



Perspective

Exploring chemical space — Generative models and their evaluation

Martin Vogt

Department of Life Science Informatics, b-it, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich Wilhelms-Universität, Friedrich-Hirzebruch-Allee, 5-6, Bonn 53115, Federal Republic of Germany



ARTICLE INFO

Keywords:

Artificial intelligence
Chemical space
Chemical space exploration
Deep neural networks
Generative models
Inverse QSAR/QSPR

ABSTRACT

Recent advances in the field of artificial intelligence, specifically regarding deep learning methods, have invigorated research into novel ways for the exploration of chemical space. Compared to more traditional methods that rely on chemical fragments and combinatorial recombination deep generative models generate molecules in a non-transparent way that defies easy rationalization. However, this opaque nature also promises to explore uncharted chemical space in novel ways that do not rely on structural similarity directly. These aspects and the complexity of training such models makes model assessment regarding novelty, uniqueness, and distribution of generated molecules a central aspect. This perspective gives an overview of current methodologies for chemical space exploration with an emphasis on deep neural network approaches. Key aspects of generative models include choice of molecular representation, the targeted chemical space, and the methodology for assessing and validating chemical space coverage.

1. Introduction

The chemical space (CS) of small molecules refers to the universe of all conceivable chemically stable small molecules [1]. Combinatorial estimates put its size in excess of 10^{60} molecules [2]. This estimate can easily vary by arbitrary orders of magnitude depending on the definition of what constitutes a “small” molecule. However, the main significance of this number is that it is so large that (a) only an almost infinitesimal fraction of CS can ever be realized, synthesized, and explored *in vitro* or *in vivo* and likewise (b) only a “slightly” larger infinitesimal fraction can be explored *in silico* regardless of the computational resources available now or in the future. Generative models (GMs) aim to explore CS not by exhaustive enumeration but by randomly sampling and exploring its properties. “Generative” refers to the central aspect of such models aiming to learn probability distributions from training data, which are then used to guide the generation of novel molecules.

Small molecule CS can be described algorithmically by using combinatorial approaches that systematically explore chemical structural graphs of increasing size. While exhaustive enumeration in this way is infeasible in general, it has been realized for molecules of limited size (up to 11, 13, and 17 heavy atoms) and limited atom types (C, N, O, S, and halogens) yielding CSs of 26.4 million, 977 million, and 166 billion molecules, respectively [3–5]. The numbers resulting from these impressive efforts demonstrate their exponential growth and the ultimate futility of extending this approach to even just moderately larger molecules. Thus, beyond a certain size such exhaustive efforts have to

be replaced by exploratory efforts that will only be able to sample a tiny fraction of CS under consideration.

The creation of arbitrary chemical structures exploring CS is, in principle at least, “easy” as any random chemical graph structure that follows basic valence rules can be considered a legal molecule. As such, a simple brute-force approach to sampling CS can be envisioned that (1) generates random mathematical graphs (of limited size) (2) labels vertices and edges randomly with atom types and bond orders, and (3) filters the resulting chemical graph structures to remove chemically nonsensical structures. Alternatively, and more closely reflecting the approach taken by many current deep learning approaches, the random graph generation process can be substituted by the generation of random strings to be interpreted as linear representations of molecules like SMILES [6]. Such a simplistic approach has its analogy in the figurative monkey randomly hitting letters on the keyboard of a typewriter as popularized by the infinite monkey theorem [7]. This theorem, when applied to CS guarantees that it is in principle possible to exhaustively explore CS this way because any molecule has a certain (although minimal) probability of being generated. At the same time, the process outlined above is not practical because the probability of generating valid molecular representations is extremely low. However, if the random generation process can be directed to mainly generate valid molecular representations, and probabilities can be adjusted to focus on reasonable chemical structures, the outlined workflow is not only feasible, but is also the principle on which most current GMs rely.

While the ability to “randomly generate valid chemical structures” is a prerequisite for a successful GM it is not a meaningful criterion to

E-mail address: martin.vogt@bit.uni-bonn.de

<https://doi.org/10.1016/j.ailsci.2023.100064>

Received 30 December 2022; Received in revised form 1 February 2023; Accepted 2 February 2023

Available online 4 February 2023

2667-3185/© 2023 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

judge its quality *per se*. For instance, for generic CS exploration it would be preferable that every molecule of the relevant CS has roughly the same chance of being generated or the bias towards certain molecules is at least reasonable (e.g., larger molecules are less likely to be generated than smaller ones). On the other hand, often CS is explored with certain goals in mind. Just as *in vitro* and *in vivo* exploration of CS in lead optimization campaigns aims to optimize endpoints like potency or ADMET properties [8], computational CS exploration can be conducted with respect to certain goals focusing on molecules with favorable properties for endpoints of interest [9].

Lead optimization campaigns usually focus on exploring CS around one or a few scaffolds. However, computational approaches are not limited in this way and can emphasize exploration based on relevant physicochemical and biological properties.

Thus, to assess the usefulness of GMs the following aspects need to be considered:

- Validity of generated representations.
- Novelty of generated molecules.
- Synthetic accessibility and drug-likeness of generated molecules.
- Extent to which the covered CS is representative and relevant for the intended purpose.

Some of these aspects like validity and novelty can be easily quantified using statistical metrics. Furthermore, computational metrics like the synthetic accessibility score [10] and the quantitative estimate of drug-likeness [11] can give indications how reasonable and practical proposed structures are. The degree to which a CS is representative can be assessed by determining distributions of structural and physicochemical quantities, while the biological relevance of a CS is characterized by the extent to which it is enriched with molecules possessing favorable properties like a target-specific activity. For GMs targeting a focused CS, e.g., CS containing molecules with specific biological activities or synthetically accessible CS, traditional benchmark approaches are not applicable as the CS to be covered by GMs can be extremely large. Thus, it should not be expected that a GM is able to recreate molecules from a hold-out test set. On the contrary, recreation of known molecules might be an indication that the CS covered is relatively small and overpopulated with molecules from already explored CS.

GMs that predate the deep learning revolution of the last years mainly used fragment-based approaches and by recombining fragments ensured that the randomly generated structures (after filtering) represent reasonable molecules [12–14]. By design, the CS covered by such GMs is clearly defined (and, at the same time, limited) by the fragment libraries used. This might guarantee that the CS covered is representative regarding structural composition, for instance if a fragment library was derived from a reference set of relevant molecules, but this does not (necessarily) extend to biological properties.

One of the expected advantages of CS exploration using deep neural networks (DNNs) in so-called deep GMs is their potential for abstracting from a given structural context and generate and explore CS not based on structural similarity alone but instead based on multiple parameters that are relevant for endpoints to be optimized [15,16]. In essence, this is also the central question of the inverse QSAR/QSPR problem [17–20] and is one of the drawing powers of applying advanced artificial intelligence (AI) methods to the exploration of CS. Architectures of DNNs do not incorporate any chemical knowledge, all of which has to be acquired during training and is encoded in the learnable parameters of the network. However, this aspect also makes it harder to assess whether a trained DNN fulfills these expectations. This perspective extends a previous review on CS exploration methodologies [21] by additionally focusing on the problem of assessing and validating CS covered by GMs.

2. Molecular graph representations for generative models

The representation of molecules and their processing in GMs play an important role in how CS is covered. In this regard, a crucial distinction

between more traditional fragment-based methods and deep learning methods can be observed. In fragment-based methods, fragments represent chemically sound substructures and recombinations and modifications are chemically meaningful. In this case, the molecular representation itself (SMILES-based or graph-based) plays a minor role because the ways these representations are manipulated are based on the underlying chemistry of the molecules. Deep learning models are agnostic of the semantics of the chosen representation and manipulation occurs on a purely syntactical level. A central step in the training of DNNs is to learn the grammar and rules of the representation so that valid representations can be generated. Furthermore, typical patterns in representations can be learned, which form syntactical “building blocks” of newly generated molecules and thus characterize the generated CS. However, if a DNN only learns the syntactical rules of a representation, generation of molecules will be dominated by the similarity of the representation, which only imperfectly correlates with molecular similarity. A recent study has shown indeed, that while deep GMs can successfully learn the grammar of a representation from relatively small training sets, larger sets might be required to more properly reflect physicochemical properties of the reference training set [22].

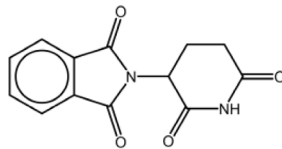
These considerations are most clearly reflected in models based on linear character-based representations. Although several deep GMs utilizing molecular graphs directly for input and output have been developed in recent years [23–26], language-based representations form the basis of most popular models. This is unsurprising as the rapid development of DNN architectures in recent years has been especially successful in signal processing and natural language processing [27].

The SMILES notation [6] is the most popular linear notation for chemical structures. It possesses a relatively simple grammar where parenthesized expressions are used to represent acyclic compounds. Additionally, matching numbers associated with pairs of atoms indicate bonds completing rings (see Fig. 1). It is the representation of choice for most generative DNN models to date. In one of the early publications using generative DNNs, Gómez-Bombarelli et al. [15] showed that DNNs can accurately learn the grammar of SMILES and can generate a high percentage of valid SMILES. However, training of the model essentially failed, when the grammatically much more complex InChI line notation [28] was used.

A single molecule can have many different SMILES notations. The notation is determined by the starting atom corresponding to the first token of the SMILES string and the order in which the atoms of the molecules are traversed in a depth-first traversal. Thus, the same molecule can have widely varying SMILES representations (see Fig. 1). While canonicalization algorithms, which assign priorities to the atoms of a molecule based on the molecular topology, can be used to obtain a unique representation for a molecule it is still unavoidable that some structurally similar molecules have significantly different canonical SMILES representations. Generally, it can be stated that similar SMILES represent similar molecules while the converse might only be true to a limited extent. This issue can be addressed during training of the models by augmenting the training set by randomizing or enumerating all possible SMILES representations of a molecule. A number of studies have shown that this has a positive effect on the performance [22,29,30].

Given the reasonable assumption that the more complex the representation, the harder it will be for a deep GM to generate valid representation, variations and alternatives to the SMILES notation have been developed aimed at making it easier to generate valid representations. The DeepSMILES notation [31] is a grammatical simplification of SMILES that does not require matching parentheses or correctly paired numbers for cyclic bonds (see Fig. 1). While this notation has not been widely adopted, studies have shown that the use of DeepSMILES can make it easier for a GM to generate valid representations although improvements were not considered major [22].

As an alternative linear representation, SELFIES [32] were designed with the aim that any combination of symbols can be interpreted as a valid chemical structure fulfilling basic topological and valence rules



Thalidomide

InChI	InChI=1S/C13H10N2O4/c16-10-6-5-9(11(17)14-10)15-12(18)7-3-1-2-4-8(7)13(15)19/h1-4,9H,5-6H2,(H,14,16,17)	
SMILES	canonical	<chem>O=C1CCC(N2C(=O)c3ccccc3C2=O)C(=O)N1</chem>
	randomized	<chem>c12c(C(=O)N(C1=O)C1CCC(NC1=O)=O)cccc2C1CC(=O)NC(=O)C1N1C(c2cccc2C1=O)=O</chem> <chem>c1ccc2C(N(C(c2c1)=O)C1CCC(NC1=O)=O)=O</chem>
DeepSMILES	canonical	<chem>O=CCCCNC(=O)cccccc6C9=O))))))C(=O)N6</chem>
	randomized	<chem>ccC(=O)NC5=O))CCCCNC6=O))))))cccc6CCC(=O)NC(=O)C6NCcccccc6C9=O))))))=O</chem> <chem>ccccCNCc5c9))=O))CCCCNC6=O))))))=O</chem>
SELFIES	canonical	<chem>[O]=[C][C][C][C][Branch2_1][Ring1][C][N][C][Branch1_2][C][=O][C][C][C][C][C][C][Ring1][Branch1_2][C][Ring1][Branch2_3][=O][C][Branch1_2][C][=O][N][Ring2][Ring1][C][C][C][Branch2_1][Ring1][Branch2_1][C][Branch1_2][C][=O][N][Branch1_1][Branch1_1][C][Ring1][Branch1_2][=O][C][C][C][C][Branch1_1][Branch1_2][N][C][Ring1][Branch1_2][=O][=O][C][C][C][C][Ring2][Ring1][Ring1]</chem>
	randomized	

Fig. 1. Molecular representations. For thalidomide, several linear representations are shown. Canonical and three randomized SMILES were generated using RDKit (<https://www.rdkit.org>). DeepSMILES and SELFIES were determined on the basis of the SMILES representations. Due to their length only one randomized SELFIES is shown.

(see Fig. 1). This ensures that any GM generating SELFIES, will, by design only produce valid molecules. In SELFIES, symbols have to be interpreted in a context-dependent manner, e.g., the same symbol can indicate a specific element, the size of a sidechain, or the size of a ring. The complex semantics of SELFIES has the effect that small changes in the syntax can lead to structurally highly dissimilar molecules. Thus, while a model based on SELFIES would generate only valid representations it might be much harder to learn structural properties from a reference CS with the goal of expanding it. This observation has been confirmed showing that indeed SELFIES-based GMs will generate 100% valid molecular representations, however the CS covered tends to differ more strongly from a reference training set regarding metrics characterizing structural and physicochemical properties [22].

3. Architectures of generative models

3.1. Conventional approaches based on probabilistic methods

Approaches from machine learning and utilization of optimization algorithms have been incorporated in GMs for CS exploration that predate the rise of deep learning in chemoinformatics [12]. Most common among these are fragment-based methods relying on fragment recombinations [33–38] that can be either explored combinatorially or using single or multi-parameter optimization strategies [39,40]. These methods typically employ probabilistic methods with the aim to generate chemically reasonable molecules and optimize these with respect to one or more endpoints.

Popular probabilistic methods used in fragment-based GMs include genetic or evolutionary optimization algorithms (GAs) [38–43], ant colony optimization [44], Markov chains [35], or Monte-Carlo tree search [43]. By design, all these methods are well suited for focused or goal directed CS exploration campaigns because their iterative probabilistic nature can guide exploration towards CS with favorable properties as part of a single or multi-objective optimization strategy. Fragment-based *de novo* design methods are not limited to exploration of local CS and can also be adopted for generic sampling of CS [38] putting the emphasis on exploring reasonable synthetically accessible CS where the distribution of generated molecules mimics that of a reference set to which the models are adapted.

Probabilistic methods are not limited to fragment-based approaches. E.g., ChemGE [42] uses a pool of chromosomes representing rules

of a context-free grammar from which SMILES [6] representations of molecules are generated, and STONED [45] uses random mutations of non-redundant SELFIES [32] for goal-directed CS exploration.

3.2. Deep neural networks

By design, fragment-based methods (and the grammar-based ChemGE method) incorporate basic chemical knowledge so that the generated output consists of correct chemical structures following basic valence rules and is directed toward chemically reasonable CS. In DNNs however, these aspects need to be learned during training. First, a model needs to learn to generate syntactically valid molecular representations, and, second, the molecules generated should properly reflect the reference CS on which the model was trained. These goals are not strictly separable during training. Initial training of DNNs requires relatively large data sets. Retraining techniques like transfer learning or reinforcement learning can be used to direct the CS of a DNN towards a specific region of interest, e.g., molecules showing favorable properties or specific bioactivities, for which only small amounts of data are available.

GMs for CS exploration have mainly focused on three types of architectures: recurrent neural networks (RNNs), variational autoencoders (VAEs), and generative adversarial neural networks (GANs).

3.2.1. Recurrent neural networks

RNNs have their origin in signal and natural language processing [27] and can thus be very well adapted for exploring CS using linear representations like SMILES [46–51]. RNNs process an input sequence symbol by symbol, where its behavior at any given point t depends on the current input symbol s_t and an internal state at $t - 1$, which in turn depends on the initial part of the sequence up to $t - 1$ (see Fig. 2a). Memory units like long short-term memory (LSTM) [52] or gated recurrent units (GRU) [53] can selectively keep track of and memorize parts of the input. RNNs resemble Markov chains in that they learn conditional probabilities for each symbol depending on the history of previous symbols. However, whereas the distributions of Markov chains are only dependent on the immediately preceding symbol, RNNs can encode long-term dependencies and are thus suited to learn complex grammars, which make them predestined for natural language processing and the generation of SMILES by encoding grammar rules like correctly parenthesized

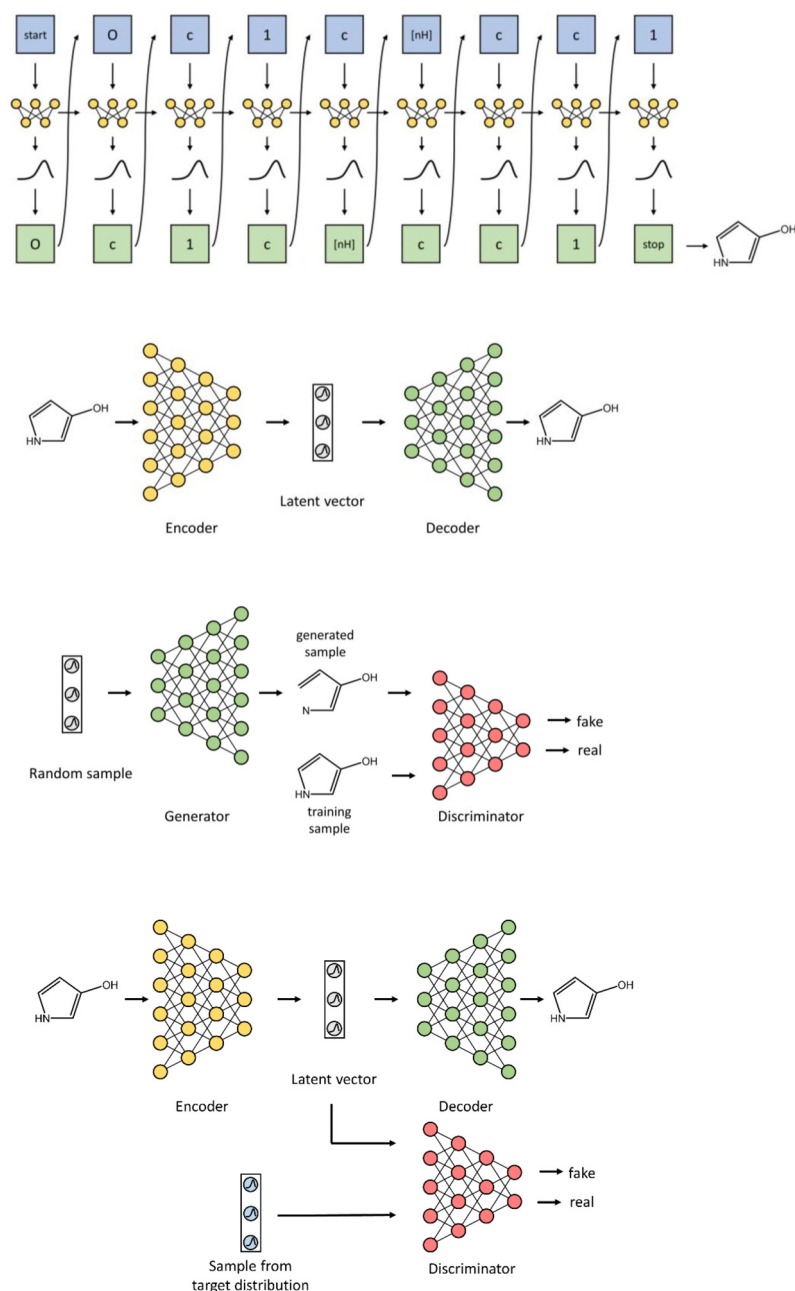


Fig. 2. DNN architectures for generative models. Four different architectures for deep generative models are displayed schematically. (a) Recurrent neural network, (b) variational autoencoder, (c) generative adversarial network, (d) adversarial autoencoder.

expressions or matching ring closure symbols in their probability distributions.

Trained RNNs can generate molecular representations by sampling sequences of symbols $s_1, \dots, s_{t-1}, s_t, \dots$ where the probability of each symbol is determined by the learned distributions. Several molecular generative networks have been suggested based on this principle [46–51]. Frequently, however, RNNs are used as modules in other DNN architectures. E.g., the learned hidden states of RNNs can be used as the basis for encoding molecules in a so-called latent space as part of variational autoencoders (VAEs) [15,29]. They have also been used as generators in GANs [54] or autoencoder-based GANs [55–57] instead of more conventional convolutional neural networks.

3.2.2. Variational autoencoders

The idea of autoencoders is to take a molecular representation and map it onto a latent continuous vector space in such a way that the original molecule can be reconstructed. To this end, an autoencoder requires

two DNNs (See Fig. 2b), one to encode the representation as a latent vector and one to decode the latent vector, recreating the original representation (or an alternative representation for the same molecule [29]). The latent space is a relatively low-dimensional mostly non-redundant vector space. Because the projection mapping of molecules into this space has to be learned and is not predefined by the architecture it is possible to train autoencoders in such a way that neighborhoods in latent space reflect similarities of properties or biological activities of interest. Thus, sampling from neighborhoods of molecules in latent space can be used to sample novel molecules with similar properties thus expanding the local CS [15,16,29,58,59].

While basic autoencoders might be able to recreate inputs without error they are not sufficient to guarantee that latent space representations have meaningful similarity relationships and thus are not well suited for generative purposes. VAEs can be seen as Bayesian equivalents of autoencoders and instead of learning a fixed representation, VAEs learn a multivariate Gaussian distribution in latent space. This helps to en-

sure continuity, so that neighborhoods in latent space reflect molecular similarity [15].

3.2.3. Generative adversarial networks

A GAN consists of two components, a generator and a discriminator [27]. Based on a trainable probability distribution, the generator will generate molecular representations while the discriminator will classify a representation as either “real” or “fake” aiming to distinguish “real” molecular representations from a reference training set from “fake” ones produced by the generator (see Fig. 2c). The two networks have opposite goals, hence the name “adversarial”; while the discriminator’s task is to accurately distinguish real from generated samples, the generator’s goal is to make this task as hard as possible for the discriminator. The generator and discriminator network are trained simultaneously, and learning is achieved by the competitive nature of the process. While different GAN architectures have been proposed for CS exploration [54,60], for instance using RNNs as generators [54], they are also often combined with autoencoders in what are known as adversarial autoencoders (AAEs). Here, a generator samples from the latent space of the autoencoder while a discriminator distinguishes real from fake latent vectors. Independent of the discriminator, the decoder is used to generate the molecular representation from the latent vector [55–57,61] (see Fig. 2d).

3.3. Training and goal-directed chemical space exploration for deep generative models

Although it has been recently shown that it is possible to successfully train DNNs on relatively small training sets of around 10,000 molecules [22], i.e., such models are able to generate valid molecular representations at an acceptable rate, these models might not be good enough to accurately reflect the CS of reference sets. However, the focused exploration of CS and the optimization of sampled molecules regarding specific (biological) properties like activity is a central task in chemoinformatics. Typically, the data available for these tasks is often at best one order of magnitude less than required to train a deep GM. For example, even for the most prominent targets, only a few thousand active molecules are reported in databases like ChEMBL [62]. Two popular approaches to address this issue are transfer learning and reinforcement learning [27]. They rely on first training a model on a large generic data set covering a more general CS, e.g., bioactive molecules, and then tuning the model towards a region of interest using a much smaller training set relevant for the task at hand. In the case of transfer learning this is done by retraining a pre-trained model using a focused reference data set of interest [46,47,50,51], while in reinforcement learning, the probabilistic distribution underlying the generative process is tuned towards a target CS [49,55,58]. This approach is well suited for GANs [54,60,63] and AAEs [55–57,61] but has also been applied to RNNs [64,65].

Considering that neighborhood relationships in latent space representations can be modified during training, they can also be tuned to reflect similarities of chemical properties of interest. Gómez-Bombarelli et al. [15] and Colby et al. [16] modified the loss function of the VAE to include regression errors with respect to molecular properties in order to control latent space representations and allow optimization of such properties in latent space. In AAEs, the distribution of molecules in latent representation is controlled directly by the discriminator, which has been shown to improve the continuity of latent space representations [66]. Polykovskiy et al. [56] and Hong et al. [61] combined molecular representations with molecular properties resulting in latent space representations that allow conditional generation of molecules with desired properties. Here, latent space of molecular representation is trained to be independent of specific molecular properties while the decoder is augmented with a property vector to generate molecules with desired properties.

4. Validation of generative models

4.1. Validation metrics

The nature of DNNs precludes easy interpretation, which makes it hard to rationalize how CS is covered by deep GMs. Generative DNN models like VAEs, AAEs, or GANs sample from a continuous latent space. They are trained with the goal that neighborhoods in latent space represent meaningful neighborhoods in generated CS characterized by continuity with respect to structural changes and/or with respect to physicochemical or biological properties. Contrast this with fragment-based methods using randomization to compose, combine, and recombine fragments. By design, such methods can, for instance, mimic the frequency of fragments or functional groups from a reference set and ensure that generated molecules share similar structural features. The obvious limitation here is that fragment-based models might be too conservative in their approach and explore CS based on structural similarity alone. While deep GMs are not restricted in this way, they have to acquire all their chemical knowledge by training, which is an additional challenge in generating CS that reflects structural and physicochemical properties of a reference set. This aspect, combined with their opaque nature makes validation and assessment of generated CS a central aspect of building successful models. Several recent publications have focused on assessing and benchmarking the CSs generated by fragment-based and deep GMs [22,67,68].

Basic aspects of the quality of GMs include validity, novelty, and uniqueness of generated molecules. These quantities can be easily calculated from a reference set and a sample of generated molecules. The validity criterion assesses to what extent a GM generates valid representations of chemically sound structures. In some sense, validity by itself can be considered the least crucial of these aspects. The validity of a molecule can be easily checked with respect to syntactic correctness of the chosen representation and the adherence to basic chemical rules regarding atomic valencies and bond orders. Thus, as long as a GM produces a significant fraction of valid molecules, invalid representations can be easily filtered out, which could still result in a model that reasonably samples CS albeit in a less efficient manner. Nevertheless, validity is an important indicator of successful training, as it shows that a model can learn the syntax and grammar of a representation, which can be considered a prerequisite for learning the chemistry and properties of the CS to be represented by the model. Validity has been demonstrated to correlate well with improved performance while training on increasing data set sizes [22].

Novelty characterizes the property of a GM to generate molecules unseen in the training set while uniqueness reflects its ability to avoid recreating the same molecule multiple times. These quantities characterize the size of the CS covered by a GM and can be close to 100% if the covered CS is multiple orders of magnitude larger than either the size of the training set or the number of molecules sampled from the GM. To put this in perspective, randomly sampling from a (theoretical) space of N molecules, one can expect to start seeing duplicate samples for samples sizes of ca. \sqrt{N} or more molecules, while in a sample of N molecules one can expect to see ca. 37% duplicates. The training set size usually represents a tiny fraction of the targeted CS so that ideally the fraction of novel molecules exceeds 99%. Low novelty can be a sign of overfitting, and low uniqueness can indicate a mode collapse. Mode collapse describes a lack of diversity of generated molecules and can be ascribed to a concentration of the probability distribution around a few molecules. While a GM should aim to maximize these metrics, its potential to do so might be limited if the size of the targeted CS is small; for instance when exploring CS around specific scaffolds [69].

Zhang and co-workers [70] trained multiple GMs on a small subset of GDB-13 [4] comprising roughly 0.1% of a completely enumerated CS of ca. one billion molecules with sizes of up to 13 heavy atoms. The

GMs were used to sample one billion compounds each and the best resulting models achieved a coverage of 39%. Following the argument above, a coverage of about 63% would be the theoretically best performance that could be expected assuming perfect coverage and identical probabilities for each molecule. Given that the GMs also generated a significant portion of molecules not covered by GDB-13 this indicated that the GMs are indeed able to cover a significant part of the whole GDB-13.

While validity, novelty, and uniqueness can confirm that a GM will indeed explore novel CS, these metrics are not sufficient to characterize it regarding relevant biological and physicochemical properties. Beyond purely random sampling of CS, the envisioned utility of GMs lies in their ability to generate novel compounds having properties that are characterized by a reference set used for training either directly or as part of re-training to focus the covered CS. The ability of GMs to adequately cover the targeted CS can be assessed by comparing the property distributions of the training set molecules and the sampled molecules. For practical applications this approach is limited to properties that can be easily computed. These include simple descriptors like molecular weight or atom composition, topological descriptors like the Bertz topological complexity [71], or physicochemical properties like the octanol-water partition coefficient $\log P(O/W)$ for which computational models exist [22,67,68] but also extends to more complex descriptors like the synthetic accessibility score [10] and the quantitative estimate of drug-likeness [11]. The similarity between reference and sample distributions is quantified using metrics like the Kullback–Leibler divergence, the Jensen–Shannon distance or the Wasserstein distance [22]. Structural similarity can be assessed based on scaffold and fragment distributions [68] and on Tanimoto similarities of structural fingerprints [22,67,68] and can also be used to detect issues like mode collapse or strong biases in the trained models.

While these distribution-based descriptors can be used to determine how representative a CS is with respect to structure and physicochemical properties, they do not address biological properties. The Fréchet ChemNet Distance (FCD) [72] has been developed with this in mind. The concept was inspired by the Fréchet inception distance [73]. However, instead of relying on the Inception v3 [74] network for image classification, it relies on a DNN, termed ChemNet, that has been trained to predict bioactivities of molecules for more than 6000 assays. The activations of the neurons of the final hidden layer of the network encode the relevant features for the predictions of the output layer. For a set of molecules the distribution of these activations are modeled as a multivariate Gaussian distribution. The distributions of a reference set of real molecules and a sample of generated molecules are compared using the Fréchet distance. While the quality of this distance will be limited by the accuracy of ChemNet and the quality and generality of the assay data, the concept is very elegant for the assessment of the biological relevance of a CS.

Under the assumption that larger training data sets should be able to improve the performance of GMs, Skinnider et al. [22] studied the correlation between the metrics discussed here and training set size. They found a very high correlation between validity and training set size (using SMILES). FCD also showed high correlation, while most structural and physicochemical descriptors showed good to satisfactory correlations. While smaller training sets were sufficient to generate valid representations at a reasonable rate, other metrics improved significantly with larger training sets. This indicated that it was easier for GMs to learn the syntax of the representations, but more information was required to accurately reflect the CS of the reference sets. Somewhat counter-intuitively, training set sizes were only slightly correlated with uniqueness and negatively correlated with novelty. A possible explanation might be that a GM that only learned the grammar and is not (yet) restricted by the properties of the reference space might be less likely to reproduce training molecules while generating unique molecules at a similar level.

4.2. Model interpretation

Beyond the characterization of generated CS using the discussed metrics the ability to rationalize the CS based on an understanding of the GM is an important aspect of assessing its quality. Given that the black-box character of DNNs makes it hard to explain outcomes, model interpretation has not been a central aspect of deep GM research in chemoinformatics and few publications have considered it.

Analyzing latent space representations can yield insights into how similarity is perceived by a deep GM. For instance, in Ref. [55] the relationship between neighborhoods in latent space and structural similarity of exemplary compounds was investigated. A more recent approach used generative topographic mapping to map the latent space of an autoencoder onto two dimensions [75], which the authors then used to construct an activity map of adenosine A2a receptors visualizing how these active compounds were distributed in latent space.

In Ref. [76], input saliency maps of SMILES were used to aid the interpretability of the generative process. Input saliency maps assign a score to each token indicating its relevance for the next token to be generated. Mapping the tokens to molecular structures then makes it possible to interpret them in a chemical context.

5. Conclusions

Although conventional fragment-based approaches have been available for CS exploration for a long time the deep learning revolution of the last years has invigorated research in GMs using DNNs. Linear molecular representations have facilitated the adoption of many DNN architectures developed for signal and natural language processing. Training and optimizing the architectures of DNNs is still a very time-consuming and demanding task. However, the increasing affordability of GPU-clusters and/or the availability of pre-trained models make their usage more attractive and widely available. The SMILES notation remains the most popular linear representation for DNNs. While alternative notations like DeepSMILES or SELFIES have been developed with the purpose of simplifying the training of DNNs they have not been adopted widely and have yet to show a significant advantage. The attractiveness of DNNs for GMs lie for one in their abstraction from a strictly structural context and their flexibility, for instance by augmenting representations with additional properties [69,77] or by tuning their behaviour using transfer or reinforcement learning. The idea that a DNN can project molecules onto a latent space, explicitly or implicitly, where neighborhood relationships are characterized by similarity of biological properties and activities is a major advantage over fragment-based methods and has the potential to lead to the discovery of more diverse structures that nevertheless maintain required properties and bioactivities. It is very hard to assess to what extent current GMs are already able to perceive CS this way. A few prospective applications of deep GMs have been reported [58,78–80]. However, whether these models still mostly rely on structural similarity or more advanced parameterization of CS is hard to rationalize. Current benchmark systems have only very limited capabilities in this regard. Their metrics are restricted to descriptors immediately derivable from the structural representation. FCD represents a promising strategy to address this issue by utilizing a latent representation aimed at classifying biological activities. - Ultimately, a better understanding of deep GMs is highly desirable. Visualizations of latent space and investigating local neighborhoods are first steps in the understanding of their structure and input saliency maps can pinpoint key dependencies in the generation of linear representations that can be interpreted on a structural level.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] Kirkpatrick P, Ellis C. Chemical space. *Nature* 2004;432:823–823.
- [2] Bohacek RS, McMartin C, Guida WC. The art and practice of structure-based drug design: a molecular modeling perspective. *Med Res Rev* 1996;16:3–50.
- [3] Fink T, Bruggesser H, Reymond JL. Virtual exploration of the small-molecule chemical universe below 160 Daltons. *Angew Chem Int Ed* 2005;44:1504–8.
- [4] Blum LC, Reymond JL. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J Am Chem Soc* 2009;131:8732–3.
- [5] Ruddigkeit L, van Deursen R, Blum LC, et al. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J Chem Inf Model* 2012;52:2864–75.
- [6] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Model* 1988;28:31–6.
- [7] Borel E. La mécanique statique et l'irréversibilité. *J de Physique Théorique Et Appliquée* 1913;3:189–96.
- [8] Wermuth CG, Aldous D, Raboisson P, et al. The practice of medicinal chemistry. Academic Press; 2015.
- [9] Vogt M. How do we optimize chemical space navigation? *Expert Opin Drug Discov* 2020;15:523–5.
- [10] Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J Cheminf* 2009;1:8.
- [11] Bickerton GR, Paolini GV, Besnard J, et al. Quantifying the chemical beauty of drugs. *Nat Chem* 2012;4:90–8.
- [12] Schneider D, Fechner U. Computer-based de novo design of drug-like molecules. *Nat Rev Drug Discov* 2005;4:649–63.
- [13] Hartenfeller M, Zettl H, Walter M, et al. DOGS: Reaction-driven de novo design of bioactive compounds. *PLoS Comput Biol* 2012;8:e1002380.
- [14] Yonchev D, Bajorath J. Integrating computational lead optimization diagnostics with analog design and candidate selection. *Future Sci OA* 2020;6:FSO451.
- [15] Gómez-Bombarelli R, Wei JN, Duvenaud D, et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Sci* 2018;4:268–76.
- [16] Colby SM, Nuñez JR, Hodas NO, et al. Deep learning to generate in silico chemical property libraries and candidate molecules for small molecule identification in complex samples. *Anal Chem* 2019;92:1720–9.
- [17] Baskin I, Gordeeva E, Devdariani R, et al. Methodology of the inverse problem solution for the structure property relation in case of topological indices. *Dokl Akad Nauk SSSR* 1989;307:613–17.
- [18] Brüggemann R, Pudenz S, Carlsen L, et al. The use of Hasse diagrams as a potential approach for inverse QSAR. *SAR QSAR Environ Res* 2001;11:473–87.
- [19] Miyao T, Funatsu K. Finding chemical structures corresponding to a set of coordinates in chemical descriptor space. *Mol Inform* 2017;36:1700030.
- [20] Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. *Science* 2018;361:360–5.
- [21] Vogt M. Using deep neural networks to explore chemical space. *Expert Opin Drug Discov* 2021;17:297–304.
- [22] Skinnider MA, Stacey RG, Wishart DS, et al. Chemical language models enable navigation in sparsely populated chemical space. *Nat Mach Intel* 2021;3:759–70.
- [23] Jin W, Barzilay R, Jaakkola T, Dy J, Krause A. Junction tree variational autoencoder for molecular graph generation. In: Proceedings of the 35th international conference on machine learning, 80. PMLR; 2018. p. 2323–32. Proceedings of Machine Learning Research 10–15 Jul.
- [24] You J, Liu B, Ying R, et al. Graph convolutional policy network for goal-directed molecular graph generation. In: NIPS'18: Proceedings of the 32nd international conference on neural information processing systems; 2018. p. 6412–22.
- [25] Li Y, Zhang L, Liu Z. Multi-objective de novo drug design with conditional graph generative model. *J Cheminf* 2018;10:33.
- [26] Mercado R, Rastemo T, Lindelöf E, et al. Graph networks for molecular design. *Mach Learn Sci Technol* 2021;2:025023.
- [27] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA: The MIT Press; 2017.
- [28] Heller SR, McNaught A, Pletnev I, et al. InChI, the IUPAC international chemical identifier. *J Cheminf* 2015;7:23.
- [29] Bjerrum E, Sattarov B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules* 2018;8:131.
- [30] Arús-Pous J, Johansson S, Prykhodko O, et al. Randomized SMILES strings improve the quality of molecular generative models. *ChemRxiv* 2019. doi:10.26434/chemrxiv.8639942.v2.
- [31] O'Boyle N, Dalke A. DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures. *ChemRxiv* 2018. doi:10.26434/chemrxiv.7097960.v1.
- [32] Krenn M, Häse F, Nigam A, et al. Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. *Mach Learn Sci Technol* 2020;1:045024.
- [33] Wang R, Gao Y, Lai L. LigBuilder: a multi-purpose program for structure-based drug design. *J Mol Model* 2000;6:498–516.
- [34] Chéron N, Jasty N, Shakhnovich EI. OpenGrowth: an automated and rational algorithm for finding new protein ligands. *J Med Chem* 2015;59:4171–88.
- [35] Kutchukian PS, Lou D, Shakhnovich EI. FOG: Fragment optimized growth algorithm for the de novo generation of molecules occupying druglike chemical space. *J Chem Inf Model* 2009;49:1630–42.
- [36] White D, Wilson RC. Generative models for chemical structures. *J Chem Inf Model* 2010;50:1257–74.
- [37] Rodrigues T, Hauser N, Reker D, et al. Multidimensional de novo design reveals 5-HT2b receptor-selective ligands. *Angew Chem Int Ed* 2014;54:1551–5.
- [38] Polishchuk P. CREM: chemically reasonable mutations framework for structure generation. *J Cheminf* 2020;12:28.
- [39] Brown N, McKay B, Gasteiger J. A novel workflow for the inverse QSPR problem using multiobjective optimization. *J Comput Aided Mol Des* 2006;20:333–41.
- [40] Nicolaou CA, Apostolakis J, Pattichis CS. De novo drug design using multiobjective evolutionary graphs. *J Chem Inf Model* 2009;49:295–307.
- [41] Brown N, McKay B, Gilardoni F, et al. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J Chem Inf Comput Sci* 2004;44:1079–87.
- [42] Yoshikawa N, Terayama K, Sumita M, et al. Population-based de novo molecule generation, using grammatical evolution. *Chem Lett* 2018;47:1431–4.
- [43] Jensen JH. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem Sci* 2019;10:3567–72.
- [44] Reutlinger M, Rodrigues T, Schneider P, et al. Multi-objective molecular de novo design by adaptive fragment prioritization. *Angew Chem Int Ed* 2014;53:4244–8.
- [45] Nigam A, Pollice R, Krenn M, et al. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *ChemRxiv* 2021. doi:10.26434/chemrxiv.13383266.v2.
- [46] Segler MHS, Kogej T, Tyrchan C, et al. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Sci* 2017;4:120–31.
- [47] Gupta A, Müller AT, Huisman BJH, et al. Generative recurrent networks for de novo drug design. *Mol Inform* 2017;37:1700111.
- [48] Ertl P, Lewis R, Martin E, et al. In silico generation of novel, drug-like chemical matter using the LSTM neural network. *arXiv* 2017. doi:10.48550/arXiv.1712.07449.
- [49] Olivecrona M, Blaschke T, Engkvist O, et al. Molecular de-novo design through deep reinforcement learning. *J Cheminf* 2017;9:48.
- [50] Amabilino S, Pogány P, Pickett SD, et al. Guidelines for recurrent neural network transfer learning-based molecular generation of focused libraries. *J Chem Inf Model* 2020;60:5699–713.
- [51] Yonchev D, DeepCOMO B.J. From structure-activity relationship diagnostics to generative molecular design using the compound optimization monitor methodology. *J Comput Aided Mol Des* 2020;34:1207–18.
- [52] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9:1735–80.
- [53] Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics; 2014. p. 1724–34.
- [54] Sanchez-Lengeling B, Outeiral C, Guimaraes GL, et al. Optimizing distributions over molecular space, an objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv* 2017. doi:10.26434/chemrxiv.5309668.v3.
- [55] Blaschke T, Olivecrona M, Engkvist O, et al. Application of generative autoencoder in de novo molecular design. *Mol Inform* 2017;37:1700123.
- [56] Polykovskiy D, Zhebrak A, Vetrov D, et al. Entangled conditional adversarial autoencoder for de novo drug discovery. *Mol Pharm* 2018;15:4398–405.
- [57] Prykhodko O, Johansson SV, Kotsias PC, et al. A de novo molecular generation method using latent vector based generative adversarial network. *J Cheminf* 2019;11:74.
- [58] Zhavoronkov A, Ivanenkov YA, Aliper A, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat Biotechnol* 2019;37:1038–40.
- [59] Iovanac NC, Savoie BM. Simpler is better: how linear prediction tasks improve transfer learning in chemical autoencoders. *J Phys Chem A* 2020;124:3679–85.
- [60] Putin E, Asadulaev A, Ivanenkov Y, et al. Reinforced adversarial neural computer for de novo molecular design. *J Chem Inf Model* 2018;58:1194–204.
- [61] Hong SH, Ryu S, Lim J, et al. Molecular generative model based on an adversarially regularized autoencoder. *J Chem Inf Model* 2019;60:29–36.
- [62] Gaulton A, Hersey A, Nowotka M, et al. The ChEMBL database in 2017. *Nucleic Acids Res* 2016;45:D945–54.
- [63] Putin E, Asadulaev A, Vanhaelen Q, et al. Adversarial threshold neural computer for molecular de novo design. *Mol Pharm* 2018;15:4386–97.
- [64] Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. *Sci Adv* 2018;4:eaap7885.
- [65] Blaschke T, Arús-Pous J, Chen H, et al. REINVENT 2.0: an AI tool for de novo drug design. *J Chem Inf Model* 2020;60:5918–22.
- [66] Makhzani A, Shlens J, Jaitly N, et al. Adversarial autoencoders. *arXiv* 2015. doi:10.48550/arXiv.1511.05644.
- [67] Brown N, Fiscato M, Segler MH, et al. GuacaMol: Benchmarking models for de novo molecular design. *J Chem Inf Model* 2019;59:1096–108.
- [68] Polykovskiy D, Zhebrak A, Sanchez-Lengeling B, et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. *Front Pharmacol* 2020;11. doi:10.3389/fphar.2020.565644.
- [69] Chen H, Vogt M, Bajorath J. DeepAC – conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds. *Digital Discov* 2022;1:898–909.
- [70] Zhang J, Mercado R, Engkvist O, et al. Comparative study of deep generative models on chemical space coverage. *J Chem Inf Model* 2021;61:2572–81.
- [71] Bertz SH. The first general index of molecular complexity. *J Am Chem Soc* 1981;103:3599–601.

- [72] Preuer K, Renz P, Unterthiner T, et al. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *J Chem Inf Model* 2018;58:1736–41.
- [73] Heusel M, Ramsauer H, Unterthiner T, et al. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Adv Neural Inf Process Syst* 2017;30:6627–38.
- [74] Salimans T, Goodfellow I, Zaremba W, Lee D, Sugiyama M, Luxburg U, et al. Improved techniques for training GANs. *Adv Neural Inf Process Syst* 2016;29:2234–42.
- [75] Sattarov B, Baskin II, Horvath D, et al. De novo molecular design by combining deep autoencoder recurrent neural networks with generative topographic mapping. *J Chem Inf Model* 2019;59:1182–96.
- [76] Bagal V, Aggarwal R, Vinod PK, et al. MolGPT: Molecular generation using a transformer-decoder model. *J Chem Inf Model* 2021;62:2064–76.
- [77] He J, You H, Sandström E, et al. Molecular optimization by capturing chemist's intuition using deep neural networks. *J Cheminf* 2021;13.
- [78] Yuan W, Jiang D, Nambiar DK, et al. Chemical space mimicry for drug discovery. *J Chem Inf Model* 2017;57:875–82.
- [79] Merk D, Friedrich L, Grisoni F, et al. De novo design of bioactive small molecules by artificial intelligence. *Mol Inform* 2018;37:1700153.
- [80] Grisoni F, Neuhaus CS, Gabernet G, et al. Designing anticancer peptides by constructive machine learning. *ChemMedChem* 2018;13:1300–2.