Received: 27 August 2021 | Revised: 17 December 2021 | Accepted: 20 January 2022

DOI: 10.1002/wcms.1603

ADVANCED REVIEW

WIREs COMPUTATIONAL MOLECULAR SCIENCE WILEY

A review of molecular representation in the age of machine learning

Daniel S. Wigh[1] | **Jonathan M. Goodman**[2] | **Alexei A. Lapkin**[1]

[1]Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge, UK

[2]Yusuf Hamied Department of Chemistry, University of Cambridge, Cambridge, UK

Correspondence
Alexei A. Lapkin, Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK.
Email: aal35@cam.ac.uk

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/S024220/1; UCB

Edited by: Raghavan Sunoj, Associate Editor

Abstract

Research in chemistry increasingly requires interdisciplinary work prompted by, among other things, advances in computing, machine learning, and artificial intelligence. Everyone working with molecules, whether chemist or not, needs an understanding of the representation of molecules in a machine-readable format, as this is central to computational chemistry. Four classes of representations are introduced: string, connection table, feature-based, and computer-learned representations. Three of the most significant representations are simplified molecular-input line-entry system (SMILES), International Chemical Identifier (InChI), and the MDL molfile, of which SMILES was the first to successfully be used in conjunction with a variational autoencoder (VAE) to yield a continuous representation of molecules. This is noteworthy because a continuous representation allows for efficient navigation of the immensely large chemical space of possible molecules. Since 2018, when the first model of this type was published, considerable effort has been put into developing novel and improved methodologies. Most, if not all, researchers in the community make their work easily accessible on GitHub, though discussion of computation time and domain of applicability is often overlooked. Herein, we present questions for consideration in future work which we believe will make chemical VAEs even more accessible.

This article is categorized under:
Data Science > Chemoinformatics

1 | INTRODUCTION

Representing chemical data in a concise and unambiguous way, understandable by both humans and machines, is not an easy task; this is particularly true for the representation of molecules. While there are numerous methods of adequately representing small and "simple" organic molecules, significant complexity may arise when considering molecules with features such as ring structures, nonstandard valency/bonding, inorganic components, or symmetry. These complexities may lead to issues such as representations being noncanonical (i.e., multiple different representations for the same molecule), being nonunique/clashing (i.e., multiple different molecules that are encoded into the same

representation), assuming the wrong number of implicit hydrogen atoms, or failing to capture tautomerism. This can make (sub)structure searching in databases difficult, and even result in representations that refer to the wrong molecules. One way of elucidating the robustness of a representation is with a so-called "round-trip conversion experiment," which tracks whether the conversion from representation to structure and back is correct for a given molecule. As an example, one could draw a molecule in ChemDraw, read it into ChemDoodle, and then check whether the same structure is obtained when reading the ChemDoodle file back into ChemDraw. Broadly speaking, molecules can be represented in a machine-readable format in four ways: as a string; with a connection table; as a collection of features, for example, a fingerprint or series of physical descriptors; or most recently, with a computer-learned representation using machine learning (ML).

As the problems that chemists tackle become increasingly complex, interdisciplinary collaboration also becomes more important, particularly with data scientists, with inherent greater ML understanding, and chemical engineers, system-level problem solvers. A key component to most computational chemistry is the choice of machine-readable molecular representation. No representation is perfect for every circumstance, and the choice will depend on a variety of factors, including whether it should be human-reader friendly (e.g., labeling a molecule in a report/spreadsheet), compatibility with other programs or algorithms (e.g., a ML model requiring a numerical input), space constraints (e.g., when populating a database with millions of entries, requiring dozens of lines for a molfile instead of dozens of characters for a simplified molecular-input line-entry system [SMILES] string can quickly add up), and more.

Representing knowledge in a machine-readable format has become a ubiquitous task in the sciences, and there are so many exciting developments within the chemistry community that covering them all would be impossible in one review. This work deals almost exclusively with the representation of small organic molecules, and how said representations can be fed to ML models. An emphasis is placed on the chemical variational autoencoder (VAE) due to this class of model being the first to showcase effective black-box generation of molecular feature vectors. Reaction representation is briefly mentioned herein, though a more complete description and analysis would require a review of its own. Similarly, discussions of the representation of biological molecules, such as proteins (e.g., AlphaFold[1]) and other macromolecules (e.g., HELM[2]), are also considered beyond the scope of this work.

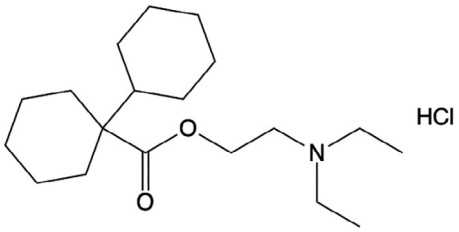
It has been argued that only the applications of molecular representations are of interest, because the basic work was complete by the 1990s.[3] However, there is arguably a renewed interest in the foundations of molecular representation due to the emergence of ML, and its ability to convert a discrete representation of molecules into one that is continuous. Continuous representation enables the use of gradient-descent for optimization with respect to a property, which is much more efficient than a brute-force approach. In addition, the development of new three-dimensional (3D) representations is proving valuable for finding optimal ligands for a given chemical system ("screening ligands"), as binding of proteins also can depend on 3D conformation and alignment.[4] This work provides an introduction to molecular representations which will help the reader appreciate complexities and subtleties which may not be apparent, while concurrently reviewing how aforementioned representations can be coupled with ML to predict molecular properties, generate novel molecular structures, and more.

2 | CLASSES OF MACHINE-READABLE REPRESENTATION OF MOLECULES

Molecules can be represented on a piece of paper using a two-dimensional (2D) scheme such as the one of dicycloverine hydrochloride in Scheme 1, but there are many different options for representing compounds computationally. When computers were first commercialized, strings of alphanumerical characters were preferred, as these required less memory to store and less computational power to process, but as computers developed and memory/processing power became less expensive, less compact representations (that are more flexible and less ambiguous) became more widespread. An overview of the different classes of molecular representation can be seen in Figure 1.

3 | STRING REPRESENTATIONS

String representations generally consist of characters from the American Standard Code for Information Interchange (ASCII) character encoding standard, and are more compact and easier for humans to read and write than other representations. An example showing various string representations for the molecule dicycloverine hydrochloride (Scheme 1) can be seen in Table 1.[5,6]



SCHEME 1 Structure graph of dicycloverine hydrochloride

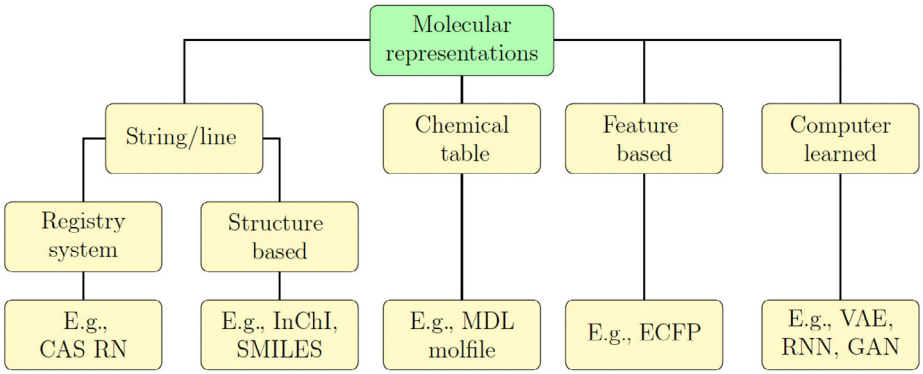


FIGURE 1 Overview of the different classes of molecular representations

TABLE 1 Various representations of the dicycloverine hydrochloride molecule

Generic names[5]	Dicycloverine HCl, benacol, bentyl, dibent, Dyspas, and so on
Mol. formula	$C_{19}H_{36}ClNO_2$
IUPAC name	2-(Diethylamino)ethyl 1-cyclohexylcyclohexane-1-carboxylate hydrochloride
CAS RN	67 − 92 − 5
Canonical SMILES	CCN(CC)CCOC(=O)C1(CCCCC1)C2CCCCC2.Cl
InChI	InChI = 1S/C19H35NO2.ClH/c1-3-20(4-2)15-16-22-18(21)19 (13-9-6-10-14-19)17-11-7-5-8-12-17;/h17H,3-16H2,1-2H3;1H InChIKey:GUBNMFJOJGDCEL-UHFFFAOYSA-N
WLN[6]	L6TJA-AL6TJAVO2N2&2&GH

3.1 | Registry systems

Various registry systems exist, and they share the feature that an arbitrary, but unique, number is assigned to a new molecule that is not already present in their database. Examples include the CAS Registry Number (RN),[7] PubChem CID,[8] ChemSpider,[9] ChEMBL,[10–12] and others. Their globally unique nature eases communication, but decoding these numbers involves referencing the relevant database.

3.2 | Wiswesser Line Notation

First described in 1949, Wiswesser Line Notation (WLN) was one of the first notation formats for representing complex molecules, and it boasted widespread popularity up until the 1970s when it was largely replaced by the more flexible SMILES representation. It sees little use today, which makes encoding/decoding more difficult, and this is unlikely to

change.[13–15] In WLN, digits from "1" to "9" represent unbranched alkyl chains, and uppercase letters represent either an atom or a collection of atoms. It uses uppercase letters for common substructures, which can make WLN quite compact; as an example, two benzene rings connected through an N- and C-atom, which could have the SMILES string "c1ccccc1NCc2ccccc2," can be represented in WLN simply with "RM1R."[6]

3.3 | Fragmentation codes

Chemical patents will often cover a wide range of chemicals, making it infeasible to represent each patented molecule separately. It is therefore useful to represent patented chemicals using Markush structures,[16] where placeholder letters (e.g., R-groups) denote independently variable groups. Although this makes it possible to enumerate all patented molecules from a Markush structure, quickly evaluating whether a seemingly novel compound has already been patented, given a set of Markush structures, can be a challenge. The Chemical Fragmentation Coding System was developed to solve this challenge. The chemical codes are used to index and retrieve chemical patents in Derwent World Patents Index, specifically sections B (pharmaceuticals), C (Agricultural Chemicals), and E (General Chemicals), hence why they are also referred to as BCE chemical codes. The BCE chemical code for a molecule will consist of a set of "words," where each word represents a functional group and is formed of typically four alphanumerical characters. The four-character code is hierarchical, with the first character representing the part, and each additional character defining a smaller, more specific, set of functional groups. As an example, "H" represents "Common Functional Groups Without >C═O or >C═S," while "H724" represents "Two conjugated >C═C< groups present." While representing molecules with fragmentation codes does incur a loss of information, they have proven invaluable for patent searching.[17]

3.4 | IUPAC

The IUPAC nomenclature for organic molecules, developed by the International Union of Pure and Applied Chemistry, uses words to represent functional groups, unlike most other string representations which use letters/numbers. As an example, the molecule CH_4 has the IUPAC name "methane," but would simply be referred to as C in SMILES. Although the use of words makes IUPAC nomenclature less compact, it also makes it easier for humans to read and pronounce. In particular, functional groups may be apparent by inspection, even by non-experts. As an example, the IUPAC name of dicycloverine hydrochloride, shown in Table 1 is the only string representation that instantly reveals the presence of two cyclic groups even to someone who does not know the grammar of the representations.

Canonicalization is important for any representation for the sake of consistency and disambiguity. Although the IUPAC nomenclature was likely intended to be canonical, it is not considered a canonical representation due to the consistent use of alternative forms, particularly retained names. The Preferred IUPAC Name (PIN)[18] was introduced to encourage the use of canonical names. Another concern with IUPAC names is that they can be difficult for computers to understand, though interpreters do exist.[19]

3.5 | Simplified molecular-input line-entry system

SMILES[20,21] represents (organic) molecules with a string of ASCII characters. Atoms are simply represented with the same one- or two-letter symbol that is used in the periodic table, which is one of the reasons why SMILES is more flexible than WLN. Single bonds can either be implicit or represented with −, and double, triple, and quadruple bonds with =, #, and $, respectively. Rings are represented with a number after the (arbitrarily chosen) initial atom in the ring and closing atom (e.g., C1CCNCC1 and N1CCCCC1 are equivalent). Branching is represented with parentheses around the branch, for example, 4-ethylheptane (a 7-C backbone with a two-carbon sidechain on the fourth carbon) would be CCCC(CC)CCC. Branches can also be nested within other branches by adding more parenthesis pairs. Aromaticity can be represented with either alternating single/double bonds, or by writing aromatically bonded atoms in lower case. To illustrate this, consider all of these equivalent representations of benzene: c1ccccc1, C1═C─C═C─C═C1, C1═CC═CC═C1.

Although it can be easy to write down a SMILES string that is syntactically correct since there are many ways to write the same thing, this noncanonical nature of SMILES can make (sub)structure searches in a database difficult.

Despite potentially being computationally expensive, various algorithms have been developed to canonicalize SMILES strings including Universal SMILES,[22] RDKit SMILES[23] and CANGEN.[24]

A chemical reaction is a rearrangement of atoms in or between molecules, and if the reaction context of two reactions is similar, one might reasonably expect the outcome to also be similar. This fact, coupled with the advent of computers, led to the production of reaction heuristics (now called rule-based expert systems or expert-defined reaction templates) which enabled computational reaction prediction in the late 1960s.[25] Automatic rules/template extraction using ML has since been developed[26] and refined[27]; the reaction template specifies the reaction centers of all participating molecules up to a certain radius, and in both papers cited, the template is represented using SMIRKS.[28] SMIRKS is a hybrid representation of SMILES and SMARTS,[29] and SMARTS is a SMILES-based representation of reactions where molecules are separated by "." and reactants are separated from products using ">>." For the template-based approach to reaction prediction to work, the correct template must be chosen for each task; using extended-connectivity fingerprints (ECFPs), in combination with ML has been shown to improve accuracy in template selection.[30]

There is growing interest in exploring how concepts from natural language processing can be repurposed to solve problems in chemistry. In Molecular Transformer chemical reaction prediction is seen as a machine translation tasks, predicting product SMILES strings given reactant SMILES strings.[31] STOUT (*S*MILES-*TO*-I*U*PAC-name *t*ranslator) is a machine translation algorithm translating molecule names from one chemical language to another.[32] Mol2vec[33] uses the Word2vec concept for chemistry; an unsupervised ML model is used to generate a vectorized representation of molecules, with similar compounds having similar vector representations.

When SMILES is used as the language of generative models, the output SMILES strings can sometimes be invalid due to the requirement of parentheses and ring-indication numbers to occur in pairs, and in the right order. As an example, CC)C(CC would not be a valid SMILES string, despite carrying a pair of parentheses. A modification of SMILES, called DeepSMILES,[34] was proposed to alleviate these issues. Instead of parentheses around the branch, only right (closing) parentheses are used, with the number of them indicating the length of the branch, for example, 4-ethylheptane is CCCCCC))CCC; however, using a large number of closing parentheses instead of simply a pair of parentheses can make it less human-reader friendly. When representing ring structures in DeepSMILES, a number follows the final atom of the ring with the value of the number indicating the size of the ring; for example, benzene (could be c1ccccc1 in SMILES) becomes cccccc6 in DeepSMILES. This has the added benefit of instantly revealing the ring-size. However, the issue of chemical validity (e.g., exceeding normal valency) was not addressed with DeepSMILES. The SELFIES[35] (SELF-referencIng Embedded Strings) representation was developed to solve the issue of invalidity of strings on a more fundamental level: SELFIES can reportedly represent every molecule, and every SELFIES string corresponds to a valid molecule. Each symbol in a SELFIES string is derived from the corresponding rule vector and state of derivation. The rule vector represents the type of chemical structure ([C], [=O], etc.), while the state of derivation represents syntactical and chemical constraints (e.g., maximal valency). The robustness of SELFIES has been exploited in a number of different ML applications.[36–39]

3.6 | International Chemical Identifier

International Chemical Identifier (InChI) is an open-source string representation that was developed by IUPAC in 2005. Key benefits include it having built-in canonicalization, being open source, being applicable for most organic and inorganic chemistry, and having a hierarchical structure which allows encoding with different levels of granularity. The representation contains "layers" of information about the compound, each separated by "/" and initiated by a prefix:

1. Main layer (core parent structure)
   - Empirical formula (always present, no prefix)
   - Skeletal connections (prefix: "/c")
   - Hydrogens (prefix: "/h")
2. Charge layer
   - Net charge (prefix: "/q")
   - Protonation/deprotonation (prefix: "/p")
3. Stereochemical layer
   - Double bond (prefix: "/b")
   - Tetrahedral (prefix: "/t")

- Indicator stereo layers (prefix: "/m", "/s")

4. Isotopic layer (prefix: "/i")
5. Fixed H layer (for tautomers, prefix: "/f")
6. Reconnected layer. Typically bonds to metals are broken as part of the normalization procedure; in this layer, the molecule can be represented as if the bonds were intact (prefix: "/r")

It is worth noting that InChI strings always start with "InChI=" followed by a sequence of letters/numbers before the first slash. In the example InChI string in Table 1, "InChI = 1S" indicates that it is a standard InChI of version 1. InChI strings for large and complex molecules can quickly become verbose, so to ensure compatibility with search engines, a 27-character fixed length, hashed version of InChI called "InChIKey" has been developed.[40]

InChI represents structures with great veracity, with InChI v1.03 and StdInChI reportedly achieving 99.95% accuracy on 39 million structures from PubChem Compound in a round-trip conversion experiment which recorded the number of correct InChI → Structure → InChI conversions.[40] As with most things, achieving 100% accuracy is near impossible, though InChI is continuously getting closer. Many improvements have been made since v1.03 such as adding compatibility with the V3000 molfile format, which enables handling of molecules with more than 1000 atoms, in v1.05,[41] and fixing bugs in the normalization procedure. To better understand what might go wrong in the normalization procedure consider the following two examples: an update in InChI v1.04 fixed an issue where some structures containing a radical atom in an aromatic ring might yield different InChI strings for the same molecule depending on the original order of the atomic numbers,[42] and an update in InChI v1.06 fixed a bug which caused a change in the InChI string upon renumbering of atoms for some molecules containing an acidic hydroxy group at a cationic heteroatom center.[43] At the time of writing, the most recent version of InChI is v1.06; a more complete overview of the changes, additions, and bug-fixes associated with this version can be found online for free.[43] With these improvements, and many more, InChI is now more than 99.99% reliable.[44]

3.7 | Other string representations

Less popular line notations than the aforementioned also exist, such as SYBYL Line Notation,[45] and are described elsewhere.[3,46]

4 | CHEMICAL TABLE REPRESENTATIONS

A chemical table (CT) lists the *x*-, *y*-, and *z*-, coordinates of each atom in a connection table (CTab), and how they are bonded to each other in a molecule. This makes the generation of 2D/3D graphic representations from a CT quite easy, and they are typically used for representing molecules within databases/programs. The most widely used is the MDL molfile, which exists in two versions (V2000 and V3000). MDL molfiles can be "bundled" into a structure–data file. One drawback of the CT is that translation or rotation of a molecule will lead to a new set of atom coordinates, despite the molecule being unchanged.

4.1 | MDL molfile

MDL molfiles consist of three main sections: a header block (containing title, timestamp and an optional comment), a CTab, and an end line which must read "M END."[47,48] The CTab consists of a number of sections, best understood by considering Figure 2.

Unfortunately, there is no uniform standard for CTs as both MDL V2000 and MDL V3000 are widely used. There are three main advantages of V3000: the counts line not being is capped at 999 atoms/bonds, an improved description of stereochemistry, and enhanced support for new chemical properties. For many purposes, such as handling of small and nonstereochemical molecules, these improvements were not significant enough to incentivize switching.

One of the primary drawbacks of CTs is related to their handling of complex chemistries. MDL molfiles only support single, double, and triple bonds, which does not work well for molecules containing bonds where simple sharing of electrons in a covalent bond is an inadequate description. It has been suggested that introducing a zero-bond order could (partially) alleviate this issue.[49] Furthermore, when the number of hydrogen atoms is not explicitly stated,

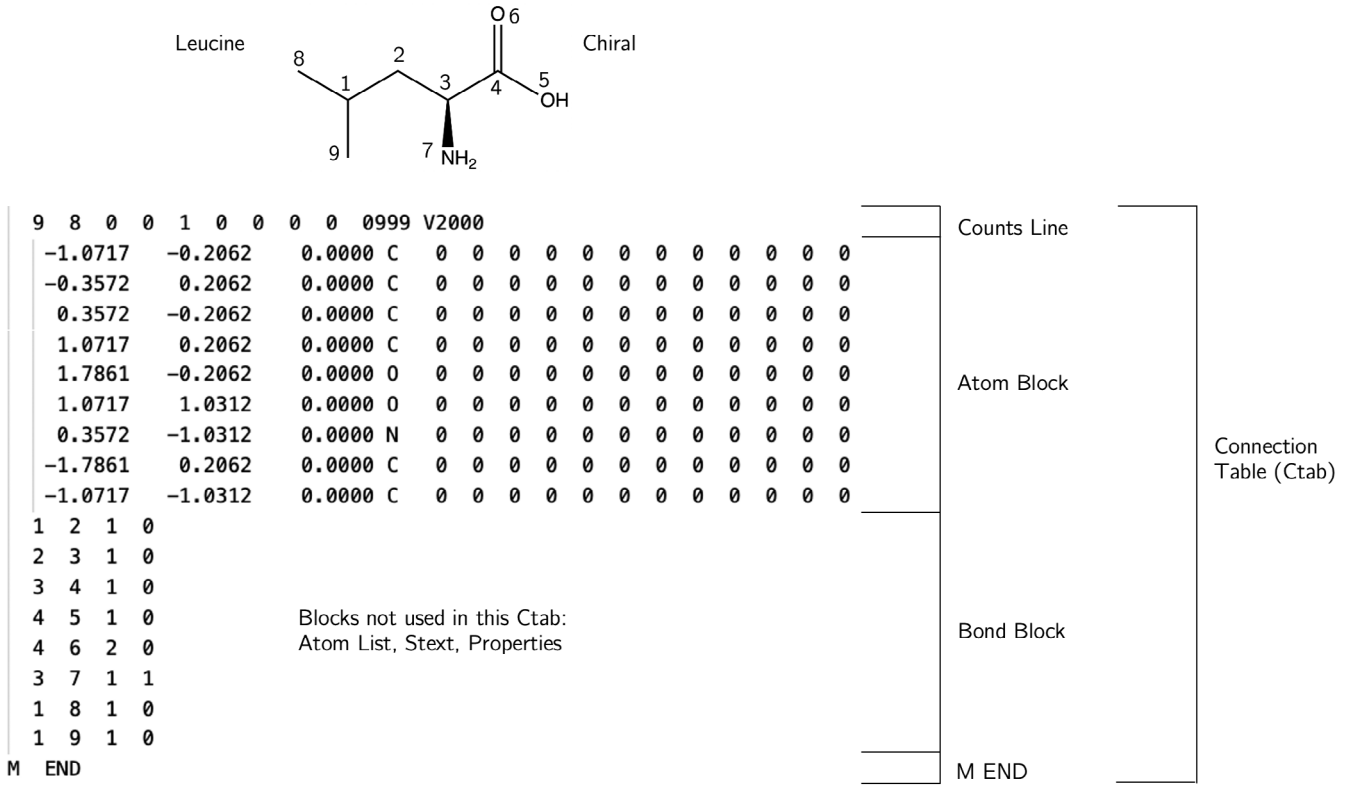


FIGURE 2 Example of a connection table and end line within an MDL molfile V2000 for the molecule leucine generated by ChemDraw[47]

nontrivial valency could lead to the wrong number of implied hydrogen atoms; an important issue not easily solved (simply requiring the number of H-atoms to be specified might compromise back-compatibility).[50]

4.2 | CDXML

CDXML is an XML-compliant version of CDX (ChemDraw Exchange), the native file format of ChemDraw. ChemDraw is a commercial piece of software for handling molecular representations, with features such as structure to name/name to structure, NMR and mass spectrum simulation, and more. Like a CT, a CDX file can record the coordinates for each atom (alternatively, coordinates are omitted and then generated by ChemDraw), though the representation format is different. A CDX file contains a set of nested objects (such as atoms, bonds, fragments) and properties (such as position, color, arrow type, bond order). Each object can have nested objects (zero or more), and also a number of properties associated with it (zero or more),[51] thus creating a tree. Since CDXML is not open source, it does not play much of a role in chemistry research, other than being used by ChemDraw.

5 | FEATURE-BASED REPRESENTATIONS

5.1 | Molecular properties

Perhaps the simplest way to describe a molecule is by listing the features of the molecule which are relevant to the problem at hand in a vector. As an example, it has been shown that a combination of physical molecular descriptors, such as molecular weight, density, melting point etc., reaction-specific descriptors, and descriptors based on screening charge density, can be used to predict which solvent might provide optimal conversion and diastereomeric excess in an Rh-Josiphos catalyzed asymmetric hydrogenation reaction.[52]

Selecting the optimal descriptors for a ML model is difficult even with good domain knowledge, in part due to the opaque nature of ML, and for this reason it is not uncommon for researchers to explore a range of different featurization for the chemical system at hand.[53] The simplest featurization is one-hot encoding, where a vector of 1's and 0's is constructed to represent whether a molecule is present or not present, respectively. As an example, representing the selection of only chemical B from the options A, B, C, and D, might take the form $[0,1,0,0]$, where the first index represents the presence of chemical A, the second index represents the presence of chemical B, and so on. For getting started with a prediction problem it may be helpful to mimic approaches found in the literature,[54] for example, using a binary one-hot encoding (time: quick, detail: none), cheminformatics descriptors generated from open-source libraries[55] (time: medium, detail: medium), and quantum chemical features computed via density-functional theory[56] (time: slow, detail: high).

5.2 | Molecular graphs

Molecules can be conveniently represented as undirected graphs, with nodes as atoms and edges as bonds. Molecular graphs can be a powerful way of representing molecules, and have found their way into many generative model strategies, as described in the section "Beyond string representations in generative models." A molecular graph with featurized nodes (atoms) and edges (bonds) is called an "attributed molecular graph." Features similar to those used in the ECFP (features such as atomic identity, formal charge and aromaticity for each node, and bond order for each edge) can be used to featurize an attributed molecular graph. Using an attributed molecular graph featurized in this way in combination with a convolutional neural network (CNN) can lead to the creation of molecular fingerprints which have enhanced performance in physical property prediction.[57] An alternative to atom-level feature attribution is the reduced graph, where each functional group is replaced by a unit, or superatom, which represents the relevant features. Although the structure to reduced graph transformation is well defined, the reverse transformation is not. The ability to generate novel molecules with favorable properties by first identifying a suitable reduced graph has been explored for de novo molecule design.[58]

5.3 | Extended-connectivity fingerprints

Molecular fingerprints are intended to represent the presence (or absence) of substructures within molecules often in a sparse vector, and they generally fall into one of two categories: matching substructures in a molecule to substructures in an expert-defined set, or algorithmic enumeration and hashing of substructures in a molecule. The ECFP is one of the most widely used chemical fingerprinting techniques due in large part to the popular open-source python package RDKit having an implementation of it, called "Morgan fingerprint." However, chemical fingerprinting techniques existed before the ECFP, see for example HOSE[59] and FREL.[60] Comparing and contrasting fingerprinting methods can be found in the literature,[61] so this text will instead give a brief overview of how ECFPs are generated and may be used in ML models.

The first step in generating an ECFP is using information about each atom in a molecule to yield a descriptor of the atom and its immediate environment that is invariable with how the atoms in the molecule are numbered. Any set of properties which fulfill these criteria could be used. For ECFPs the property set came from the daylight atomic invariants rule: the number of connections, number of nonhydrogen bonds, atomic number, sign of charge, absolute charge, and number of attached hydrogens. The daylight atomic invariants rule was developed by Daylight Chemical Information Systems Inc., the company that invented SMILES, SMARTS, and SMIRKS.[24] The property set used for the ECFP was augmented with an additional feature defining whether the atom is part of a ring. These seven property values are then hashed to yield a 32-bit integer value which can be used to initialize the ECFP algorithm. In the first iteration an array is built from the iteration number, bond orders (single: 1, double: 2, triple: 3, aromatic: 4), and hash values of the neighboring atoms within the appropriate radius (as illustrated in Figure 3); this array is then hashed into a new 32-bit integer, which effectively serves as a label for the substructure. Applying the algorithm iteratively with increasing radius, and saving the intermediate labels, then yields a complete set of labels for all substructures within the molecule up to a given user-specified radius. Barring bit-collision, each unique substructure will map to a unique integer. One interpretation of these integers is as the index of 1's in a vector otherwise consisting of 0's, that is, a sparse vector representation of molecular substructures for the molecule. It is worth noting that such a vector would never be constructed

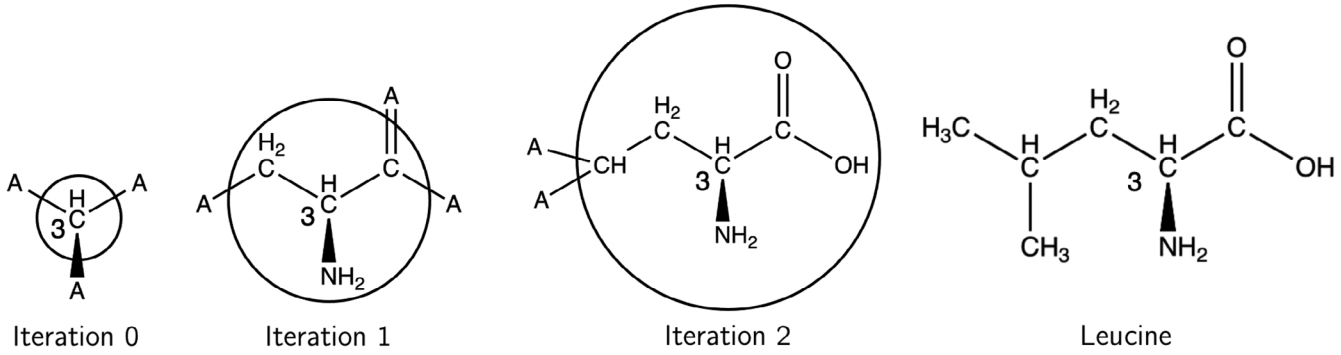


FIGURE 3 Illustration of the extended-connectivity fingerprint (ECFP) algorithm on Leucine using the carbon assigned "3" in Figure 2. With each iteration, the atoms/bonds considered for the next hashed identifier increases as a circle growing around the atom under consideration (hence the name "circular fingerprints"). Iterations are initiated on all non-H atoms in the structure[62]

in practice, since it would be of length $2^{32} \approx 4.3 * 10^9$. Since the size of the labels depends on the hash function, it is possible to create a smaller fingerprint (e.g., 2048 bits) by hashing the labels into smaller space.[62] Although this "folding" operation can worsen quality and increase the risk of bit-collision, there is some evidence to suggest that much of the information is retained.[63] A detailed discussion on initializing, bit-collisions, handling duplicates, and so on can be found elsewhere.[62] RDKit includes an implementation of the ECFP, dubbed "Morgan fingerprint," which can be found online and used for free.[64]

Morgan fingerprints are lightweight, quick to compute, and represent salient features of molecules well, and have thus found many uses beyond computing similarity. Being already formatted as a vector with numerical entries, they are well suited to be used as features in ML models for chemistry.[65]

6 | COMPUTER-LEARNED REPRESENTATIONS

A molecule is a discrete, 3D collection of atoms bonded together through the favorable interaction of their electrons. Many representations of molecules are indeed also discrete, often consisting of a combination of letters and numbers. However, to perform operations on molecules with a computer, a representation entirely consisting of numbers is required. Two approaches were presented above in the section on feature-based representations. Constructing vectors of molecular properties involves a human deciding which properties to include, and as with ECFPs, there is no guarantee that the vectors that these methods produce capture all relevant information; additionally, they are not generally invertible (i.e., it is generally not possible to deduce the molecular structure from the vector). Developing a continuous and invertible representation of molecules within a latent space could be powerful, as this would enable the use of various simple numerical operations on molecules, such as interpolation between molecules and gradient descent optimization with respect to certain properties, which might yield interesting novel molecules which would otherwise be expensive to find with a brute-force systematic combinatorial approach. In January 2018, the first implementation of a computer learned molecular representation that was both continuous and invertible was published, where molecules were encoded by converting their SMILES string to a continuous valued vector using a neural network (NN).[66] Since then, at least 45 papers have been published demonstrating new techniques that can be used to enable computer generated representation of molecules. Three popular deep-learning architectures are recurrent neural networks (RNNs), autoencoders, which includes both VAEs and adversarial autoencoders (AAEs), and finally generative adversarial networks (GANs).[67]

6.1 | Molecule generation based on strings

6.1.1 | What is a VAE?

A VAE consists of two NNs, an encoder and a decoder. The input layer of the encoder consists of a large number of nodes, with each subsequent layer in the encoder containing fewer and fewer nodes, which forces the NN to carry only

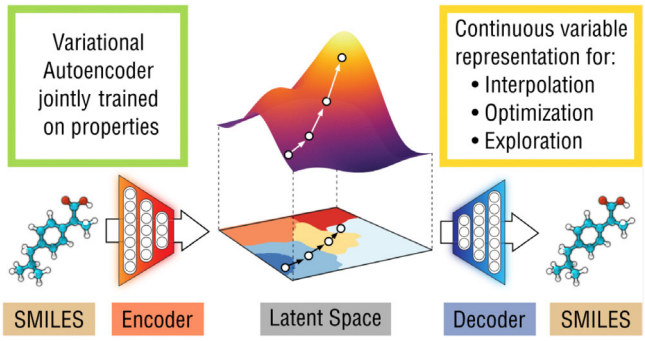


FIGURE 4 This graphic shows how a variational autoencoder (VAE) can be used to interpolate/optimize/explore molecular properties in a continuous latent space, before being decoded to yield a (potentially) valid SMILES string.[66] Figure reused with permission from American Chemical Society (ACS). Further permissions related to the material excerpted should be directed to the ACS

crucial information forward to the next layer. The encoder finally yields a vector in the latent space. This vector is decoded with the decoder to yield an output as similar to the input as possible. The difference between input and output of the VAE is incorporated into the loss function which is used to train the system. Once the decoder is able to consistently produce outputs which are adequately similar (or possibly identical) to the input to the VAE, the VAE is said to be "well trained," and the fact that a well-trained VAE can encode and then decode something despite the constriction in the number of nodes implies that the latent vector in some way represents key features of the input.[68]

Gómez-Bombarelli et al., under the supervision of Professor Aspuru-Guzik, were the first to train a VAE using SMILES strings, allowing them to generate a continuous and invertible representation of molecules.[66] The model architecture used was doubly probabilistic. Gaussian noise was added to the encoder as this would allow the decoder to encounter a broader variety of points in the latent space, resulting in a more robust representation. Noise was also introduced by using nondeterministic sampling of the decoder's final layer. Once well trained, the VAE is capable of encoding a SMILES string as a vector which captures characteristic features about the structure, while the decoder is capable of converting the vector back to a SMILES string, as shown in Figure 3. It is worth noting that the stochastic nature of the VAE, as well as querying the model in sparsely trained regions, may result in the decoder producing SMILES strings different from the one fed to the encoder. A surrogate model $f(z)$ for predicting target properties of the encoded molecule from its encoded vector z was jointly trained with the VAE. This amalgamation contributed to shaping the latent space, placing molecules with similar target properties close to each other in the latent space. Having a smooth latent space organized according to target properties allows for efficient search for points with desirable characteristics using methods such as gradient descent. Decoding optimal points into SMILES strings is a new method for guided novel molecule generation (Figure 4).

6.1.2 | Issues with molecule-generation based on string representations

Although SMILES strings are perhaps the most prevalent molecular representation used with deep learning for novel molecular structure generation,[67] there are a number of issues associated with using these. The two most pervasive issues are that the SMILES strings which are generated may be invalid, and that NNs may be learning the SMILES syntax rather than learning the underlying properties of the molecules that the SMILES strings represent. This is partially due to the noncanonical nature of SMILES; one example of the implications of this is that the SMILES strings for two different molecules may in certain circumstances be more similar than two equivalent but different SMILES strings for the same molecule. When using RNNs, the rate of valid SMILES strings is reportedly greater than 90%.[69,70] However, when using different methods, the rate of generated strings that are valid deteriorates significantly; in one particular implementation of a VAE, the decoding rate was around 73%–79% for points close to known molecules. However, this dropped to roughly 4% for randomly selected points in the latent space.[66]

Given the complexity of real molecular behavior, any choice of representation will almost invariably be a simplification, and when training a model one must keep in mind whether the representation is capable of carrying information which is critical to understanding the molecular behavior (e.g., chirality, tautomerism, etc.), while balancing this against keeping the representation as simple as possible. The obvious approach to strengthening the association

between representation and underlying molecular structure is to feed the algorithm with more data. However, this is not always possible nor productive, thus numerous alternative strategies have been suggested, such as providing models with multiple different, but equivalent, SMILES strings for each molecule.[71] Winter et al. proposed continuous data-driven descriptors[72] which were generated by training a NN to convert between the semantically equivalent but syntactically different string representations SMILES and InChI. A third proposed approach added the semantic and syntactic constraints of SMILES to a VAE decoder using context and attribute free grammar.[73,74] Furthermore, DeepSMILES has been proposed as an alternative to SMILES to solve two of the most common reasons for syntactically invalid SMILES strings, with SELFIES taking a step further by also addressing chemical invalidity (see SMILES).

6.1.3 | Areas of further investigation for chemical VAEs

It is encouraging that most, if not all, researchers make their work easily available postpublication on GitHub, particularly their pretrained models as this allows other researchers to explore potential applications. Cloning a repository may take as little as 5 minutes, and once the model has been loaded to your machine, calculating latent vectors from a molecular representation (e.g., SMILES string) may take less than a second per molecule. We believe there are a few things that the community could do to make using the work of others even more accessible.

The domain of applicability is the area of chemical space where a model can be expected to work "well," with predictions made on molecules outside the domain of applicability being either less accurate or beyond the scope of the model. The importance of defining the domain of applicability should be self-evident: without it one would not know the uncertainty associated with a new data point. Using continuous variables, one might reasonably define the domain of applicability of a model as the hypercube formed from the most extreme point in each dimension, though since molecules are discrete structures, defining a domain of applicability for chemical VAEs is not trivial; indeed, predicting out-of-distribution samples in VAEs is a difficult task.[75] To ensure that the domain of applicability extends to the area of chemical space relevant to a new project, researchers may want to retrain a VAE model on new data, but how accessible is this retraining in terms of hardware specifications, time, and hyperparameter tuning? Hyperparameter tuning is the act of tweaking the parameters controlling the training process of a ML model. We hope authors of publications describing novel chemical VAE architectures will discuss the following questions:

1. How much RAM memory would it take to retrain the model on n data points (i.e., can the model be retrained on a regular laptop, or does it require a high-performance cluster?).
2. How much time did it take to train the model, and on which hardware specifications? If possible, how long would it approximately take on a laptop PC/MacBook?
3. What is the domain of applicability? If an encoded molecule does not decode to the original molecule how would one know if this is due to inherent stochasticity or because the queried molecule is beyond the scope of the model?
4. Training a VAE on the same datasets as previous work eases comparison, but how does the accuracy and domain of applicability change when smaller datasets are used?
5. How were the hyperparameters chosen? What approach do you suggest to other researchers wanting to use your method on a different dataset?

Providing estimates of training time can be quite valuable as this gives the reader an idea of the order of magnitude to expect if they were to attempt to reproduce the results. The work of the Aspuru-Guzik group, reported in reference[66] used 108,000 and 250,000 molecules from the QM9[76] and ZINC[77] databases, respectively, and many subsequent approaches used the same datasets to ease comparison. Recent work has shown that it is possible to set up a VAE with as little as 2500 molecules (training for 30 h on a single GPU) which can achieve comparable accuracy on predicting $\log(P)$ to VAEs which use hundreds of thousands of data points.[78] Of course, using a smaller training dataset also shrinks the domain of applicability. We would expect chemical VAEs to continue developing, requiring smaller datasets and less training time without sacrificing domain of applicability, and hope that the questions above will reduce the barriers to entry for new researchers interested in chemical VAEs.

In the 2500 molecule VAE example described above,[78] the encoder takes a SMILES string as input and turns it into a continuous representation which can be used to predict $\log(P)$. It is worth noting that for applications like this, no

decoder is required, since there is no need to decode novel points in the latent space into SMILES strings. A VAE, by definition, consists of an encoder and a decoder, both of which are necessary for training, and the quality of a VAE may be judged by its ability to recreate its input. However, if only a well-trained encoder is required, is this the correct metric by which to judge a VAE? An inefficient decoder which rarely produces useful SMILES strings may still have value for training an encoder, and it is possible that the best encodings for a task, such as property prediction, are not ones which work well with decoders.

6.2 | Beyond string representations in generative models

Using SMILES strings in generative models is increasingly widespread. However, more sophisticated representations are also being developed. In addition to the specific issues with SMILES mentioned above, there are also crucial features of molecules that string representations cannot capture, such as 3D configuration which can be particularly important for biological applications, for example, when molecules interact with enzymes/receptors in the body. 2D/3D representations may specify the coordinates of the atoms in the molecule, but the syntax used to specify how atoms are connected is something models must learn, and this may result in generative models proposing chemically invalid molecules. One way to overcome the chemical invalidity issue might be to use the Junction Tree VAE, which generates molecular graphs by sequentially adding chemically valid functional groups to a molecular backbone (known as fragment-by-fragment molecular generation), as opposed to adding atoms one at a time (known as node-by-node molecular generation).[79] This will lead to more robust, though less flexible, molecular generation. Generative models using molecular graphs have become popular in recent years, leading to a wide range of such methods being developed, for example, using molecular hypergraph grammar[80] and graph neural networks.[81]

6.2.1 | Molecule generation in 3D

Real molecules exist in three dimensions, so a representation in fewer dimensions must necessarily incur a loss of information, which may or may not be relevant to the task at hand. Molecules with multiple low-energy conformations also cannot be adequately represented simply with a single static 2D/3D representation. A molecule represented in 3D space would have different coordinates following translation and/or rotation, despite still being the same molecule, and answering even the simple question of identicality of two molecules can be computationally expensive. Tensor field NNs, which are locally equivariant to rotations, translations, and permutations in 3D, have recently been introduced, and they have been shown to be capable of handling molecular structures (when molecules are treated as 3D point-clouds).[82] There are also a number of examples of 3D molecular representations being developed to be compatible with CNNs.[83,84] Deep learning has been shown to make accurate predictions about biological function from electron density fields and electrostatic potential fields.[85] Representing molecules in 3D adds additional degrees of freedom, and while this may be a good thing, because it allows the representation to more closely align with the real world, it also would require more data to yield a well-trained model. We believe access to high-quality standardized data is one of the most significant bottle-necks in computational chemistry and drug discovery[86,87] today, and while this issue is indeed receiving much attention (e.g., with the Open Reaction Database initiative[88]), we are cautious about methods which would require more data to work well, rather than less.

6.2.2 | Challenges with new techniques

As with most new fields, rigorous comparison of new and old techniques can be difficult due to the lack of an agreed upon standard to test models against. This has led to a subtle bias toward demonstrating that novel methods outperform existing techniques, an outcome which sometimes proves difficult to reproduce.[89] Some meta-work has been done on GANs which also found reproducibility to be a key issue, in part due to the challenging nature of training GANs, which often requires neural architecture engineering, excessive hyperparameter tuning, and nontrivial "tricks" — all of which are non-standardized.[90] Indeed, it has been demonstrated that with enough hyperparameter tuning and random restarts, most GAN models reach similar results.[91] Similar issues regarding lack of reproducibility have also been demonstrated within reinforcement learning.[92]

7 | DISCUSSION

As each molecular representation has its own (dis)advantages, it follows that the ideal representation will depend on the task. String and chemical table representations are invaluable for communication, as these can most accurately convey the underlying structure of the molecule which is being represented. However, their discrete nature makes them difficult to use as much more than a label for the molecule. This is especially true for registry system representations such as CAS RNs, which themselves contain no information about the underlying structure of the molecule, and instead represent a link to the relevant record in a database, which does contain a great deal of (structural) information. Answering interesting questions about molecules requires a numerical description of the structure or the molecular features to allow computational handling, and this also true when ML is used to solve a problem.

Loosely speaking, the ECFP can be thought of as a vector containing 1 s and 0 s according to which substructures are present in a particular molecule, and this makes fingerprints useful for representing molecules in ML models when predicting variables which depend largely on the molecular structure. 2D fingerprint-based models have been shown to perform equally well to state-of-the-art 3D structure-based models on a variety of tasks, such as predicting partition coefficients, toxicity, and solubility, though falling short of the 3D methods when predicting complex-based protein-ligand binding affinity.[93] For a task such as solvent selection, the features resulting from the molecular structure may be more relevant to use than an embedded representation of the structure itself.[52] Just as the mathematics of ML algorithms dictate how they may rationally be used, so too must chemical knowledge be incorporated in choosing the representation. The ECFP and computer learned representations focus on the structure of the molecule to yield a numerical representation. One particular issue with using the ECFP, which may not also be problematic with a computer learned representation, is the loss of structural connectivity in the molecular representation. ECFPs reflect which substructures are present in a molecule, but the interconnectedness (particularly over large distances) is lost. In contrast, computer learned representations lack interpretability. We know of only one study rigorously comparing molecular fingerprints and descriptors to the newer methods for learning molecular embeddings; the comparison task was quantitative structure–activity relationship modeling on a variety of datasets, and interestingly it was found that the embedded representations generated by deep learning methods did not significantly outperform the more traditional molecular representations.[94] Whether deep learning will emerge as superior for molecular representation remains to be seen.

A number of techniques for de novo molecule generation with favorable properties have been discussed herein. Generative algorithms may often suggest synthetically inaccessible or otherwise unrealistic molecules, so screening molecule libraries is still an important method for identifying potential "hits." Numerous methods for generating and handling libraries exist, such as BRICS[95] and RECAP[96] which break retrosynthetically interesting compounds into fragments which can be combinatorally recombined, DOGS[97] for the de novo design of drug-like molecules using a ligand-based strategy, and combining CoLibri with FTrees-FS[98] for chemical space creation and similarity search.

Screening a library of molecules involves predicting or otherwise evaluating target properties for each molecule in the dataset to find molecules with an attractive property profile. The probability of finding suitable molecules will of course increase the larger the library, and this has led to an explosion in the size of molecule libraries, perhaps the largest of which being the proprietary GSK XXL database which contains 10^{26} molecules. When dealing with databases of this magnitude, efficient navigation of the chemical space is crucial.[99] It is possible to search upwards of 400 million molecules per second, though with linear scaling: $O(n)$. The scaling behavior of an algorithm deals with how much additional time it would take to handle more data points; linear scaling implies that doubling the number of data points would also double the time needed for computation. Development of sublinear scaling algorithms would allow much faster handling of these massive databases, and is still an active area of research; see for example NextMove Software's SmallWorld.[100,101] Although it is theoretically possible for molecule databases to get even larger, they are already at a near unmanageable size; at 400 million molecules/second it would take 10^9 years to screen the whole of the GSK XXL database. This highlights a clear need for new and faster algorithms such that the field may transition from brute force screening to intelligent and guided search.[102]

An aspect of molecular representation that also deserves more attention is stereochemistry: molecules which have the same atoms and same connectivity but are distinct species. Simple examples include E-but-2-ene and Z-but-2-ene. SMILES strings are not capable of distinguishing stereoisomers, implying that any ML method relying on SMILES cannot distinguish stereoisomers either. InChI strings have stereochemical representation built in with the stereochemcial layer, though they have not, to date, been used in ML workflows as often as SMILES. Consideration of stereochemistry with SMILES is possible with the three stereochemical descriptors ("@," "/," and "\"), and a few different approaches

to stereochemical SMILES exist: InChIfied SMILES,[22] Jmol SMILES and Jmol SMARTS,[103] ChemAxon Extended SMILES,[104] and RDChiral.[27]

Encoding stereochemical information in 2D graph representations is also not trivial, as there is no coordinate information in the third dimension; stereochemical handling has successfully been built into graph-based canonicalization algorithms.[105] Stereochemistry has thus far largely been ignored in generative models such as molecular VAEs, in an attempt to keep the representation syntax a bit simpler. In the development of the Junction Tree VAE[79] it was indeed empirically found that considering stereochemistry during molecular generation was not as efficient as splitting molecule generation and stereochemical handling into two separate steps. In the stereochemical handling step RDKit's EnumerateStereoisomers generated all possible stereoisomers; each stereoisomer was then encoded using the VAE encoder, and the stereoisomer selected was the one with the highest cosine similarity to the latent representation of the query molecule. However, the empirical finding that stereochemical handling is more efficient as a separate step to molecule generation does not imply that this is true in general. In an MDL molfile the coordinates of all atoms are given in all three dimensions together with information of the interatomic bonding, which may aid in representing stereochemistry. In addition, the fifth number in the counts line (see Figure 2) specifies whether the molecule is [1] or is not [0] chiral, while the fourth number for each atom in the bond block specifies whether the bond is in line with the page [0], pointing toward you [1], or pointing into the page [6].

ML models are notoriously data hungry, and this is doubly true in chemistry due to the sparse nature of organic compounds and their reactivity (the number of possible organic molecules is near infinite). Therefore, using ML models for predictions in chemistry requires a large amount of data, highly descriptive features, and/or a constriction of the chemical space. The prediction of stereoselective organic and organometallic catalysis with only small datasets available is an example of a task which might require highly descriptive features, for example, in the form of hand-crafted descriptors which can incorporate mechanistic knowledge and account for complicated 3D-conformations. As our ability to do density functional theory calculations is increasingly automated, and more flexible molecular representations are developed further, semi-automatic methods not relying on hand-crafted descriptors may soon rival expert-curated feature sets even on complex prediction tasks, particularly as data availability grows.[106]

Although this work is intended as an introduction to and comparison of molecular representation in machine readable format, and, in particular, how these various representations can interact with ML methods, it is worth noting that many of the representations that researchers use today have moved beyond what is described herein. Before moving on to state-of-the-art it is important to grasp the basics, and understanding differences and similarities between representations based on time, training, and precision will aid in the selection of representation for your project. The needs and culture of different research fields can have a large influence of what is considered the "gold standard," and a degree of uniformity within a field can ease cooperation. However, methodological uniformity, and the drawbacks/benefits associated with particular techniques, can influence the direction of the research itself. One thing known to aid advancements within a field is the development of standardized problems which allows for fair benchmarking of various methods, similar to the MNIST[107,108] dataset for computer vision. Various datasets for benchmarking ML algorithms in cheminformatics do exist, such as ALChemy[109] and QM9,[110,111] though the diversity of tasks and vastness of chemical space means that benchmarking is still a challenge.

8 | CONCLUSIONS

The effective representation of molecules is imperative for most chemical problems, and given the increasingly complex interactions between established representations and ML, accessible material on this topic is crucial for lowering the barriers to entry. A wide range of molecular representations has been introduced across four categories: string, chemical table, feature-based, and computer-learned representations. Although the issue of representing molecules for communication between humans has largely been solved, we believe that the advent of ML has sparked renewed efforts in attempting to refine molecular representation such that we can feed models with information which enable them to predict, extrapolate, and ultimately solve important problems in chemistry. The simplest way of representing a molecule within a model is with a one-hot encoding, relying on either descriptors about the molecule or vast amounts of data to arrive at a well-trained model. Although SMILES and InChI strings and MDL molfiles all represent the structure of the molecule, these representations are not in a format which is directly compatible with a prediction model, since models typically must have strictly numeric inputs. Various approaches have been developed to arrive at a numeric representation of molecules of which the ECFP and computer-learned representations were discussed in detail. ECFPs

have proven useful in a wide range of scenarios, though their sparse nature and large size can make them unsuitable in regimes with low data availability. It is generally not advisable to feed a model with data having more input dimensions than there are data points, as this might lead to overfitting, and excessive "folding" of the ECFP to arrive at a smaller input vector may deteriorate the quality of the fingerprint. An alternative to folding could be using a dimensionality reduction method, such as principal component analysis. Morgan fingerprints may be a good place to start for a wide variety of tasks given their track-record, ease of use (RDKit has extensive and easy-to-follow documentation), and light-weight nature; a Morgan fingerprint can be calculated in mere milliseconds and computation time scales linearly with the number of fingerprint calculations.

The use of VAEs for generating continuous representations of molecules is an exciting new development, and the vast number of papers presenting new ideas since the idea was first presented in the beginning of 2018[66] speaks both to the high expectations in the community for this method, and also that it will likely require much more work before it becomes clear how to best put such a model together. The first chemical VAE was trained on hundreds of thousands of molecules, and many subsequent papers trained on the same datasets for ease of comparison, though recent work would suggest that a well-trained VAE can be set up with as little as 2500 molecules. The effort put into making finished models easily available for others to use through platforms such as GitHub is commendable. To aid reproducibility we believe features such as hardware specifications, training time, and model domain of applicability deserves more detailed mention and are too often implicit.

Although preliminary results certainly are interesting, current research efforts mostly are focused on improving the representation method, rather than exploring applications. For this reason, we believe it is too early to attempt to predict how it will change the industry, though we are cautiously optimistic that this new class of representation will bring about a wave of new discoveries.

ACKNOWLEDGMENTS

This work is co-funded by UCB Pharma and Engineering and Physical Sciences Research Council via project EP/S024220/1 EPSRC Centre for Doctoral Training in Automated Chemical Synthesis Enabled by Digital Molecular Technologies'.

CONFLICT OF INTEREST

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

AUTHOR CONTRIBUTIONS

Daniel S. Wigh: Conceptualization (lead); investigation (lead); writing—original draft (lead). **Jonathan M. Goodman:** Supervision (equal); writing—review and editing (equal). **Alexei A. Lapkin:** Funding acquisition (lead); project administration (lead); supervision (equal); writing—review and editing (supporting).

DATA AVAILABILITY STATEMENT

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

ORCID

Daniel S. Wigh https://orcid.org/0000-0002-0494-643X
Jonathan M. Goodman https://orcid.org/0000-0002-8693-9136
Alexei A. Lapkin https://orcid.org/0000-0001-7621-0889

RELATED WIREs ARTICLES

Machine learning methods in chemoinformatics
Representation of chemical structures

REFERENCES

1. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596:583–9.
2. Zhang T, Li H, Xi H, Stanton RV, Rotstein SH. HELM: a hierarchical notation language for complex biomolecule structure representation. J Chem Inf Model. 2012;52:2796–806.
3. Warr WA. Representation of chemical structures. WIREs Comput Mol Sci. 2011;1:557–79.

4. Giganti D, Guillemain H, Spadoni J-L, Nilges M, Zagury J-F, Montes M. Comparative evaluation of 3D virtual ligand screening methods: impact of the molecular alignment on enrichment. J Chem Inf Model. 2010;50:992–1004.
5. NCIB. PubChem Compound Summary for CID 441344, Dicyclomine Hydrochloride. Available from: https://pubchem.ncbi.nlm.nih.gov/compound/441344.
6. Sayle, R. A.; O'Boyle, N. M.; Landrum, G. A.; Affentranger, R. Open sourcing a Wiswesser Line Notation (WLN) parser to facilitate electronic lab notebook (ELN) record transfer using the Pistoia alliance's UDM (Unified Data Model) standard. 2019. Available from: https://www.nextmovesoftware.com/talks.html.
7. Dittmar PG, Farmer NA, Fisanick W, Haines RC, Mockus J. The CAS ONLINE search system. 1. General system design and selection, generation, and use of search screens. J Chem Inf Comput Sci. 1983;23:93–102.
8. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, et al. PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res. 2021;49:D1388–95.
9. Pence HE, Williams A. ChemSpider: an online chemical information resource. J Chem Educ. 2010;87:1123–4.
10. Mendez D, Gaulton A, Bento AP, Chambers J, de Veij M, Félix E, et al. ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Res. 2019;47:D930–40.
11. Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, et al. The ChEMBL database in 2017. Nucleic Acids Res. 2017;45:D945–54.
12. Davies M, Nowotka M, Papadatos G, Dedman N, Gaulton A, Atkinson F, et al. ChEMBL web services: streamlining access to drug discovery data and utilities. Nucleic Acids Res. 2015;43:W612–20.
13. Wiswesser WJ. Thw (sic) Wiswesser line formula notation. Chem Eng News Arch. 1952;30:3523–6.
14. Wiswesser WJ. 107 Years of line-formula notations (1861–1968). J Chem Doc. 1968;8:146–50.
15. Wiswesser WJ. How the WLN began in 1949 and how it might be in 1999. J Chem Inf Comput Sci. 1982;22:88–93.
16. Lynch MF, Barnard JM, Welford SM. Generic structure storage and retrieval. J Chem Inf Comput Sci. 1985;25:264–70.
17. Derwent C. Derwent World Patents Index; London, England: Clarivate; 2020. p. 311 ISBN: 1 903836 37 4 (Revised Edition 1).
18. Favre HA, Powell WH. Nomenclature of organic chemistry: IUPAC recommendations and preferred names 2013. Cambridge, England: The Royal Society of Chemistry; 2014.
19. Lowe DM, Corbett PT, Murray-Rust P, Glen RC. Chemical name to structure: OPSIN, an open source solution. J Chem Inf Model. 2011;51:739–53.
20. Weininger DS. A chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci. 1988;28:31–6.
21. Daylight Chemical Information Systems Inc. 3. SMILES—A Simplified Chemical Language. 2019; [Cited 2019 Dec 9]. Available from: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.
22. O'Boyle NM. Towards a universal SMILES representation—a standard method to generate canonical SMILES based on the InChI. J Chem. 2012;4:22.
23. Schneider N, Sayle RA, Landrum GA. Get your atoms in order—an open-source implementation of a novel and robust molecular canonicalization algorithm. J Chem Inf Model. 2015;55:2111–20.
24. Weininger D, Weininger A, Weininger JL. SMILES. 2. Algorithm for generation of unique SMILES notation. J Chem Inf Comput Sci. 1989;29:97–101.
25. Corey EJ, Wipke WT. Computer-assisted design of complex organic syntheses. Science. 1969;166:178–92.
26. Christ CD, Zentgraf M, Kriegl JM. Mining electronic laboratory notebooks: analysis, retrosynthesis, and reaction based enumeration. J Chem Inf Model. 2012;52:1745–56.
27. Coley CW, Green WH, Jensen KF. RDChiral: an RDKit wrapper for handling stereochemistry in retrosynthetic template extraction and application. J Chem Inf Model. 2019;59:2529–37.
28. Daylight Chemical Information Systems Inc. 5. SMIRKS—A reaction transform language; [cited 2021 July 22]. Available from: https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html
29. Daylight Chemical Information Systems Inc. 4. SMARTS—A language for describing molecular patterns; [cited 2021 July 22]. Available from https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html
30. Segler MHS, Waller MP. Neural-symbolic machine learning for retrosynthesis and reaction prediction. Chem Eur J. 2017;23:5966–71.
31. Schwaller P, Laino T, Gaudin T, Bolgar P, Hunter CA, Bekas C, et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. ACS Cent Sci. 2019;5:1572–83.
32. Rajan K, Zielesny A, Steinbeck C. STOUT: SMILES to IUPAC names using neural machine translation. J Chem. 2021;13:34.
33. Jaeger S, Fulle S, Turk S. Mol2vec: unsupervised machine learning approach with chemical intuition. J Chem Inf Model. 2018;58:27–35.
34. O'Boyle N, Dalke A. DeepSMILES: An adaptation of SMILES for use in machine-learning of chemical structures. ChemRxiv. 2018. https://doi.org/10.26434/chemrxiv.7097960.v1
35. 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;1:1757–72.
36. Nigam A, Friederich P, Krenn M, Aspuru-Guzik A. A. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. International Conference on Learning Representations (ICLR-2020); Preprint. 2020, arXiv: 1909.11655.
37. 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;2:03LT02.

38. Nigam, A.; Pollice, R.; Aspuru-Guzik, A.: JANUS: parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. 2021, arXiv: 2106.04011.
39. Nigam A, Pollice R, Krenn M, Gomes GDP, Aspuru-Guzik A. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. Chem Sci. 2021;12:7079–90.
40. Heller SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D. InChI, the IUPAC international chemical identifier. J Chem. 2015;7:23.
41. IUPAC and InChI Trust. InChI version 1, software version 1.05 release notes. 2017. Available from: https://www.inchi-trust.org/downloads/
42. IUPAC and InChI Trust. InChI version 1, software version 1.04 release notes. 2011. Available from: https://www.inchi-trust.org/downloads/
43. IUPAC and InChI Trust. InChI version 1, software version 1.06 release notes. 2020. Available from: https://www.inchi-trust.org/downloads/
44. Goodman JM, Pletnev I, Thiessen P, Bolton E, Heller SR. InChI version 1.06: now more than 99.99% reliable. J Chem. 2021;13:40.
45. Ash S, Cline MA, Homer RW, Hurst T, Smith GB. SYBYL Line Notation (SLN): a versatile language for chemical structure representation. J Chem Inf Comput Sci. 1997;37:71–9.
46. Gasteiger J. Handbook of Chemoinformatics. Weinheim, Germany: John Wiley & Sons, Inc; 2008.
47. Biova Databases. CTFile Formats. 2016. Available from: http://help.accelrysonline.com/ulm/onelab/1.0/content/ulm_pdfs/direct/reference/ctfileformats2016.pdf
48. Dalby A, Nourse JG, Hounshell WD, Gushurst AKI, Grier DL, Leland BA, et al. Description of several chemical structure file formats usedby computer programs developed at Molecular Design Limited. J Chem Inf Comput Sci. 1992;32:244–55.
49. Clark AM. Accurate specification of molecular structures: the case for zero-order bonds and explicit hydrogen counting. J Chem Inf Model. 2011;51:3149–57.
50. Clark AM, Williams AJ, Ekins S. Machines first, humans second: on the importance of algorithmic interpretation of open chemistry data. J Chem. 2015;7:9.
51. PerkinElmer, Inc. CDX format specification; [cited 2019 Dec 13]. Available from: http://www.cambridgesoft.com/services/documentation/sdk/chemdraw/cdx/
52. Amar Y, Schweidtmann AM, Deutsch P, Cao L, Lapkin A. Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. Chem Sci. 2019;10:6697–706.
53. Durand DJ, Fey N. Computational ligand descriptors for catalyst design. Chem Rev. 2019;119:6561–94.
54. Shields BJ, Stevens J, Li J, Parasram M, Damani F, Alvarado JIM, et al. Bayesian reaction optimization as a tool for chemical synthesis. Nature. 2021;590:89–96.
55. Moriwaki H, Tian Y-S, Kawashita N, Takagi T. Mordred: a molecular descriptor calculator. J Chem. 2018;10:4.
56. Ahneman DT, Estrada JG, Lin S, Dreher SD, Doyle AG. Predicting reaction performance in C–N cross-coupling using machine learning. Science. 2018;360:186–90.
57. Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF. Convolutional embedding of attributed molecular graphs for physical property prediction. J Chem Inf Model. 2017;57:1757–72.
58. Pogány P, Arad N, Genway S, Pickett SD. De novo molecule design by translating from reduced graphs to SMILES. J Chem Inf Model. 2019;59:1136–46.
59. Bremser W. Hose—a novel substructure code. Anal Chim Acta. 1978;103:355–65.
60. Dubois JE, Panaye A, Attias R. DARC system: notions of defined and generic substructures. Filiation and coding of FREL substructure (SS) classes. J Chem Inf Comput Sci. 1987;27:74–82.
61. Cereto-Massagué A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallvé S, Pujadas G. Molecular fingerprint similarity search in virtual screening. Methods. 2015;71:58–63.
62. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. 2010;50:742–54.
63. Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of MDL keys for use in drug discovery. J Chem Inf Comput Sci. 2002;42:1273–80.
64. Landrum, G. The RDKit documentation; [Cited 2020 Jan 10]. Available from: https://www. rdkit.org/docs/
65. Gao H, Struble TJ, Coley CW, Wang Y, Green WH, Jensen KF. Using machine learning to predict suitable conditions for organic reactions. ACS Cent Sci. 2018;4:1465–76.
66. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez- Lengeling B, Sheberla D, et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent Sci. 2018;4:268–76.
67. Elton DC, Boukouvalas Z, Fuge MD, Chung PW. Deep learning for molecular design—a review of the state of the art. Mol Syst Design Eng. 2019;4:828–49.
68. Kingma DP, Welling M. Auto-encoding variational Bayes. The 2nd International Conference on Learning Representations (ICLR). 2013; Scottsdale, AZ.
69. 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.
70. Neil D, Segler M, Guasch L, Ahmed M, Plumbley D, Sellwood M, Brown N. Exploring deep recurrent models with reinforcement learning for molecule design. International Conference on Learning Representations (ICLR) workshop. 2018; Vancouver, Canada.
71. Tetko IV, Karpov P, Bruno E, Kimber TB, Godin G. Augmentation is what you need! Artificial neural networks and machine learning—ICANN 2019: workshop and special sessions, Cham, 2019. p. 831–835.

72. Winter R, Montanari F, Noé F, Clevert D-A. Learning continuous and datadriven molecular descriptors by translating equivalent chemical representations. Chem Sci. 2019;10:1692–701.
73. Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for structured data. International Conference on Learning Representations (ICLR). 2018; Vancouver, Canada.
74. Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. International Conference on Machine Learning (ICML). 2017; Sydney, Australia.
75. Xiao, Z.; Yan, Q.; Amit, Y. Likelihood regret: an out-of-distribution detection score for variational auto-encoder. 34th Conference on Neural Information Processing Systems (NeurIPS); 2020.
76. Ramakrishnan R, Dral PO, Rupp M, von Lilienfeld OA. Quantum chemistry structures and properties of 134 kilo molecules. Sci Data. 2014;1:140022.
77. Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG. ZINC: a free tool to discover chemistry for biology. J Chem Inf Model. 2012;52:1757–68.
78. Grosnit A, Tutunov R, Maraval AM, Griffiths R-R, Cowen-Rivers AI, Yang L, Zhu L, Lyu W, Chen Z, Wang J, Peters J & Bou-Ammar H. High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning. Preprint. 2021, arXiv: 2106.03609.
79. Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. Proceedings of the 35th International Conference on Machine Learning (ICML); 2019; Stockholm, Sweden.
80. Kajino H. Molecular hypergraph grammar with its application to molecular optimization. International Conference on Machine Learning (ICML); 2019; Long Beach, California, arXiv:1809.02745.
81. Bongini P, Bianchini M, Scarselli F. Molecular graph generation with graph neural networks. Neurocomputing. 2021;450:242–52.
82. Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, Riley P. Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds. Preprint. 2018, arXiv:1802.08219.
83. Kajita S, Ohba N, Jinnouchi R, Asahi R. A universal 3D voxel descriptor for solid-state material informatics with deep convolutional neural networks. Sci Rep. 2017;7:16991.
84. Kuzminykh D, Polykovskiy D, Kadurin A, Zhebrak A, Baskov I, Nikolenko S, et al. 3D molecular representations based on the wave transform for convolutional neural networks. Mol Pharm. 2018;15:4378–85.
85. Golkov V, Skwark MJ, Mirchev A, Dikov G, Geanes AR, Mendenhall J, Meiler J, Cremers D. 3D deep learning for biological function prediction from physical fields. International Conference on 3D Vision (3DV). 2017.
86. Bender A, Cortés-Ciriano I. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 1: ways to make an impact, and why we are not there yet. Drug Discov Today. 2021;26:511–24.
87. Bender A, Cortes-Ciriano I. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 2: a discussion of chemical and biological data. Drug Discov Today. 2021;26:1040–52.
88. Kearnes SM, Maser MR, Wleklinski M, Kast A, Doyle AG, Dreher SD, et al. The open reaction database. J Am Chem Soc. 2021;143: 18820–6.
89. Melis G, Dyer C, Blunsom P. On the state of the art of evaluation in neural language models. International Conference on Learning Representations (ICLR). 2017; Toulon, France.
90. Kurach K, Lucic M, Zhai X, Michalski M, Gelly S. A large-scale study on regularization and normalization in GANs. Proceedings of the 36th international conference on machine learning (ICML); 2019. Long Beach, CA.
91. Lucic M, Kurach, K, Michalski M, Gelly S, Bousquet O. Are GANs created equal? A large-scale study. 32nd Conference on Neural Information Processing Systems (NeurIPS). 2018; Montréal, Canada.
92. Henderson P, Islam R, Bachman P, Pineau J, Precup D, Meger D. Deep reinforcement learning that matters. AAAI Conference on Artificial Intelligence (AAAI). 2019; Honolulu, HI.
93. Gao K, Duy Nguyen D, Sresht V, Mathiowetz AM, Tu M, Wei G-W. Are 2D fingerprints still valuable for drug discovery? Phys Chem Chem Phys. 2020;22:8373–90.
94. Sabando MV, Ponzoni I, Milios EE, Soto AJ. Using molecular embeddings in QSAR modeling: does it make a difference? Brief Bioinform. 2021;23:1–21.
95. Degen J, Wegscheid-Gerlach C, Zaliani A, Rarey M. On the art of compiling and using "drug-like" chemical fragment spaces. ChemMedChem. 2008;3:1503–7.
96. Lewell XQ, Judd DB, Watson SP, Hann MM. RECAP—retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. J Chem Inf Comput Sci. 1998;38:511–22.
97. Hartenfeller M, Zettl H, Walter M, Rupp M, Reisen F, Proschak E, et al. DOGS: reaction-driven de novo design of bioactive compounds. PLoS Comput Biol. 2012;8:e1002380.
98. Boehm M, Wu T-Y, Claussen H, Lemmen C. Similarity searching and scaffold hopping in synthetically accessible combinatorial chemistry spaces. J Med Chem. 2008;51:2468–80.
99. Hoffmann T, Gastreich M. The next level in chemical space navigation: going far beyond enumerable compound libraries. Drug Discov Today. 2019;24:1148–56.
100. Software N. SmallWorld. Cambridge, England: NextMove Software; 2021. Available from: https://www.nextmovesoftware.com/smallworld.html.
101. Irwin JJ, Tang KG, Young J, Dandarchuluun C, Wong BR, Khurelbaatar M, et al. ZINC20—a free ultralarge-scale chemical database for ligand discovery. J Chem Inf Model. 2020;60:6065–73.

102. Warr W. Report on an NIH workshop on Ultralarge Chemistry Databases. 2021; ChemRxiv. Available from: 10.26434/chemrxiv.14554803.v1.
103. Hanson RM. Jmol SMILES and Jmol SMARTS: specifications and applications. J Chem. 2016;8:50.
104. ChemAxon, ChemAxon SMILES extensions. ChemAxon Docs. Available from: https://docs.chemaxon.com/display/docs/chemaxon-smiles-extensions.md
105. Koichi S, Iwata S, Uno T, Koshino H, Satoh H. Algorithm for advanced canonical coding of planar chemical structures that considers stereochemical and symmetric information. J Chem Inf Model. 2007;47:1734–46.
106. Gallegos LC, Luchini G, St John PC, Kim S, Paton RS. Importance of engineered and learned molecular representations in predicting organic reactivity, selectivity, and chemical properties. Acc Chem Res. 2021;54:827–36.
107. LeCun Y, Cortes C, Burges CJ. The MNIST database of handwritten digits; [Cited 2021 April 27]. Available from: http://yann.lecun.com/exdb/mnist/.
108. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86:2278–324.
109. Chen G, Chen P, Hsieh C-Y, Lee C-K, Liao B, Liao, R, Liu W, Qiu J, Sun Q, Tang J, Zemel R, Zhang S. Alchemy: a quantum chemistry dataset for benchmarking AI models. International Conference on Learning Representations (ICLR). 2019; New Orleans, LA.
110. Rappe AK, Casewit CJ, Colwell KS, Goddard WA, Skiff WM. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. J Am Chem Soc. 1992;114:10024–35.
111. Guha R, Howard MT, Hutchison GR, Murray-Rust P, Rzepa H, Steinbeck C, et al. The blue obelisk—interoperability in chemical informatics. J Chem Inf Model. 2006;46:991–8.

How to cite this article: Wigh DS, Goodman JM, Lapkin AA. A review of molecular representation in the age of machine learning. WIREs Comput Mol Sci. 2022;12:e1603. https://doi.org/10.1002/wcms.1603