic complexity learned from a reaction corpus. J Chem Inf Model. 2018;58: 252–61.
92. Thakkar A, Chadimova V, Bjerrum EJ, Engkvist O, Reymond J-L, et al. Retrosynthetic accessibility score (RAscore)—rapid machine learned synthesizability classification from AI driven retrosynthetic planning. Chem Sci. 2021;12:3339–3349.
93. Gao W, Coley CW. The synthesizability of molecules proposed by generative models. J Chem Inf Model. 2018;58:252–61.
94. Button A, Merk D, Hiss JA, Schneider G. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. Nat Mach. 2019;1:307–15.
95. Korovina K, Xu S, Kandasamy K, Neiswanger W, Poczos B, Schneider J, et al. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. AISTATS. 2019;108:3393–3403.
96. Horwood J, Noutahi E. Molecular design in synthetically accessible chemical space via deep reinforcement learning. ACS Omega. 2020;5:32984–94.
97. Gottipati SK, Sattarov B, Niu S, Pathak Y, Wei H, Liu S, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. PMLR. 2020;119:3668–3679.
98. Mervin LH, Johansson S, Semenova E, Giblin KA, Engkvist O. Uncertainty quantification in drug design. Drug Discov Today. 2021; 26(2):474–89.

99. Walters WP, Barzilay R. Applications of deep learning in molecule generation and molecular property prediction. Acc Chem Res. 2021; 54:263–70.
100. Scalia G, Grambow CA, Pernici B, Li Y-P, Green WH. Evaluating scalable uncertainty estimation methods for deep learning based molecular property prediction. J Chem Inf Model. 2020;60(6):2697–717.
101. Hie B, Bryson B, Berger B. Learning with uncertainty for biological discovery and design. Cell Syst. 2020;11:461–77.
102. Ryu S, Kwon Y, Kim WY Uncertainty quantification of molecular property prediction with Bayesian neural networks; 2019. arXiv: 1903.08375 [cs, stat].
103. Ovadia Y, Fertig E, Ren J, Rado Z, Sculley D, Nowozin S, et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. 2019. arXiv preprint arXiv:1906.02530.
104. Hirschfeld L, Swanson K, Yang K, Barzilay R, Coley CW. Uncertainty quantification using neural networks for molecular property prediction. J Chem Inf Model. 2020;60(8):3770–80.
105. Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. 2017. arXiv preprint arXiv:1612.01474.
106. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. ICML. 2016;48:1050–1059.
107. Thiede LA, Krenn M, Nigam A, Aspuru-Guzik A. Curiosity in exploring chemical space: intrinsic rewards for deep molecular reinforcement learning. 2020. arXiv preprint arXiv:2012.11293.
108. Shen C, Krenn M, Eppel S, Aspuru-Guzik A. Deep molecular dreaming: inverse machine learning for de-novo molecular design and interpretability with surjective representations. Mach Learn Sci Technol. 2021;2:3.
109. Sumita M, Yang X, Ishihara S, Tamura R, Tsuda K. Hunting for organic molecules with artificial intelligence: molecules optimized for desired excitation energies. ACS Cent Sci. 2018;4:1126–33.
110. Polykovskiy D, Zhebrak Z, Vetrov D, Ivanenkov Y, Aladinskiy V, Mamoshina P et al. Entangled conditional adversarial autoencoder for de novo drug discovery. Mol Pharm. 2018;15(10):4398–4405.
111. Zhavoronkov A. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat Biotechnol. 2019;37:1038–40.
112. Tan X, Jiang X, He Y, Zhong F, Li X, Li Z et al. Automated design and optimization of multitarget schizophrenia drug candidates by deep learning. Eur J Med Chem. 2020;204:112572.
113. Walters P. Assessing the impact of generative AI on medicinal chemistry. Nat Biotechnol. 2020;38:142–50.
114. Chen C-T, Gu GX. Generative deep neural networks for inverse materials design using backpropagation and active learning. Adv Sci. 2020;7:1902607.
115. Dimitrov T, Kreisbeck C, Becker JS, Aspuru-Guzik A, Saikin SK. Autonomous molecular design: then and now. ACS Appl Mater Interfaces. 2019;11:24825–36.
116. Soldatov M. A, Butova V. V, Pashkov D, Butakova M. A, Medvedev P. V, Chernov A. V et al. Self-driving laboratories for development of new functional materials and optimizing known reactions. J Nanomater. 2021;11:619.
117. Nigam A, Pollice R, Krenn M. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. Chem Sci. 2021;12:7079–7090.
118. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys. 2021;3:422–40.

How to cite this article: Bilodeau C, Jin W, Jaakkola T, Barzilay R, Jensen KF. Generative models for molecular discovery: Recent advances and challenges. WIREs Comput Mol Sci. 2022;12:e1608. https://doi.org/10.1002/wcms.1608