

## REVIEW

# Inverse molecular design using machine learning: Generative models for matter engineering

Benjamin Sanchez-Lengeling<sup>1</sup> and Alán Aspuru-Guzik<sup>2,3,4\*</sup>

The discovery of new materials can bring enormous societal and technological progress. In this context, exploring completely the large space of potential materials is computationally intractable. Here, we review methods for achieving inverse design, which aims to discover tailored materials from the starting point of a particular desired functionality. Recent advances from the rapidly growing field of artificial intelligence, mostly from the subfield of machine learning, have resulted in a fertile exchange of ideas, where approaches to inverse molecular design are being proposed and employed at a rapid pace. Among these, deep generative models have been applied to numerous classes of materials: rational design of prospective drugs, synthetic routes to organic compounds, and optimization of photovoltaics and redox flow batteries, as well as a variety of other solid-state materials.

Many of the challenges of the 21st century (1), from personalized health care to energy production and storage, share a common theme: materials are part of the solution (2). In some cases, the solutions to these challenges are fundamentally limited by the physics and chemistry of a material, such as the relationship of a materials bandgap to the thermodynamic limits for the generation of solar energy (3).

Several important materials discoveries arose by chance or through a process of trial and error. For example, vulcanized rubber was prepared in the 19th century from random mixtures of compounds, based on the observation that heating with additives such as sulfur improved the rubber's durability. At the molecular level, individual polymer chains cross-linked, forming bridges that enhanced the macroscopic mechanical properties (4). Other notable examples in this vein include Teflon, anesthesia, Vaseline, Perkin's mauve, and penicillin. Furthermore, these materials come from common chemical compounds found in nature. Potential drugs either were prepared by synthesis in a chemical laboratory or were isolated from plants, soil bacteria, or fungus. For example, up until 2014, 49% of small-molecule cancer drugs were natural products or their derivatives (5).

In the future, disruptive advances in the discovery of matter could instead come from unexplored regions of the set of all possible molecular and solid-state compounds, known as chemical space (6, 7). One of the largest collections of molecules, the chemical space project (8), has

mapped 166.4 billion molecules that contain at most 17 heavy atoms. For pharmacologically relevant small molecules, the number of structures is estimated to be on the order of  $10^{60}$  (9). Adding consideration of the hierarchy of scale from sub-nanometer to microscopic and mesoscopic further complicates exploration of chemical space in its entirety (10). Therefore, any global strategy for covering this space might seem impossible.

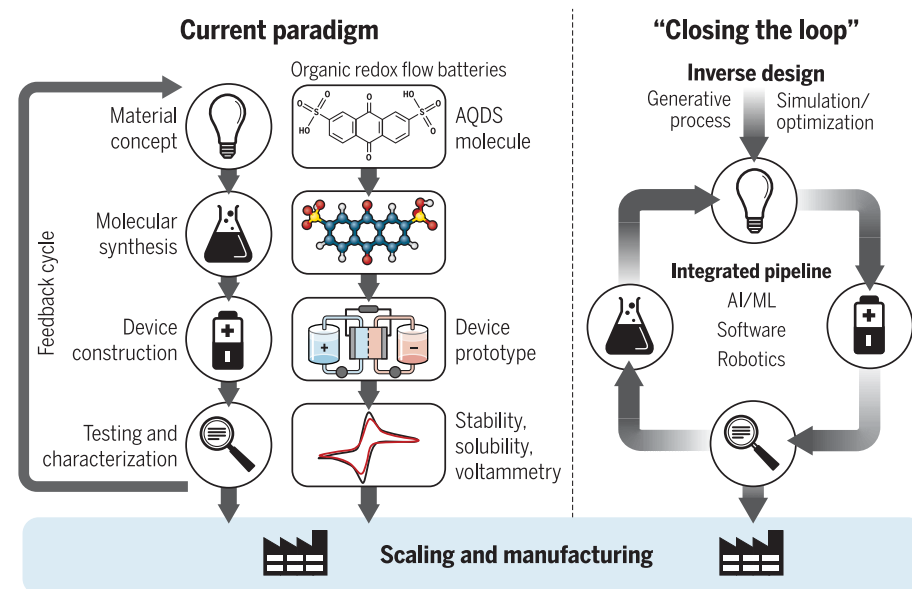
Simulation offers one way of probing this space without experimentation. The physics and chemistry of these molecules are governed by quantum mechanics, which can be solved via the Schrödinger equation to arrive at their ex-

act properties. In practice, approximations are used to lower computational time at the cost of accuracy.

Although theory enjoys enormous progress, now routinely modeling molecules, clusters, and perfect as well as defect-laden periodic solids, the size of chemical space is still overwhelming, and smart navigation is required. For this purpose, machine learning (ML), deep learning (DL), and artificial intelligence (AI) have a potential role to play because their computational strategies automatically improve through experience (11). In the context of materials, ML techniques are often used for property prediction, seeking to learn a function that maps a molecular material to the property of choice. Deep generative models are a special class of DL methods that seek to model the underlying probability distribution of both structure and property and relate them in a nonlinear way. By exploiting patterns in massive datasets, these models can distill average and salient features that characterize molecules (12, 13).

Inverse design is a component of a more complex materials discovery process. The time scale for deployment of new technologies, from discovery in a laboratory to a commercial product, historically, is 15 to 20 years (14). The process (Fig. 1) conventionally involves the following steps: (i) generate a new or improved material concept and simulate its potential suitability; (ii) synthesize the material; (iii) incorporate the material into a device or system; and (iv) characterize and measure the desired properties. This cycle generates feedback to repeat, improve, and refine future cycles of discovery. Each step can take up to several years.

In the era of matter engineering, scientists seek to accelerate these cycles, reducing the



**Fig. 1. Schematic comparison of material discovery paradigms.** The current paradigm is outlined at left and exemplified in the center with organic redox flow batteries (92). A closed-loop paradigm is outlined at right. Closing the loop requires incorporating inverse design, smart software (93), AI/ML, embedded systems, and robotics (87) into an integrated ecosystem.

<sup>1</sup>Department of Chemistry and Chemical Biology, Harvard University 12 Oxford Street, Cambridge, MA 02138, USA.

<sup>2</sup>Department of Chemistry and Department of Computer Science, University of Toronto, Toronto Ontario, M5S 3H6, Canada. <sup>3</sup>Vector Institute for Artificial Intelligence, Toronto, Ontario M5S 1M1, Canada. <sup>4</sup>Canadian Institute for Advanced Research (CIFAR) Senior Fellow Toronto, Ontario M5S 1M1, Canada.

\*Corresponding author. Email: [aspuru@utoronto.ca](mailto:aspuru@utoronto.ca)

time between steps. The ultimate aim is to concurrently propose, create, and characterize new materials, with each component transmitting and receiving data simultaneously. This process is called “closing the loop,” and inverse design is a critical facet (12, 15).

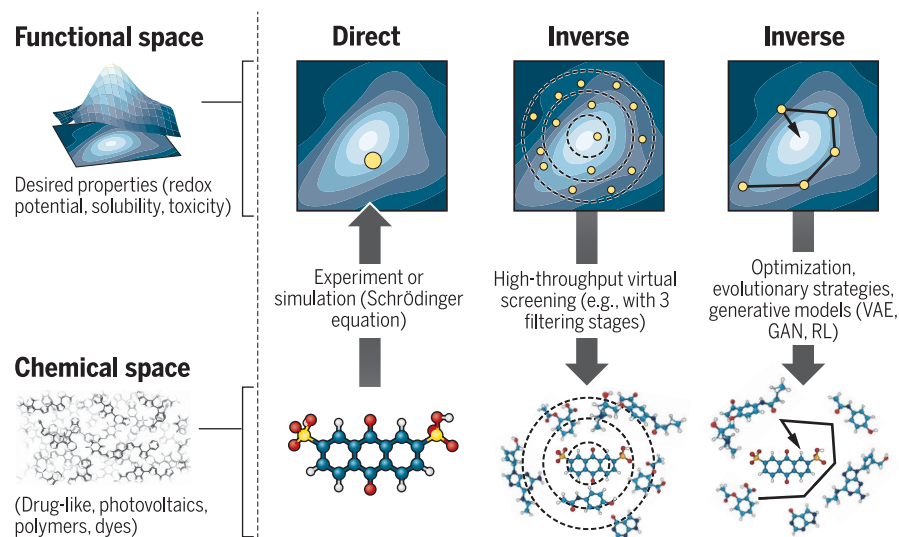
### Inverse design

Quantum chemical methods reveal properties of a molecular system only after specifying the essential parameters of the constituent atomic nuclei and their three-dimensional (3D) coordinate positions (16). Inverse design, as its name suggests, inverts this paradigm by starting with the desired functionality and searching for an ideal molecular structure. Here the input is the functionality and the output is the structure. Functionality need not necessarily map to one unique structure but to a distribution of probable structures. Inverse design (Fig. 2) uses optimization, sampling, and search methods to navigate the manifold of functionality of chemical space (17, 18).

One of the earliest efforts in inverse design was the methodology of high-throughput virtual screening (HTVS). HTVS has its roots in the pharmaceutical industry for drug discovery, where simulation is an exploratory tool for screening a large number of molecules (19, 20). HTVS starts with an initial library of molecules built on the basis of researchers’ intuition, which narrows down the pool of possible candidate molecules to a tractable range of a thousand to a million. Initial candidates are filtered on the basis of focused targets such as ease of synthesis, solubility, toxicity, stability, activity, and selectivity. Molecules are also filtered by expert opinion, eventually considered as potential lead compounds for organic synthesis. Successful motifs and substructures are further incorporated in future cycles to further optimize functionality.

Although HTVS might seem like an ensemble version of the direct approach for material design, it differs in its underlying philosophy (4). HTVS is focused on data-driven discovery, which incorporates automation, time-critical performance, and computational funnels; promising candidates are further processed by more expensive methodologies. A crucial component is feedback between theory and experiment.

The HTVS methodology has been quite successful at generating new and high-performing materials in other domains. In organic photovoltaics, molecules have been screened on the basis of frontier orbital energies and photovoltaic conversion efficiency and orbital energies (21–24). In organic redox flow batteries, redox potential, solubility, and ease of synthesis (25, 26) are prioritized. For organic light-emitting diodes, molecules have been screened for their singlet-triplet gap and photoluminescent emission (27). Massive explorations of reactions for catalysis (28) or redox potentials in biochemistry have been undertaken (28). For inorganic materials, the Materials Project (29) spawns many applications such as dielectric and optical materials (30), photoanode materials for generation of



**Fig. 2. Schematic of the different approaches toward molecular design.** Inverse design starts from desired properties and ends in chemical space, unlike the direct approach that leads from chemical space to the properties.

chemical fuels from sunlight (31), and battery electrolytes (32).

Arguably, an optimization approach is preferable to HTVS because it generally visits a smaller number of configurations when exploring the manifold of functionality. An optimization incorporates and learns geometric information of the functionality manifold, guided by general trends, directions, and curvature (17).

Within discrete optimization methods, Evolution Strategies (ES) is a popular choice for global optimization (33–35) and has been used to map chemical space (36). ES involves a structured search that incorporates heuristics and procedures inspired by natural evolution (37). At each iteration, parameter vectors (“genotypes”) in a population are perturbed (“mutated”) and their objective function value (“fitness”) evaluated. ES has been likened to hill-climbing in high-dimensional space, following the numerical finite difference across parameters that are more successful at optimizing the fitness. With appropriately designed genotypes and mutation operations, ES can be quite successful at hard optimization problems, even overcoming state-of-the-art machine learning approaches (38).

In other cases, inverse design is realized by incorporating expert knowledge into the optimization procedure, via improved Bayesian sampling with sequential Monte Carlo (39), invertible system Hamiltonians (18), deriving analytical gradients of properties with respect to a molecular system (40), optimizing potential energy surfaces of chemical systems (41), or discovering design patterns via data-mining techniques (42, 43).

Finally, another approach involves generative models stemming from the field of machine learning. Before delving into the details, it is appropriate to highlight the differences between generative and discriminative models. A discriminative model tries to determine conditional

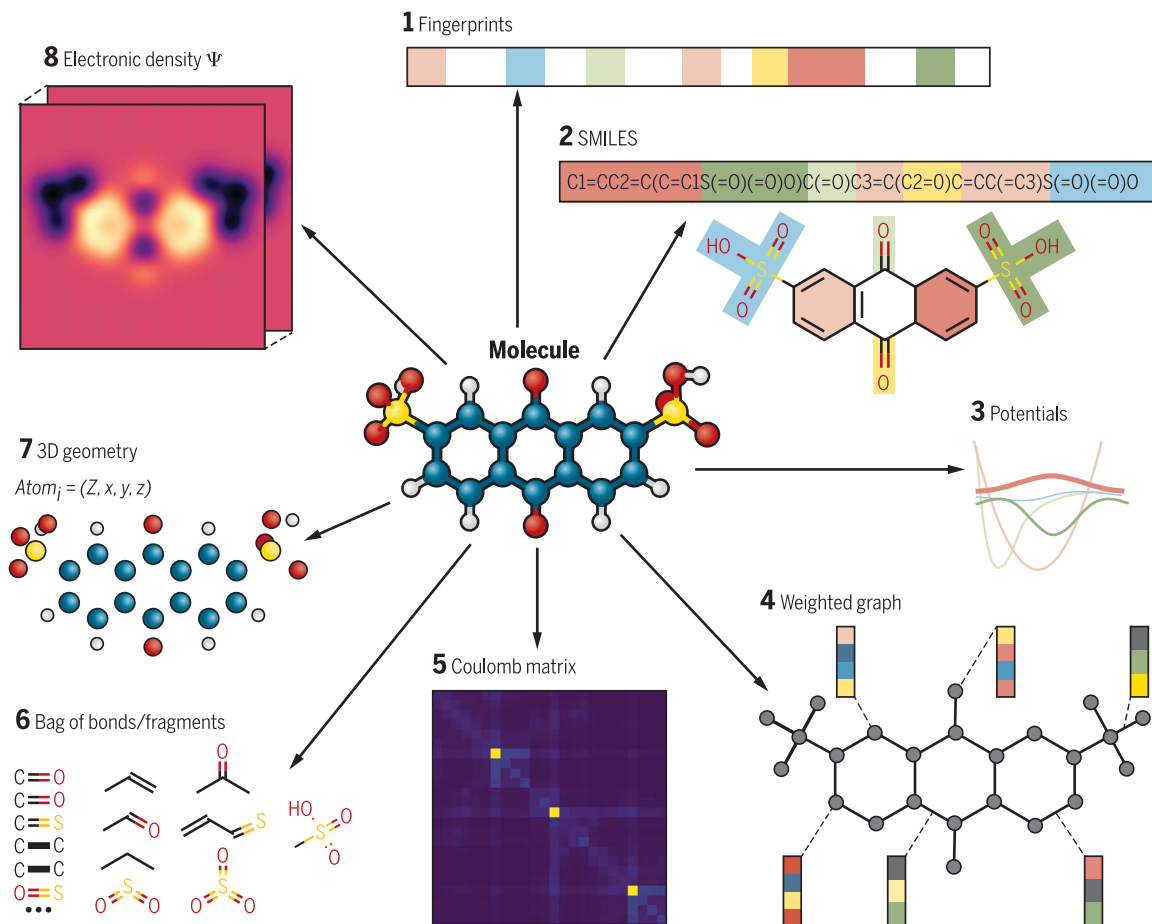
probabilities ( $p(y|x)$ ): that is, the probability of observing properties  $y$  (such as the bandgap energy or solvation energy), given  $x$  (a molecular representation). By contrast, a generative model attempts to determine a joint probability distribution  $p(x, y)$ : the probability of observing both the molecular representation and the physical property. By conditioning the probability on a molecule ( $x$ ) or a property ( $y$ ), we retrieve the notion of direct ( $p(y|x)$ ) and inverse design ( $p(x|y)$ ).

As expected, deep generative models are more challenging to create than direct ML approaches, but DL algorithms and computational strategies have advanced substantially in the last few years, producing astonishing results for generating natural-looking images (44), constructing high-quality audio waveforms containing speech (45), generating coherent and structured text (46), and most recently, designing molecules (47). There are several ways of building generative models, but for the purposes of this Review, we will focus on three main approaches: variational autoencoders (VAEs) (48), reinforcement learning (RL) (49), and generative adversarial networks (GANs) (44).

Before describing how each approach differs, we consider representations of molecules, which in turn determine the types of tools available and the types of information that can be exploited in the models.

### Representation of molecules

To model molecular systems accurately, we must solve the Schrödinger equation (SE) for the molecular electronic Hamiltonian, from which we obtain properties relating to the energy, geometry, and curvature of the potential energy surface of our system. In the SE, the molecule is represented as a set of nuclear charges and the corresponding Cartesian coordinates of the atomic positions in 3D space. Meanwhile,



**Fig. 3. Different types of molecular representations applied to one molecule, AQDS, which is used in the construction of organic redox flow batteries.** Clockwise from top: (1) A fingerprint vector that quantifies presence or absence of molecular environments; (2) SMILES strings that use simplified text encodings to describe the structure of

a chemical species; (3) potential energy functions that could model interactions or symmetries; (4) a graph with atom and bond weights; (5) Coulomb matrix; (6) bag of bonds and bag of fragments; (7) 3D geometry with associated atomic charges; and (8) the electronic density.

ML algorithms benefit from having representations that expose more easily constraints and properties of the physics of interest, so a 3D representation might not be the most efficient. Having a more direct representation allows the model to spend fewer computational resources learning patterns from first principles. A representation that can span all of chemical space should ideally capture all the symmetries of the SE: permutational, rotational, reflectional, and translational invariance for particles of the same type (50). Convolutions, Fourier transforms, and determinants are some of the mathematical structures that can preserve these symmetries and are often incorporated into the representation or model (51, 52). Molecular representation is a current open research problem; there are many representations, and no one representation seems to work for all properties (53).

Current molecular representations fall into three broad categories: discrete (e.g., text), continuous (e.g., vectors and tensors), and weighted graphs. Although graphs can be represented as sparse matrices, they differ fundamentally in how they are processed within a model. Typically

a representation will have a fixed length size via padding or the addition of dummy atoms. For inverse design, a desired property is invertibility—the capability to map back to a molecule structure that can then potentially be synthesized and characterized. Alternatively, if not invertible, it would be sufficient to have an ideal target representation and then either scan or evolve a molecule to match in a fast manner. Among invertible representations, we find molecular graphs and Hamiltonians.

Graphs are a natural representation of molecules. Following empirical principles of bonding, a molecule is interpreted as an undirected graph where each atom is a node and the bonds are the edges. To reduce complexity, hydrogen atoms are treated implicitly because they are deduced from standard chemistry valence rules. One standard for molecular graphs is SMILES strings (54), 1D text encodings that follow a particular grammar syntax. More advanced representations forgo the text encoding and use a weighted graph representation, with a variety of vectorized features on edges and nodes such as bonding type, aromaticity, charge, and distance.

(55–57). Graphs are normally not uniquely represented, which can be advantageous for data augmentation (58) or disadvantageous when this representation degeneracy introduces noise to a model (53).

Whereas Hamiltonians rely only on the known physics and atomic constants of a molecule, the Coulomb matrix representation is based on Coulombic forces between charges of each atom (59). When combined via concatenation, summation, or differences, these base representations represent reactions (60), molecular ensembles, or conformers.

Other representations are better suited to prediction and could be rendered invertible via lookup tables: bag of bonds (61), atoms (62), fingerprints (63, 64), electronic density (51), symmetry functions (65), and chemical environments (50). Figure 3 shows several of these representations.

### Generative models for exploring chemical space

Molecular representations are often inputs for deep neural network (DNN) models. The original data are transformed across several stages (layers),



usually by a linear transformation followed by a nonlinear function. For a given task and associated loss function, parameters for each layer (weights) are optimized via the backpropagation algorithm. When optimized, each intermediate (hidden) representation will tend to capture high- or low-level transformed features of the original data. In this sense, DL is a form of representation learning (66) because the DNN architecture is optimized to transform the original data into another representation that is more efficient for a given task such as regression, classification, or generation.

By attaching additional structure to the hidden representation, either in the form of statistical priors or probability distributions, we arrive at the idea of latent variable models. Each observed datum (molecule) has a corresponding latent representation, often a vector, within a latent variable space that encodes the relevant semantic features of the data.

The goal of a generative model is to model a data distribution, by training a model on large amounts of data and attempting to generate data like it. The loss function encodes the notion of likeness, measuring the differences between two distributions, the empirically observed and the generated.

We center most of our discussion on deep generative models using SMILES as a representation. Nonetheless, many of these approaches are quite general and are applicable to other representations. We expect future work to extend these architectures toward other molecular representations.

For the generation of sequences, recurrent neural networks (RNNs) (46, 67) serve as a common starting point, creating sequences incrementally one step at a time and predicting what comes next. RNNs can be augmented to take into account complex time-dependent patterns with long short-term memory cells (67, 68) (LSTMs), and attention and memory mechanisms (69). Figure 4 displays several architectures for generative models.

### Variational autoencoders, reinforcement learning, and adversarial training

Besides generation tasks, for inverse design the generative process must be controlled or biased toward desirable qualities. With VAEs, the optimization of properties is performed explicitly over a continuous representation. By comparison, with GANs and RNNs, the optimization of properties can be achieved by biasing the generation process, typically with RL by rewarding or penalizing generative behaviors.

VAEs (48) give control over the data generation via latent variables. An autoencoder (AE) model includes an encoding and a decoding network. The encoder maps the molecule to a vector in a lower-dimensional space known as the latent space, and the decoder maps the latent vector back to the original representation. The encoder acts as a compression and the decoder as a decompression operation. The AE is trained to process and reproduce the original datum. In the act of distilling and expanding information, the AE is expected to learn some of the essential features of the data. The AE is sufficient to re-

produce the training data, but it can easily learn to memorize the data. To be able to extrapolate and sample new molecules, we must fill the uncovered spaces of the latent space. The VAE achieves better generalizability by constraining the encoding network to generate latent vectors following a probability distribution on the latent space; often the distribution is Gaussian, owing to its accessible numerical and theoretical properties. Therefore, a molecule is represented not as a fixed point but as a probability distribution over latent space. In practice, this is done as a sampling procedure; when training, noise is added to the latent vector, so the VAE must reconstruct the same molecule from a noisy vector.

Arguably, the most interesting part of a VAE is the latent space. Molecules are represented as continuous and differentiable vectors residing on a probabilistic manifold. Latent space encodes a geometry; for a given molecule, we can sample nearby to decode similar molecules, and with increasing distance, we decode increasingly dissimilar molecules.

Given two molecules, we can trace a path between their corresponding latent coordinates, interpolate among the path, and decode interpolated molecules. Initially, VAEs were proposed for encoding characters of SMILES and then extended to take into account grammar and syntax features, which improve the generation of syntactically valid structures (70, 71).

Latent space allows for direct gradient-based optimization of properties, as latent space is a continuous vector space. Nevertheless, the manifold of molecules has many local minima. One approach has been to explore a smoothed version of the manifold via Bayesian optimization (71) or constrained optimization with Gaussian processes (47).

By jointly training the VAE to reproduce molecules and properties, in a semi-supervised fashion, the latent molecular space reorganizes itself so that molecules with similar properties are close to each other. For a given property, there will exist preferred dimensions and directions. By changing the quality of their Gaussian processes, Gómez-Bombarelli *et al.* (47) demonstrated the capability of local or global optimization across the generated distribution.

Another way of building a generative model is with adversarial training under the GAN framework. Here, the generator competes against a discriminative model; specifically, the generator tries to generate synthetic data from sampling a noise space, whereas the discriminator tries to distinguish data as synthetic or real. Both models train in alternation, with the goal of the generator learning to structure noise toward producing data that the discriminator cannot classify better than chance. Convergence for GANs is not straightforward and can suffer from several issues, including mode collapse and overwhelming of the generator by the discriminator during training. Improving training for GANs is a current research topic (72), and dealing with discrete data, which suffers from nondifferentiability, has some workarounds (73, 74).

To bias the generation process with GANs and RNNs, a gradient is needed to guide the optimization of a network toward desired properties. These properties could be modeled via neural networks and backpropagated to the generator, as is the case with GANs where the optimization metric is that the output looks like real data. To incorporate properties from chemoinformatic tools, simulations, or experimental measures, we need to create a gradient estimator that can backpropagate the generator.

The field of RL provides several approaches to this problem; among the most prominent are Q-learning (49) and policy gradients (75). RL considers the generator as an agent that must learn how to take actions (add characters) within an environment or task (SMILES generation) to maximize some notion of reward (properties). With SMILES, assigning rewards can only be done once the sequence is completed. To overcome this problem, Monte Carlo Tree Search (MCTS) is often used as it constructs a tree of probabilities and weights, simulating several possible completions for sequences, evaluating their reward, and weighting paths through the tree based on their success or failure at the given task. The completion behavior (policy) is learned as a neural network.

Because of these features, several molecular applications have adopted RL and MCTS for generation of drug-like molecules (76–78) and reaction synthesis planning (79, 80).

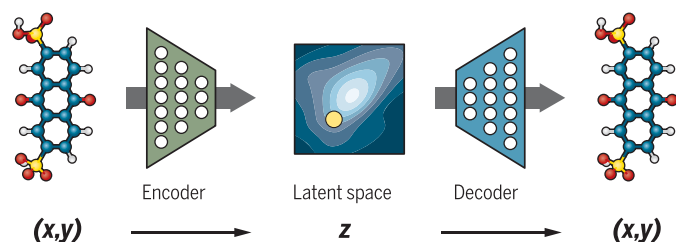
The aforementioned approaches are not exclusive; they can be mixed to yield advantages from each. For instance, druGAN (81) adopts an adversarial autoencoder network, and ORGANIC (82, 83) adopts both adversarial and RL approaches.

It should be noted that most results of generative models have been used in a pharmaceutical context, optimizing properties relevant to potential drugs such as solubility in water, melting temperature, synthesizability, and presence or absence of certain substructures. For example, Popova *et al.* (78) optimized molecules for putative inhibitors of Janus protein kinase 2 (JAK2), and Olivecrona *et al.* (77) optimized molecules active against the target dopamine receptor type 2.

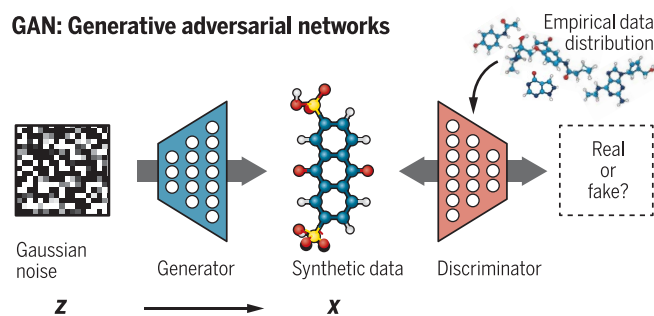
Part of the focus on the SMILES representation has been driven by the adoption of natural language processing deep learning tools. It should be noted that SMILES represents only a subset of possible molecules; for example, a syntactically invalid SMILES string might still be a valid molecular structure, but its physics is not encoded by basic valence rules as used in SMILES. The introduction of more molecular representations and easy-to-use molecular property predictors will expand the use of generative models in other molecular contexts.

Looking ahead, new theoretical developments in ML are opening the door for generative models dealing with graphs. The VAE framework has been extended to molecular graphs (84), and message passing networks are used to incrementally build graphs (57). Even so, there many challenges remain; it is not yet clear how one

## VAE: Variational autoencoders

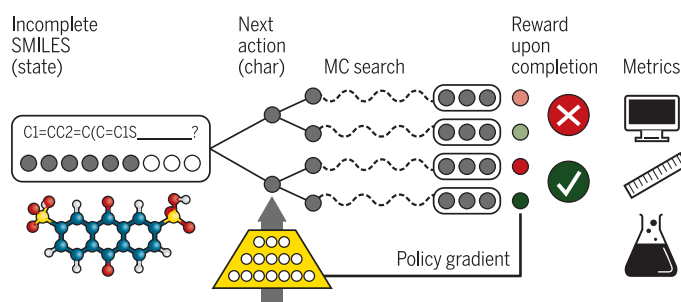


## GAN: Generative adversarial networks

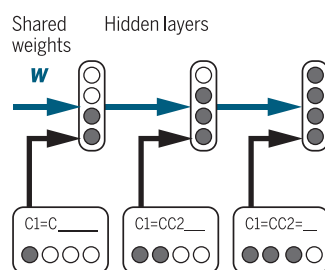


## RL: Reinforcement learning

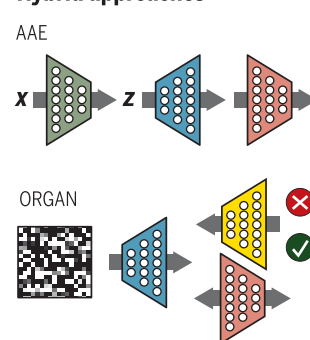
Policy gradient with Monte Carlo tree search (MCTS)



## RNN: Recurrent neural network



## Hybrid approaches



**Fig. 4. Schematic representation of several architectures found in generative models.** RNNs are used for sequence generation. The VAE shows the semi-supervised variant, jointly trained by molecules ( $x$ ) and properties ( $y$ ). Latent space is denoted with  $Z$ , and latent vectors with  $z$ . In the GAN setting, the noise eventually acquires structure via the

adversarial training. Reinforcement learning (RL) shows a policy gradient with MCTS in the task of SMILES completion with arbitrary rewards. Shown in the lower right are hybrid architectures such as AAE (adversarial autoencoders) and ORGAN, which represents GAN and RL.

can deal practically with approximation methods for the graph isomorphism problem.

Additionally, improved sequence generation models are possible with the ability to read and write to memory (69). These approaches demonstrate better ability for learning long- and short-term patterns. More work is needed on Riemannian optimization methods that exploit the geometry of latent space. Structured architectures such as multilevel VAE (85) offer new ways of organizing latent space and are promising research directions. New approaches also lie in inverse RL, geared toward learning a reward or loss function (86). Research in this direction will allow for the discovery of reward functions associated with different materials discovery tasks.

## Outlook

Inverse design is an important component of the complex framework required to design matter at an accelerated pace. The tools for inverse design, especially those stemming from the field of machine learning, have shown rapid progress in the last several years and have allowed chemical space to be framed into probabilistic data-driven models. Generative models produce large numbers of candidate molecules, and the physical realizations of these candidates will require automated high-throughput efforts to validate the generative approach. The community has yet has to show more than a few examples of successful

closed-loop approaches for the design of matter (87). The blurring of the barriers between theory and experiment will lead to AI-enabled automated laboratories (88, 89).

The combination of inverse design tools with active learning approaches such as Bayesian optimization (90, 91) can enable a model that adapts as it explores chemical space, which allows for expanding a model in regions of high uncertainty and enabling the discovery of regions of molecular space with desirable properties as a function of composition. Active learning in the space of objective functions could lead to a better understanding of the best rewards to seek while carrying out machine learning.

As seen, central to machine learning methodologies is the representation of molecules; representations that encode the relevant physics will tend to generalize better. Despite considerable progress, much work remains. Graph and hierarchical representations of molecules are an area requiring further study.

The integration of machine learning as a new pillar of knowledge in the curricula of chemical, biochemical, medicinal, and materials sciences will allow for a more rapid adoption of the methodologies summarized in this work.

## REFERENCES AND NOTES

- Royal Geographical Society, *21st Century Challenges* (2015); <https://21stcenturychallenges.org/challenges/>.

- D. Segal, *Materials for the 21st Century* (Oxford Univ. Press, 2017).
- M. C. Scharber et al., *Adv. Mater.* **18**, 789–794 (2006).
- E. O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, A. Aspuru-Guzik, *Annu. Rev. Mater. Res.* **45**, 195–216 (2015).
- D. J. Newman, G. M. Cragg, *J. Nat. Prod.* **79**, 629–661 (2016).
- P. Kirkpatrick, C. Ellis, *Nature* **432**, 823–823 (2004).
- A. Mullard, *Nature* **549**, 445–447 (2017).
- J.-L. Reymond, *Acc. Chem. Res.* **48**, 722–730 (2015).
- A. M. Virshup, J. Contreras-García, P. Wipf, W. Yang, D. N. Beratan, *J. Am. Chem. Soc.* **135**, 7296–7303 (2013).
- C. Qian, T. Siler, G. A. Ozin, *Small* **11**, 64–69 (2015).
- M. I. Jordan, T. M. Mitchell, *Science* **349**, 255–260 (2015).
- A. Aspuru-Guzik, R. Lindh, M. Reiher, *ACS Cent. Sci.* **4**, 144–152 (2018).
- P. B. Jørgensen, M. N. Schmidt, O. Winther, *Mol. Inform.* **37**, 1700133 (2018).
- E. Maine, E. Garnsey, *Res. Policy* **35**, 375–393 (2006).
- A. Aspuru-Guzik, K. Persson, *Materials Acceleration Platform: Accelerating Advanced Energy Materials Discovery by Integrating High-Throughput Methods and Artificial Intelligence*. Mission Innovation (2018); Innovation Challenge 6.
- T. Weymuth, M. Reiher, *Int. J. Quantum Chem.* **114**, 823–837 (2014).
- J. R. Broach, J. Thorner, *Nature* **384** (suppl.), 14–16 (1996).
- S. Hoelder, P. A. Clarke, P. Workman, *Mol. Oncol.* **6**, 155–176 (2012).
- D. Xiao, L. A. Martini, R. C. Snoeberger 3rd, R. H. Crabtree, V. S. Batista, *J. Am. Chem. Soc.* **133**, 9014–9022 (2011).
- S. A. Lopez, B. Sanchez-Lengeling, J. de Goes Soares, A. Aspuru-Guzik, *Joule* **1**, 857–870 (2017).
- I. Y. Kanai, S. G. Owens, J. S. Bechtel, G. R. Hutchison, *J. Phys. Chem. Lett.* **4**, 1613–1623 (2013).
- J. Hachmann et al., *Energy Environ. Sci.* **7**, 698–704 (2014).

25. L. Cheng *et al.*, *J. Phys. Chem. Lett.* **6**, 283–291 (2015).
26. K. Lin *et al.*, *Nat. Energy* **1**, 16102 (2016).
27. R. Gómez-Bombarelli *et al.*, *Nat. Mater.* **15**, 1120–1127 (2016).
28. D. W. Robbins, J. F. Hartwig, *Science* **333**, 1423–1427 (2011).
29. A. Jain *et al.*, *Comput. Mater. Sci.* **50**, 2295–2310 (2011).
30. I. Petousis *et al.*, *Sci. Data* **4**, 160134 (2017).
31. Q. Yan *et al.*, *Proc. Natl. Acad. Sci. U.S.A.* **114**, 3040–3043 (2017).
32. X. Qu *et al.*, *Comput. Mater. Sci.* **103**, 56–67 (2015).
33. A. Supady, V. Blum, C. Baldauf, *J. Chem. Inf. Model.* **55**, 2338–2348 (2015).
34. I. Y. Kanak, G. R. Hutchison, Rapid computational optimization of molecular properties using genetic algorithms: Searching across millions of compounds for organic photovoltaic materials. arXiv:1707.02949 [physics.app-ph] (2017).
35. N. Yoshikawa, K. Terayama, T. Honma, K. Oono, K. Tsuda, Population-based de novo molecule generation, using grammatical evolution. arXiv:1804.02134 [physics.chem-ph] (2018).
36. C. Rupakheti, A. Virshup, W. Yang, D. N. Beratan, *J. Chem. Inf. Model.* **55**, 529–537 (2015).
37. G. Schneider, U. Fechner, *Nat. Rev. Drug Discov.* **4**, 649–663 (2005).
38. T. Salimans, J. Ho, X. Chen, S. Sidor, I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864 [stat.ML] (2017).
39. H. Ikebata, K. Hongo, T. Isomura, R. Maezono, R. Yoshida, *J. Comput. Aided Mol. Des.* **31**, 379–391 (2017).
40. T. Weymuth, M. Reiher, Gradient-driven molecule construction: An inverse approach applied to the design of small-molecule fixating catalysts. arXiv:1401.1491 [physics.chem-ph] (2014).
41. M. Wang, X. Hu, D. N. Beratan, W. Yang, *J. Am. Chem. Soc.* **128**, 3228–3232 (2006).
42. P. Raccuglia *et al.*, *Nature* **533**, 73–76 (2016).
43. B. Meredig, C. Wolverton, *Chem. Mater.* **26**, 1985–1991 (2014).
44. I. J. Goodfellow *et al.*, Generative adversarial networks. arXiv:1406.2661 [stat.ML] (2014).
45. A. van den Oord *et al.*, WaveNet: A generative model for raw audio. arXiv:1609.03499 [cs.SD] (2016).
46. S. R. Bowman *et al.*, Generating sentences from a continuous space. arXiv:1511.06349 [cs.LG] (2015).
47. R. Gómez-Bombarelli *et al.*, *ACS Cent. Sci.* **4**, 268–276 (2018).
48. D. P. Kingma, M. Welling, Auto-encoding variational Bayes. arXiv:1312.6114v10 [stat.ML] (2013).
49. V. Mnih *et al.*, Playing Atari with deep reinforcement learning. arXiv:1312.5602 [cs.LG] (2013).
50. A. P. Bartók, R. Kondor, G. Csányi, *Phys. Rev. B Condens. Matter* **87**, 184115 (2013).
51. M. Hirn, S. Mallat, N. Poilvert, *Multiscale Model. Simul.* **15**, 827–863 (2017).
52. K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, A. Tkatchenko, *Nat. Commun.* **8**, 13890 (2017).
53. O. A. von Lilienfeld, *Int. J. Quantum Chem.* **113**, 1676–1689 (2013).
54. D. Weininger, A. Weininger, J. L. Weininger, *J. Chem. Inf. Comput. Sci.* **29**, 97–101 (1989).
55. S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, *J. Comput. Aided Mol. Des.* **30**, 595–608 (2016).
56. Z. Wu *et al.*, MoleculeNet: A benchmark for molecular machine learning. arXiv:1703.00564 [cs.LG] (2017).
57. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry. arXiv:1704.01212 [cs.LG] (2017).
58. E. J. Bjerrum, SMILES enumeration as data augmentation for neural network modeling of molecules. arXiv:1703.07076 [cs.LG] (2017).
59. M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, *Phys. Rev. Lett.* **108**, 058301 (2012).
60. J. N. Wei, D. Duvenaud, A. Aspuru-Guzik, *ACS Cent. Sci.* **2**, 725–732 (2016).
61. K. Hansen *et al.*, *J. Phys. Chem. Lett.* **6**, 2326–2331 (2015).
62. B. Huang, O. A. von Lilienfeld, The “DNA” of chemistry: Scalable quantum machine learning with “amons.” arXiv:1707.04146 [physics.chem-ph] (2017).
63. D. Duvenaud *et al.*, Convolutional networks on graphs for learning molecular fingerprints. arXiv:1509.02922 [cs.LG] (2015).
64. D. Rogers, M. Hahn, *J. Chem. Inf. Model.* **50**, 742–754 (2010).
65. J. Behler, *J. Chem. Phys.* **134**, 074106 (2011).
66. D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models. arXiv:1401.4082 [stat.ML] (2014).
67. A. Graves, Generating sequences with recurrent neural networks. arXiv:1308.0850 [cs.NE] (2013).
68. S. Hochreiter, J. Schmidhuber, *Neural Comput.* **9**, 1735–1780 (1997).
69. C. Olah, S. Carter, *Distill* **1**, (2016).
70. M. J. Kusner, B. Paige, J. M. Hernández-Lobato, Grammar variational autoencoder. arXiv:1703.01925 [stat.ML] (2017).
71. H. Dai, Y. Tian, B. Dai, S. Skiena, L. Song, Syntax-directed variational autoencoder for structured data. arXiv:1802.08786 [cs.LG] (2018).
72. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN. arXiv:1701.07875 [stat.ML] (2017).
73. L. Yu, W. Zhang, J. Wang, Y. Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. AAAI (2017); www.aai.org/ocs/index.php/AAAI/AAAI17/paper/download/14344/14489.
74. R. D. Hjelm *et al.*, Boundary-seeking generative adversarial networks. arXiv:1702.08431 [stat.ML] (2017).
75. D. Silver *et al.*, *Nature* **529**, 484–489 (2016).
76. X. Yang *et al.*, *Sci. Technol. Adv. Mater.* **18**, 972–976 (2017).
77. M. Olivecrona, T. Blaschke, O. Engkvist, H. Chen, *J. Cheminform.* **9**, 48 (2017).
78. M. Popova, O. Isayev, A. Tropsha, Deep reinforcement learning for de-novo drug design. arXiv:1711.10907 [cs.AI] (2017).
79. M. H. S. Segler, M. Preuss, M. P. Waller, *Nature* **555**, 604–610 (2018).
80. Z. Zhou, X. Li, R. N. Zare, *ACS Cent. Sci.* **3**, 1337–1344 (2017).
81. A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, A. Zhavoronkov, *Mol. Pharm.* **14**, 3098–3104 (2017).
82. G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, A. Aspuru-Guzik, Objective-Reinforced Generative Adversarial Networks (ORGAN) for sequence generation models. arXiv:705.10843 [stat.ML] (2017).
83. B. Sanchez-Lengeling, C. Outeiral, G. L. Guimaraes, A. Aspuru-Guzik, Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). *ChemRxiv* [Preprint] (2017).
84. W. Jin, R. Barzilay, T. Jaakkola, Junction tree variational autoencoder for molecular graph generation. arXiv:1802.04364 [cs.LG] (2018).
85. D. Bouchacourt, R. Tomioka, S. Nowozin, Multi-level variational autoencoder: learning disentangled representations from grouped observations. arXiv:1705.08841 [cs.LG] (2017).
86. C. Finn, P. Christiano, P. Abbeel, S. Levine, A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. arXiv:1611.03852 [cs.LG] (2016).
87. P. Nikolaev *et al.*, *Npj Comput. Mater.* **2**, 16031 (2016).
88. S. V. Ley, D. E. Fitzpatrick, R. J. Ingham, R. M. Myers, *Angew. Chem. Int. Ed. Engl.* **54**, 3449–3464 (2015).
89. P. J. Kitson *et al.*, *Science* **359**, 314–319 (2018).
90. B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, *Proc. IEEE* **104**, 148–175 (2016).
91. F. Häse, L. M. Roch, C. Kreisbeck, A. Aspuru-Guzik, PHOENICS: A universal deep Bayesian optimizer. arXiv:1801.01469 [stat.ML] (2018).
92. B. Huskinson *et al.*, *Nature* **505**, 195–198 (2014).
93. L. M. Roch *et al.*, *Sci. Robot.* **3**, eaat5559 (2018).

## ACKNOWLEDGMENTS

We thank A. Frøseth for support of this work. A.A.-G. is a cofounder of Kebotix, a startup company that works in automated materials discovery.

10.1126/science.aat2663



## Inverse molecular design using machine learning: Generative models for matter engineering

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik

*Science* **361** (6400), . DOI: 10.1126/science.aat2663

### View the article online

<https://www.science.org/doi/10.1126/science.aat2663>

### Permissions

<https://www.science.org/help/reprints-and-permissions>

Use of this article is subject to the [Terms of service](#)

---

*Science* (ISSN 1095-9203) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science* is a registered trademark of AAAS.

Copyright © 2018 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works