


Machine learning approaches for the prediction of materials properties

Cite as: APL Mater. **8**, 080701 (2020); <https://doi.org/10.1063/5.0018384>

Submitted: 15 June 2020 • Accepted: 16 July 2020 • Published Online: 04 August 2020

 Siwar Chibani and  François-Xavier Coudert

COLLECTIONS

 This paper was selected as Featured



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Machine learning for materials design and discovery](#)

Journal of Applied Physics **129**, 070401 (2021); <https://doi.org/10.1063/5.0043300>

[Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science](#)

APL Materials **4**, 053208 (2016); <https://doi.org/10.1063/1.4946894>

[Perspective: Machine learning potentials for atomistic simulations](#)

The Journal of Chemical Physics **145**, 170901 (2016); <https://doi.org/10.1063/1.4966192>

APL Materials

SPECIAL TOPIC: Emerging Materials
for Spin-Charge Interconversion

Read Now!



Machine learning approaches for the prediction of materials properties

Cite as: APL Mater. 8, 080701 (2020); doi: 10.1063/5.0018384

Submitted: 15 June 2020 • Accepted: 16 July 2020 •

Published Online: 4 August 2020



Siwar Chibani  and François-Xavier Coudert^{a)} 

AFFILIATIONS

Chimie ParisTech, PSL University, CNRS, Institut de Recherche de Chimie Paris, 75005 Paris, France

^{a)} Author to whom correspondence should be addressed: fx.coudert@chimieparistech.psl.eu

ABSTRACT

We give here a brief overview of the use of machine learning (ML) in our field, for chemists and materials scientists with no experience with these techniques. We illustrate the workflow of ML for computational studies of materials, with a specific interest in the prediction of materials properties. We present concisely the fundamental ideas of ML, and for each stage of the workflow, we give examples of the possibilities and questions to be considered in implementing ML-based modeling.

© 2020 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0018384>

I. INTRODUCTION

The pace of systematic materials discovery has quickened in the last decade. The number of studies systematically exploring various families of materials, with the goal of discovering existing materials with unsuspected properties, or designing novel materials with targeted properties, is growing at an astounding rate. Databases of experimental structures—in particular, crystalline structures—continue to grow at a steady pace and are complemented with larger and larger databases of physical and chemical properties. High-throughput experiments and combinatorial materials synthesis are aided by robotics and artificial intelligence, performing reactions and analysis faster. On the computational side of things, molecular simulations have expanded in scale, allowing scientists to predict the structure and properties of complex materials even before they are synthesized. The prediction of compound properties with high accuracy can be coupled with high-throughput screening techniques to help search for new materials. Yet, despite the advances in the computational power, computational methods—whether at the quantum or classical level—are still relatively time consuming and can hardly explore the properties of all possible chemical compositions and crystal structures. In order to reach this goal of systematic exploration of chemical space and to help leverage the large-scale databases of structures and properties that are nowadays available, computational chemistry and materials science are turning more and more often to machine learning (ML), a subset of

artificial intelligence (AI) that has seen tremendous developments in recent years and widespread application across all fields of research.

The main idea of artificial intelligence emerged in the 1950s when Turing wondered if a machine could “think.”¹ The term “artificial intelligence” (AI) was first coined by John McCarthy in 1955 and is defined as the set of theories and techniques implemented in order to create machines capable of simulating intelligence. In other words, AI is the endeavor to replicate the human intelligence in computers. In 1959, Samuel produced computer programs that were playing checkers (drafts) better than the average human and that could learn to improve from past games.² Since then, AI and data-intensive algorithms have seen such an important development that they are sometimes called the “fourth paradigm of science”³ or the “fourth industrial revolution.” AI is now routinely used in different fields: face recognition, image classification, information engineering, linguistics, psychology, and medicine, and it has impact in the fields of philosophy and ethics.

AI-powered machines are usually classified under two broad categories: general and narrow. The artificial general intelligence (AGI) is a machine that can learn to solve any problem that the human intellect can solve. Also referred to as “strong AI” or “full AI,” it is currently hypothetical, the kind of artificial intelligence that we see in science fiction movies. The creation of AGI is an important goal for some AI researchers, but is an extremely difficult quest and generally considered too complex to be achieved in

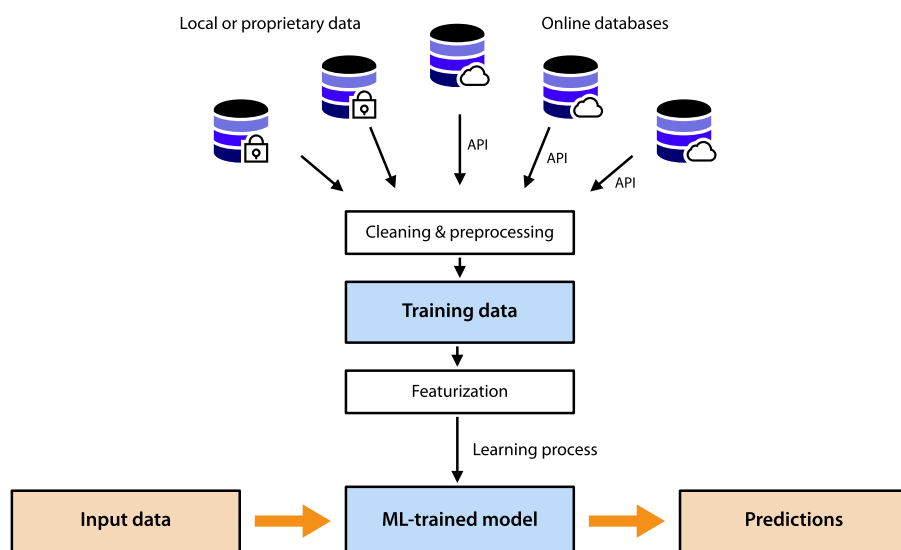


FIG. 1. Simplified overview of a machine learning workflow. The machine learning model is trained on input data gathered from multiple databases. Once it is trained, it can be applied to make predictions for other input data.

the near future.⁴ In contrast, narrow AI (or “weak AI”) is a kind of artificial intelligence focused on performing specific tasks, defined in advance. Narrow AI has seen a very large number of successful realizations of artificial intelligence to date, sometimes in applications where the machine seems intelligent (in the human way) and sometimes hidden under the hood. Much of these successes of narrow AI have been made possible by advances in machine learning (ML), in general, and in deep learning (DL), more specifically in the past few years.

Machine learning aims at developing algorithms that can learn and create statistical models for data analysis and prediction. The ML algorithms should be able to learn by themselves—based on data provided—and make accurate predictions, without having been specifically programmed for a given task. Beyond theoretical developments, recent years have seen rapid advances in the application of machine learning, not only by computer scientists and experts in the development of AI algorithms but also by other researchers in different fields who adopt these techniques for their own purposes. Among many other fields of research, chemical and materials sciences have been impacted by the application of machine learning to accelerate certain computational tasks or to solve problems for which traditional modeling methods were ill-suited. Deep learning, a subset of machine learning based on artificial neural networks (ANNs), promises to escalate the advances of AI even further.

In this paper, we set out to illustrate the workflow of machine learning in the computational materials context (schematized in Fig. 1) and give examples at each stage of the possibilities and questions to be considered in implementing ML-based modeling. In this very active and rapidly expanding field of research, we will try to highlight some—but not all—of the machine learning techniques that have been successfully applied in real applications for computational chemistry of materials, either as they are representative of what is done in the field or because they represent recent and exciting developments. We will also discuss the specific contributions made to our field by deep learning studies, although they are currently more limited. The goal of this paper is to provide a brief

overview to chemists and materials scientists, but we do not try to be exhaustive in our discussion of the state of the art. For a full review on machine learning for molecular and materials science, we refer the reader to the excellent introductory yet thorough review of Butler *et al.*⁵

II. IMPLEMENTING A MACHINE LEARNING METHODOLOGY

A. Gathering data

As stated in the Introduction, machine learning algorithms are trained on existing datasets to learn and improve. In order to create accurate models, the size and quality of the datasets used for training play a crucial role. This identification, gathering, or creating (in some cases) of the training dataset is the first step of the machine learning workflow and will, of course, heavily depend on the goal of the model you want to train. For generic purposes, in order to “learn by doing” the various steps in implementing a ML workflow, one can find free datasets through platforms such as Kaggle (<https://www.kaggle.com>), the UC Irvine Machine Learning Repository (<https://archive.ics.uci.edu/ml>), or on governmental or agency websites that gather and promote open data (<https://www.data.gouv.fr> in France, <https://www.data.gov/> in the USA).

When it comes to materials sciences, there are a number of available datasets that have been published and validated in the scientific literature. Many of them are open and publicly available to any potential user, but others have stricter licensing terms or require a paid subscription for access to some or all features. On practical terms, they mainly differ in the data featured within the dataset, with two main categories: databases that focus (exclusively or predominantly) on structural information and databases that focus on physical or chemical properties of materials. Table I presents a short list of selected databases in materials sciences.

The first category, i.e., databases of structures of materials, is probably the most well-known and historically the most

TABLE I. Some of the largest and most used databases in materials sciences, classified into two categories: databases restricted to crystalline structures only and databases focused on both the structures and materials properties.

Database	Structures	Properties
Structural databases		
Cambridge Structural Database (CSD)	1 031 632	
Inorganic Crystal Structure Database (ICSD)	218 839	
Crystallography Open Database (COD)	457 771	
International Centre for Diffraction Data (ICDD)	1 004 568	
Databases of structures and properties		
AFLOW	3 249 264	Formation energy, band structures, and Bader charges Elastic and thermal properties Binary, ternary, and quaternary systems
Materials project	654 758	Band structures Elastic and piezoelectric tensors Porous volume and surface
Open Quantum Materials Database (OQMD)	637 644	Formation energy and band structures

developed. They include the ubiquitous Cambridge Structural Database (CSD, <https://www.ccdc.cam.ac.uk>), featuring more than one million experimental crystal structures, which is considered a standard repository for the publication of new crystal structures ranging from organic, metal–organic, and organometallic molecules and compounds. Other such databases include the Inorganic Crystal Structure Database (ICSD, <https://icsd.fiz-karlsruhe.de>) for inorganic crystals, the freely accessible Crystallography Open Database⁶ (COD, <http://crystallography.net>), the International Centre for Diffraction Data (ICDD, <http://www.icdd.com/>), and many others. Similar databases exist for other chemical systems, such as GDB⁷ (<http://gdb.unibe.ch/downloads/>) for small organic molecules and ZINC⁸ (<https://zinc15.docking.org/>) for commercially available compounds for virtual screening.

As should be apparent when reading the above list, it is important to be aware of an important bias on how these databases cover the field of materials science: they are all limited to crystallographic structures. While this is obviously linked to the nature of the determination of the structure and its representation, it is important to be aware of such a bias. This is only one example—and there are many others—of how databases exhibit, by the very own choice of their scope and the representations chosen, a biased representation of the wide scope of the field of materials sciences. It is, therefore, necessary to be aware of the biases in the datasets one is using, both implicit and explicit.

Beyond structural databases, the past few years have seen the development of another category, with a rapid growth in the number of structure–property databases available—often, again, with a specific focus on a particular class of materials or specific properties. The existence of such databases with a large amount of data, most of which are open access and collaborative, presents a significant opportunity to train and to validate new machine learning models. We list some here, whose characteristics are summarized in Table I; the choice made is not to try and be exhaustive (which

would necessarily fail, given the large and ever-growing number of existing databases) but to highlight those that appear commonly used and have easy access and are well-documented for newcomers to the materials discovery field. Among the largest databases, we can cite the Materials Project⁹ (<https://materialsproject.org/>) for inorganic materials, the AFLOWLIB¹⁰ “Automatic Flow for Materials Discovery” (<http://aflowlib.org/>), and the Open Quantum Materials Database¹¹ (<http://oqmd.org>). In addition to these generic sources, there are also specific databases of computed properties for specific classes of materials, such as the Harvard Clean Energy Project¹² (previously at <https://cepdb.molecularspace.org>, currently being migrated) for organic solar materials, TE Design Lab¹³ (<http://tedesignlab.org>) for thermoelectric materials, and NREL Materials Database¹⁴ (<https://materials.nrel.gov>) for renewable energy materials. Finally, other online portals allow the sharing and exchange of computational data on materials from different origins, resulting in a more heterogeneous dataset, such as the Materials Cloud¹⁵ (<https://www.materialscloud.org/>).

Most of these databases are accessible through both a web front-end, for simple exploration and visualization purposes, and an Application Programming Interface (API). An API is a web interface with well-documented behavior, whose queries and results are machine-readable in an agreed-upon format, making them well-suited for automated exploitation. These API are typically accompanied with a software layer to facilitate integration into projects, such as the Python Materials Genomics (pymatgen)¹⁶ Python package that integrates with the Materials Project RESTful API, or the Automated Interactive Infrastructure and Database (AiiDA).¹⁷

Finally, we would be remiss if we did not note that it is also possible to generate data to form a machine learning training set “on the fly,” by performing high-throughput calculations on materials of interest. When this is done, it is then expected that the dataset produced is published alongside the work as supplementary material or submitted to an online data repository.

B. Cleaning and preprocessing data

We have described above how to use existing datasets of materials structures and properties (or generate one's own), yet these data cannot be used directly in the original format and loaded “as is” in a machine learning workflow. When it comes to large datasets, four points are considered crucial in big data workflows, called the “four V’s,” and they are also relevant in machine learning methodologies: (i) *volume*, the amount of data available; (ii) *variety*, the heterogeneity of the data in both form and meaning; (iii) *veracity*, the knowledge of the uncertainties associated with each data point; and (iv) *velocity*, how fast the data are generated and have to be treated—not usually an issue in our workflows, which do not have to work in real time.

Therefore, data have to be homogenized and cleaned before it can be used. This means identifying possible erroneous, missing, or inconsistent data points (outliers), using criteria based on physical or chemical knowledge. This cleaning and homogenization of the data is a key step in order to build more accurate predictors through machine learning. The need for this may depend on the workflow followed: for example, some ML algorithms are more robust than others in the presence of outliers. Some algorithms (such as the Random Forest family) do not support null values in their input, while others can handle those.

To give one example of the necessity of this curation of the training dataset, we recently performed a large-scale exploration of the elastic properties of inorganic crystalline materials available on the Materials Project database. By analyzing the elastic data present, we quantified that out of 13 621 crystals, only 11 764 structures were mechanically stable, while 1857 (around 14% of materials in the database) had elastic tensors that indicated mechanical instability (and were, therefore, unusable for further analysis).¹⁸ Other materials had elastic moduli that were mathematically acceptable but unphysically large and those needed to be removed as well before using the dataset.

C. Representing data

Once data have been cleaned up and homogenized, the next step in the machine learning workflow is the encoding of these data into a set of specific variables, which will be manipulated directly by the ML algorithm. The data collected are often in a raw format and will need to be converted into a format suitable for learning procedure, usually as a series of scalar or vector variables for each entry of the dataset. This step can include the transformation of existing data (such as physical properties) by rescaling, normalization, or binarization to bring it to such a state that the algorithm can easily parse it. Some basic preprocessing techniques are widely available in ML software, such as the `MinMaxScaler` or `StandardScaler` methods in `scikit-learn`.¹⁹ In all cases, the effect of this preprocessing of the data needs to be studied carefully: it is important to mention that sometimes algorithms can deliver better results without preprocessing, and with excessive preprocessing, it may not be possible to identify the crucial features that will give the best performance for the target variable.

When the input data consist of chemical structures, the choice of representation of the data is not always obvious: chemical compounds and materials are complex three-dimensional objects, whose direct representation as vectors of coordinates may not be efficient

as input for the ML workflow. This question of the best representation of the data for the learning algorithm is called *featurization* or *feature engineering*. It is a very active area of research, in particular, when it comes to describing chemical structures. The two main goals of feature engineering are (i) preparing the input data in a form that the ML algorithms will work well with (and that can depend on the specific characteristics of the algorithm chosen); (ii) improving the performance of ML models, by using our knowledge of the materials and their important features (chemical intuition) in the building of the input data.

The chemical information about a given system can then be transformed into a series of the so-called descriptors, which encode the chemical information, creating a new input that should describe the key features of the dataset in a way that allows for the ML algorithm to train efficiently. Many different mathematical representations are used as descriptors for chemical and materials structures (and their properties). We will cite here examples for molecular structures such as Coulomb matrix,²⁰ SMILES,²¹ bag of bonds,²² molecular graphs,²³ and BAML (bonds, angles, machine learning; using bonds, angles, and higher order terms).²⁴ Among the representations used for crystals, we find representations such as translation vectors, fractional coordinates of the atoms, radial distribution functions,²⁵ Voronoi tessellations of the atomic positions,²⁶ and property-labeled materials fragments.²⁷

D. Machine learning models

1. Supervised learning

Let us now move to fourth step of the machine learning workflow: the “learning” part itself, i.e., the training of the ML algorithm. Using the curated and pre-processed dataset as input, there are, then, three main categories of ML models: supervised, unsupervised, and semi-supervised (see Fig. 2). In *supervised learning*, the

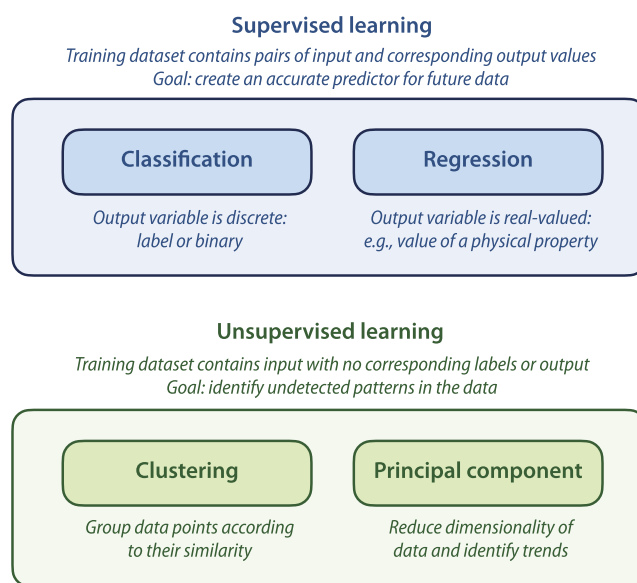


FIG. 2. Different categories of machine learning models.

dataset is considered as a training set and must contain both input variables and their corresponding output variable: think of chemical structures and their properties for a common example in chemistry. Then, the goal of the ML algorithm is to learn, from these training data, the mapping function from the input (the structures) to the output (the properties). The main goal of the machine learned algorithm in a supervised learning approach is to be able to make a prediction for new data, with an acceptable level of accuracy, once it has been trained.

It is useful to know that supervised learning problems can be broadly categorized into two main types: regression and classification techniques. Both of them have as a goal the construction of a model that can predict values from the available variables. The only difference between the two is the type of the output variable: in a regression-type prediction problem, the variable that is to be predicted is continuous, taking real values: melting point, bandgap, elastic modulus, etc. Regression learning algorithms include linear regression, lasso linear regression, ridge regression, elastic net regression, Gaussian process regression, and classification and regression trees. The simplest of these algorithms is the linear regression (LR), which—similar to the common linear regression in 2D graphs—tries to create the best linear model based on the descriptors given. If the model is complex, a lasso linear regression algorithm can be used instead, which is a modification of LR where the loss function is modified to minimize the complexity of the model.

On the other hand, for classification problems, the algorithm has to predict a categorical variable, i.e., attribute a label to the input data. In the simplest case, these categories are reduced to two in a binary variable, such as: is the material conducting or isolating? and is it porous or not? (or in another context: is this email a spam or not?). A large number of different classification algorithms can be used: logistic regression, linear discriminant analysis, *k*-nearest neighbor, naïve Bayes, classification and regression trees, support vector machine, and kernel ridge regression (KRR). To give an example, Ghiringhelli *et al.* used KRR with descriptors derived from the energy levels and radii of valence orbitals to predict crystalline arrangements between zinc blende and wurtzite-type crystal structures.²⁸

2. Unsupervised learning

Unsupervised learning works in a completely different way, trying to draw inferences about the input data without any corresponding output variables: it is used to look for previously undetected patterns in data with minimal human supervision or guidance. It can, for example, act on unlabeled data to discover the inherent groupings. The ML algorithm tries to identify trends that could be of interest to rationalize the dataset and present the available data in a novel way. One family of the unsupervised learning algorithm is the clustering or cluster analysis: there, the ML workflow will split the data into several groups (or clusters) of records presenting similar features, with no prior assumption on the nature of these groups (unlike, for example, in supervised learning classification tasks). Classical methods used for clustering include Gaussian mixtures, *k*-means clustering, hierarchical clustering, and spectral clustering. Other types of unsupervised learning method are association rule learning and principal component algorithms that aim to establish relationships between multiple features in a large dataset,

something that is difficult to do by hand. Popular algorithms for generating association rules have been proposed, such as *A priori*, Eclat, and FP-Growth (Frequent-Pattern).

When it comes to applications in chemistry and materials science, supervised machine learning is a lot more common, but there are some examples of unsupervised learning, nonetheless; for a recent perspective on this specific topic, see Ref. 29. Saad *et al.* applied both the supervised and unsupervised ML techniques to predict the structure and properties of crystals (such as the melting point) for binary compounds.³⁰ Supervised ML models have been trained to reproduce the LUMO and HOMO for organic solar cells³¹ and to predict key thermodynamic parameters such as adsorption energy,³² activation energy,³³ and active site³⁴ in catalytic processes. Isayev *et al.* developed predictive Quantitative Materials Structure–Property Relationship (QMSPR) models through machine learning, in order to predict the critical temperatures of known superconductors,³⁵ and Woo and co-workers established the QMSPR model to achieve high-throughput screening of metal–organic frameworks (MOFs) to capture CO₂ by adsorption, relying on machine learning to explore the large dimensionality spanned by their exceptional structural tunability.

One disadvantage of supervised ML algorithms lies in the acquisition of labeled data, an expensive process requiring especially when dealing with large amounts of data. Unlabeled data, on the other hand, are relatively inexpensive and easy to collect and store—but applications of unsupervised ML in materials sciences have been relatively limited. An alternative exists in the form of a third training ML model named *semi-supervised*, which is halfway between supervised and unsupervised learning, as its name implies. In such a workflow, we have a large amount of input data and only a limited amount of corresponding output data. Semi-supervised ML aims at learning from the labeled part of the dataset, training an accurate model. First, the workflow will use the unsupervised learning algorithm to identify and cluster similar data. Then, it will use supervised learning techniques to train and make prediction for the rest of the unlabeled data. Semi-supervised learning algorithms make three assumptions about the data, which somewhat restrict their applicability: (i) continuity assumption: the close points are more likely to share a label; (ii) cluster assumption: data can form discrete clusters and points in the same cluster are more likely to share an output label; and (iii) manifold assumption: the data lie approximately on a manifold of much lower dimension than the input space. Semi-supervised algorithms are widely used in text and speech analysis, internet content classification, and protein sequence classification applications.

Semi-supervised learning was used by Court *et al.* to create a materials database of Curie and Néel temperatures for 39 822 by text mining a corpus of 68 078 chemistry and physics articles, applying natural language processing and a semi-supervised relationship extraction algorithm to obtain the values (and units) of the properties from the texts.³⁶ Similarly, Huo *et al.* demonstrated the efficiency and accuracy of semi-supervised machine learning to classify inorganic materials synthesis procedures from written natural language.³⁷ Recently, Kunselman *et al.* used semi-supervised learning methods to analyze and classify microstructure images, training their model on a dataset where only a fraction of the microstructures was labeled initially.³⁸

3. Applications

Most of the applications of ML in chemical and materials sciences, as we have said, feature supervised learning algorithms. The goal there is to supplement or replace traditional modeling methods, at the quantum chemical or classical level, in order to predict the properties of molecules or materials directly from their structure or their chemical composition. To give some recent examples in an area our group has worked in, ML has been used in providing a better understanding and prediction of the mechanical properties of crystalline materials. In 2016, de Jong *et al.* using supervised ML with gradient boosting regression have developed a model to predict the elastic properties (such as the bulk modulus K and shear modulus G) for a large set of inorganic compounds.³⁹ These authors used 187 descriptors for the materials, including a diverse set of composition and structural descriptors, and training data from quantum chemistry calculations at the Density Functional Theory (DFT) level. The trained predictor was then used to provide predictions of K and G for a large range of inorganic materials outside of the training dataset, and these predicted values are now available (as estimates) for materials in the Material Project,⁹ both through the API and on the website.

Our research group was applying the same idea on a narrower range of materials, trying to confirm that for a given chemical composition, geometrical descriptors of a material's structure could lead to accurate predictions of its mechanical features: we used local, structural, and porosity-related descriptors. Evans and Coudert⁴⁰ trained a gradient boosting regressor algorithm on data for 121 pure silica zeolites⁴¹ (SiO_2 polymorphs) and also used it to predict K and G elastic moduli of 590 448 hypothetical frameworks. The results highlighted several important correlations and trends in terms of stability for zeolitic structures. Later, in Gaillac *et al.*, we expanded this ML study to look into anisotropic mechanical properties, which are typically more difficult to model. We obtained an algorithm for the prediction of auxeticity and Poisson's ratio of more than 1000 zeolites.⁴²

Beyond the applications described above for prediction of molecules or materials properties, machine learning has been used at another level, in order to improve existing computational methods. One of the areas where it has been done is in order to improve the exchange-correlation functional in density functional theory (DFT) calculations, for example, in order to provide a better description of weak chemical interactions and highly correlated systems. In this area, much effort has been spent trying to leverage machine learning to produce a universal density functional,^{43,44} to solve the Kohn-Sham equations,⁴⁵ to optimize DFT exchange-correlation functionals,^{46,47} and to create adaptive basis sets.⁴⁸ In the realm of classical molecular simulation, machine learning can be used to optimize interatomic potentials (a.k.a. force fields) with the predetermined analytical form^{49,50} or to create *de novo* force fields,^{51–53} for both molecules and materials.

We should note here that machine learning can also be applied to discover and design entirely new compounds, creating novel opportunities for computationally assisted discovery of materials. Designing materials with targeted physical and chemical properties is recognized as an outstanding challenge in materials research. Using kernel regression, Calfa and Kitchin predicted the electronic properties of 746 binary metal oxides and elastic properties of 1173

crystals.⁵⁴ Then, they used the special features to design a new crystal with an exhaustive enumeration (EE) algorithm that evaluated all the possible combinations of crystals from the dataset. Based on the electronic properties of binary metal oxides, the authors obtained 1 153 504 combinations that should be iterated. Faber and co-workers identified 128 novel structures through the development of a ML model that trained to reproduce the formation energies of two million combinations of elements presenting the ABC_2D_6 formula. The 128 new structures are added later to the Materials Project database.⁵⁵ Other applications in the material exploration include the design of novel catalysts⁵⁶ and novel cathode materials to improve the performance of lithium ion batteries.⁵⁷

E. Learners

In this section, we discuss in a bit more detail an important (but more technical) part of the machine learning workflow: the choice of learning algorithms or *learners*. We have already mentioned in passing above the names of a few of these, which can be applied depending on the type of the data and the underlying problem to be solved. This choice is a crucial step in any ML workflow as the selection of algorithm plays a key role in the accuracy of the prediction.⁵⁸ Many algorithms are readily accessible for non-expert users, with packages written in Python (scikit-learn, Keras, PyTorch), C++ (mlpack, Tensorflow), R (caret), and others. We highlight here some of the possible choices of learners, in an overview which is not remotely exhaustive, but wants to give the reader a glance of the diversity of the methods available.

The family of k -nearest neighbor (k -NN) algorithms can be used for classification and regression tasks in supervised ML. The k -NN algorithm assumes that similar objects in the data are near each other. For a given observation X that we want to predict, the k -NN algorithm will look for the k points closest in the dataset; then, it will use the output variable associated with these nearest neighbors to predict the value of X . **The upsides of k -NN are that the model is simple and easy to implement and that it is non-parametric, with no need for tuning several hyperparameters. On the downside, it gets significantly slower as the volume of the data increases.**

Decision trees (DTs) are another family of learners that can handle both classification and regression supervised ML (forming classification trees and regression trees, respectively). The use of decision trees in machine learning represents an option that is simple to understand and interpret, as the trees can be explicitly visualized. In the tree structure, the root of the tree is the input data, and each branch represents a possible decision. The input data are broken down into smaller and smaller subsets in the tree, with splitting rules implemented in each internal node of the tree based on the data. The leaves of the tree represent the output of the algorithm. The choice of the DT model is important to avoid overfitting (with unnecessarily complex trees)—this can be achieved by mechanisms such as setting the maximum depth of the tree and the minimum number of samples required at a leaf node. Different types of decision tree-based learners exist, such as Random Forest (RF) and Gradient Boosting Decision Tree (GBDT). RF uses averaging to reduce the overfitting and improve the predictive accuracy, while GBDT tries to correct the error of previous trees by merging several trees on smaller depths.

Another family is that of the naïve Bayes classifiers, supervised ML algorithms for classification. They are based, as their name indicates, on Bayes' theorem with an assumption of independence between the features. They have been studied and developed for a long time, are easy to employ, and are extremely scalable. Different kinds exist, depending on the nature of the data. Their main downside is the assumption of independence of the predictors, which rarely holds in complex use cases.

As the last family we will cite here, kernel methods are a class of algorithms for pattern analysis. At their core, they rely on the use of kernel functions, which are applied to the input data to map the original nonlinear observations into a higher-dimensional space in which they become separable. That feature space is implicit because the coordinates of the data in this space are never directly computed, but only the inner product between pairs of data, which is computationally less expensive. Perhaps the best known and most widely used algorithm in this family is the support vector machine kernel ridge regression.

F. Model evaluation

As we explained above, the main goal of machine learning is to train and generate an efficient computational model, whose predictions will be accurate. This accuracy should be confirmed, in order to check that the model captures correctly the underlying patterns in the data, but cannot be validated solely on the results obtained from the training dataset. The best way to check the accuracy is to assess the performance of the trained ML model on data that was not included in its training dataset. However, starting from a dataset of a given size, there are statistical techniques that are better than simply splitting the data into two sets (training and validation), called cross-validation techniques. One of the most used cross-validation methods is the *k*-fold cross-validation. In this method, the dataset is divided into a number of subsets. Then, during the ML training, the model is trained using all the subsets as the training sets but leaves one subset for later testing. The process of training and evaluation is repeated several times, and each time a different subset of data is used for validation.

Among the problems that have to be checked in the evaluation of the ML model is the agreement between the level of complexity

of the model and that of the data. In the case where the data are not sufficiently detailed or the model is too simple, the resulting model can have a bias, a situation named underfitting. On the contrary, if the model is too complex, with a large number of parameters, overfitting can occur. To produce an optimal model, a balance to avoid both underfitting and overfitting by adjustment of the hyperparameters is crucial. This necessary step of tuning the machine learning hyperparameters to select the optimal values is not always easy as it requires systematic searches and patience.

III. DEEP LEARNING

As stated before, machine learning techniques are a form of weak AI and, therefore, not fully autonomous and require some guidance, e.g., in the adjustment of hyperparameters. To go beyond the traditional ML approaches, deep learning (DL) methods were developed that try to mimic more closely some aspects of human cognition. This allows them to outperform other ML algorithms in accuracy and speed, without need for manual intervention from the programmer. DL is a subtype of ML that runs its inputs through a biologically inspired artificial neural network (NN) architecture. Over time, it has been established that NN outperform many other algorithms in accuracy and speed by their strong ability to capture the relevant information from a large amount of data. Deep learning is, in particular, capable of modeling and processing very complex nonlinear relationships.

There are many variants of deep learning methods available, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), artificial neural networks (ANNs), and deep neural networks (DNNs). The neural networks they rely on contain artificial neurons arranged in multiple layers such that each layer communicates only with the layers immediately preceding and following (see Fig. 3). Information travels in the NN from the first layer, or input layer (which receives the external data), to the last layer, or output layer (which produces the final result), through several hidden layers in between. The number of hidden layers depends on the complexity of the problem to be solved.

In each of the hidden layers, the neurons receive the input signal from others neurons, process it by combining the input with

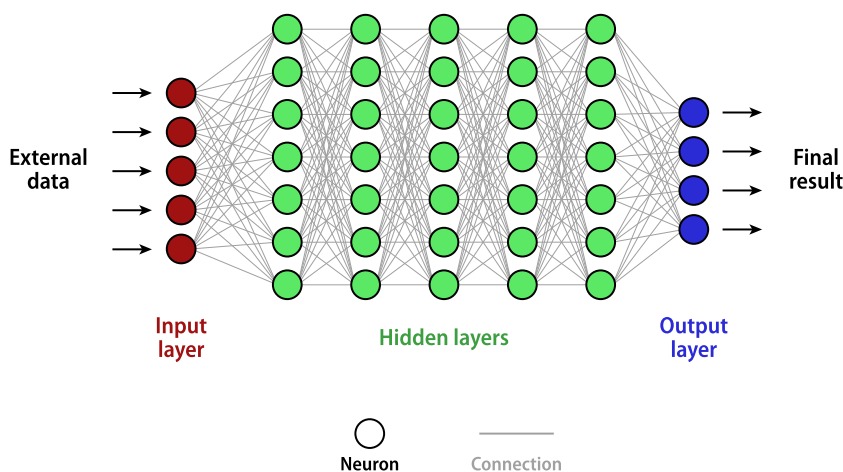


FIG. 3. Schematic representation of a deep neural network, a type of artificial neural network featuring several hidden layers of neurons between the input and output layers.

their internal state, and produce an output signal. The neural network links these neurons through connections, providing the output of one neuron as an input to another neuron. Each neuron can have multiple input and output connections, to which are assigned weights, forming the overall layer of the NN. The learning process involves the adaptation of the network, i.e., the weights of the connections, by minimizing the observed errors in the output of the neural network. Because of this very generic and self-adapting architecture of the NN, deep learning reduces need for feature engineering and can identify and work around defects that would be difficult to spot in other techniques. However, the training of artificial neural networks requires a very large amount of data to train accurately and is computationally expensive due to the large number (millions or more) of parameters to optimize during training.

Several research groups have proposed applications of deep learning to problems in chemical and materials sciences.⁵⁹ For example, Willighagen *et al.* used supervised self-organizing maps (a kind of ANN) to explore large numbers of experimental and simulated crystal structures, in order to visualize structure–property relationships.⁶⁰ Using an ANN implemented in the open-source PyBrain code,⁶¹ Ma *et al.* trained a model with a set of *ab initio* adsorption energies and electronic fingerprints of idealized bimetallic surface (spatial extent of metal *d*-orbitals, atomic radius, ionization potential, electron affinity, and Pauling electronegativity).⁶² This model was able to capture complex nonlinear adsorbate–substrate interactions as it is applied to the electrochemical reduction of carbon dioxide on metal electrodes.

Rule-based expert systems (rules are applied to the reactants to obtain the product in reaction prediction or to the product for retrosynthesis) cannot predict outside of their knowledge and often fail because they ignore the molecular context, which leads to reactivity conflicts. To overcome this problem, deep learning techniques have been used to predict chemical synthesis routes by combining NN with rule-based expert systems. Using this combination, Segler and Waller have ranked the candidate synthetic pathways by the computation of mean reciprocal rank (MRR).⁶³ This model was trained on 3.5×10^6 reactions with a success rate of 95% in retrosynthesis and 97% for reaction prediction. In another work, Cole *et al.* have determined the probability of the predicted product using more than 800 000 organic and organometallic crystal structures in the CSD.⁶⁴ Deep learning is also used in the study and discovery of drug-like molecules, e.g., Gómez-Bombarelli *et al.* have estimated the chemical properties from the latent continuous vector representation of the molecule using the RNN method.⁶⁵ The model they developed allows us to generate new molecules for efficient exploration and optimization.

IV. ARTIFICIAL INTELLIGENCE IN THE LAB

While this paper is focused on providing an introductory description of machine learning approaches for the prediction of chemical systems, in general, and materials properties, more specifically, we want to end it by noting that the use of artificial intelligence techniques in chemistry and materials science is much broader than machine learning and its computational applications—and it provides a lot of exciting avenues for research in the near future.⁶⁶ We refer the reader to Ref. 67 for an in-depth and very insightful review

of the multiple avenues of research opened by artificial intelligence in the field of synthetic organic chemistry.

One particular area of recent achievements for artificial intelligence in chemistry is its integration into chemistry labs achieved through robotics.⁶⁸ Robotic synthesis based on flow chemistry takes high-throughput discovery to an entirely new scale—where chemical syntheses can be described through standardized method descriptions, i.e., “source code for chemistry,” which is then compiled for the specific hardware of a synthesis robot.⁶⁹ Furthermore, this allows high-throughput synthesis and characterization to be tightly coupled with computational screening procedures.⁷⁰ In this approach, it is, thus, possible to leverage an artificial intelligence algorithm to propose synthetic routes, coupled with a robotic microfluidic platform to realize the synthesis and characterize its results.⁷¹ To list two recent examples, Coley *et al.* proposed a robotic platform for flow synthesis of organic compounds, paired with computational prediction techniques based on artificial intelligence,⁷² and Granda *et al.* demonstrated an integrated system where machine learning is used for decision making in real time, during a trial-and-error search for new reactivity where the analysis results from experiments are fed back into the ML algorithm.⁷³

ACKNOWLEDGMENTS

A large part of the work discussed in this article requires access of scientists to large supercomputer centers. Although no original calculations were performed in the writing of this paper, we acknowledge GENCI for high-performance computing CPU time allocations (Grant No. A0070807069). This work was funded by the Agence Nationale de la Recherche under the project “MATAREB” (Grant No. ANR-18-CE29-0009-01).

DATA AVAILABILITY

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

REFERENCES

- 1 A. M. Turing, “Computing machinery and intelligence,” *Mind* **LIX**, 433–460 (1950).
- 2 A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM J. Res. Dev.* **3**, 210–229 (1959).
- 3 *The Fourth Paradigm: Data-Intensive Scientific Discovery*, edited by T. Hey, S. Tansley, and K. Tolle (Microsoft Research, Redmond, WA, 2009).
- 4 *Artificial General Intelligence*, edited by B. Goertzel and C. Pennachin (Springer Berlin Heidelberg, 2007).
- 5 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, “Machine learning for molecular and materials science,” *Nature* **559**, 547–555 (2018).
- 6 S. Gražulis, D. Chateigner, R. T. Downs, A. F. T. Yokochi, M. Quirós, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck, and A. Le Bail, “Crystallography open database—An open-access collection of crystal structures,” *J. Appl. Crystallogr.* **42**, 726–729 (2009).
- 7 T. Fink and J.-L. Reymond, “Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: Assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery,” *J. Chem. Inf. Model.* **47**, 342–353 (2007).
- 8 T. Sterling and J. J. Irwin, “ZINC 15—Ligand discovery for everyone,” *J. Chem. Inf. Model.* **55**, 2324–2337 (2015).

- ⁹A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," *APL Mater.* **1**, 011002 (2013).
- ¹⁰C. E. Calderon, J. J. Plata, C. Toher, C. Oses, O. Levy, M. Fornari, A. Natan, M. J. Mehl, G. Hart, M. B. Nardelli, and S. Curtarolo, "The AFLOW standard for high-throughput materials science calculations," *Comput. Mater. Sci.* **108**, 233–238 (2015).
- ¹¹S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, "The open quantum materials database (OQMD): Assessing the accuracy of DFT formation energies," *npj Comput. Mater.* **1**, 864 (2015).
- ¹²J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. S. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. M. Brockway, and A. Aspuru-Guzik, "The Harvard clean energy project: Large-scale computational screening and design of organic photovoltaics on the world community grid," *J. Phys. Chem. Lett.* **2**, 2241–2251 (2011).
- ¹³P. Gorai, D. Gao, B. Ortiz, S. Miller, S. A. Barnett, T. Mason, Q. Lv, V. Stevanović, and E. S. Toberer, "TE design lab: A virtual laboratory for thermoelectric material design," *Comput. Mater. Sci.* **112**, 368–376 (2016).
- ¹⁴V. Stevanović, S. Lany, X. Zhang, and A. Zunger, "Correcting density functional theory for accurate predictions of compound enthalpies of formation: Fitted elemental-phase reference energies," *Phys. Rev. B* **85**, 115104 (2012).
- ¹⁵L. Talirz, S. Kumbhar, E. Passaro, A. V. Yakutovich, V. Granata, F. Gargiulo, M. Borelli, M. Uhrin, S. P. Huber, S. Zoupanos, C. S. Adorf, C. W. Andersen, O. Schütt, C. A. Pignedoli, D. Passerone, J. VandeVondele, T. C. Schulthess, B. Smit, G. Pizzi, and N. Marzari, "Materials cloud, a platform for open computational science," *arXiv:2003.12510* [cond-mat.mtrl-sci] (2020).
- ¹⁶S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python materials genomics (pymatgen): A robust, open-source python library for materials analysis," *Comput. Mater. Sci.* **68**, 314–319 (2013).
- ¹⁷G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, "AiiDA: Automated interactive infrastructure and database for computational science," *Comput. Mater. Sci.* **111**, 218–230 (2016).
- ¹⁸S. Chibani and F.-X. Coudert, "Systematic exploration of the mechanical properties of 13 621 inorganic compounds," *Chem. Sci.* **10**, 8589–8599 (2019).
- ¹⁹F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
- ²⁰M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Phys. Rev. Lett.* **108**, 058301 (2012).
- ²¹D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Model.* **28**, 31–36 (1988).
- ²²K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space," *J. Phys. Chem. Lett.* **6**, 2326–2331 (2015).
- ²³F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld, "Prediction errors of molecular machine learning models lower than hybrid DFT error," *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).
- ²⁴B. Huang and O. A. von Lilienfeld, "Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity," *J. Chem. Phys.* **145**, 161102 (2016).
- ²⁵K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, "How to represent crystal structures for machine learning: Towards fast prediction of electronic properties," *Phys. Rev. B* **89**, 205118 (2014).
- ²⁶L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton, "Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations," *Phys. Rev. B* **96**, 024104 (2017).
- ²⁷O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, and A. Tropsha, "Universal fragment descriptors for predicting properties of inorganic crystals," *Nat. Commun.* **8**, 15679 (2017).
- ²⁸L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, "Big data of materials science: Critical role of the descriptor," *Phys. Rev. Lett.* **114**, 105503 (2015).
- ²⁹M. Ceriotti, "Unsupervised machine learning in atomistic simulations, between predictions and understanding," *J. Chem. Phys.* **150**, 150901 (2019).
- ³⁰Y. Saad, D. Gao, T. Ngo, S. Bobbitt, J. R. Chelikowsky, and W. Andreoni, "Data mining for materials: Computational experiments with AB compounds," *Phys. Rev. B* **85**, 104104 (2012).
- ³¹F. Pereira, K. Xiao, D. A. R. S. Latino, C. Wu, Q. Zhang, and J. Aires-de-Sousa, "Machine learning methods to predict density functional theory B3LYP energies of HOMO and LUMO orbitals," *J. Chem. Inf. Model.* **57**, 11–21 (2016).
- ³²T. Toyao, K. Suzuki, S. Kikuchi, S. Takakusagi, K. Shimizu, and I. Takigawa, "Toward effective utilization of methane: Machine learning prediction of adsorption energies on metal alloys," *J. Phys. Chem. C* **122**, 8315–8326 (2018).
- ³³K. Takahashi and I. Miyazato, "Rapid estimation of activation energy in heterogeneous catalytic reactions via machine learning," *J. Comput. Chem.* **39**, 2405–2408 (2018).
- ³⁴R. Jinnouchi and R. Asahi, "Predicting catalytic activity of nanoparticles by a DFT-aided machine-learning algorithm," *J. Phys. Chem. Lett.* **8**, 4279–4283 (2017).
- ³⁵O. Isayev, D. Fourches, E. N. Muratov, C. Oses, K. Rasch, A. Tropsha, and S. Curtarolo, "Materials cartography: Representing and mining materials space using structural and electronic fingerprints," *Chem. Mater.* **27**, 735–743 (2015).
- ³⁶C. J. Court and J. M. Cole, "Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction," *Sci. Data* **5**, 17 (2018).
- ³⁷H. Huo, Z. Rong, O. Kononova, W. Sun, T. Botari, T. He, V. Tshitoyan, and G. Ceder, "Semi-supervised machine-learning classification of materials synthesis procedures," *npj Comput. Mater.* **5**, 62 (2019).
- ³⁸C. Kunselman, V. Attari, L. McClenny, U. Braga-Neto, and R. Arroyave, "Semi-supervised learning approaches to class assignment in ambiguous microstructures," *Acta Mater.* **188**, 49–62 (2020).
- ³⁹M. de Jong, W. Chen, R. Notestine, K. Persson, G. Ceder, A. Jain, M. Asta, and A. Gamst, "A statistical learning framework for materials science: Application to elastic moduli of *k*-nary inorganic polycrystalline compounds," *Sci. Rep.* **6**, 15004 (2016).
- ⁴⁰J. D. Evans and F.-X. Coudert, "Predicting the mechanical properties of zeolite frameworks by machine learning," *Chem. Mater.* **29**, 7833–7839 (2017).
- ⁴¹F.-X. Coudert, "Systematic investigation of the mechanical properties of pure silica zeolites: Stiffness, anisotropy, and negative linear compressibility," *Phys. Chem. Chem. Phys.* **15**, 16012 (2013).
- ⁴²R. Gaillac, S. Chibani, and F.-X. Coudert, "Speeding up discovery of auxetic zeolite frameworks by machine learning," *Chem. Mater.* **32**, 2653–2663 (2020).
- ⁴³R. Christensen, H. A. Hansen, and T. Vegge, "Identifying systematic DFT errors in catalytic reactions," *Catal. Sci. Technol.* **5**, 4946–4949 (2015).
- ⁴⁴J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding density functionals with machine learning," *Phys. Rev. Lett.* **108**, 253002 (2012).
- ⁴⁵F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the Kohn–Sham equations with machine learning," *Nat. Commun.* **8**, 872 (2017).
- ⁴⁶J. Hollingsworth, T. E. Baker, and K. Burke, "Can exact conditions improve machine-learned density functionals?," *J. Chem. Phys.* **148**, 241743 (2018).
- ⁴⁷N. Mardirossian and M. Head-Gordon, " ω B97M-V: A combinatorially optimized, range-separated hybrid, meta-GGA density functional with VV10 nonlocal correlation," *J. Chem. Phys.* **144**, 214110 (2016).
- ⁴⁸O. Schütt and J. VandeVondele, "Machine learning adaptive basis sets for efficient large scale density functional theory simulation," *J. Chem. Theory Comput.* **14**, 4168–4175 (2018).
- ⁴⁹Y. Li, H. Li, F. C. Pickard, B. Narayanan, F. G. Sen, M. K. Y. Chan, S. K. R. S. Sankaranarayanan, B. R. Brooks, and B. Roux, "Machine learning force field parameters from *ab initio* data," *J. Chem. Theory Comput.* **13**, 4492–4503 (2017).

- ⁵⁰J. P. Dürholt, G. Fraux, F.-X. Coudert, and R. Schmid, "Ab initio derived force fields for zeolitic imidazolate frameworks: MOF-FF for ZIFs," *J. Chem. Theory Comput.* **15**, 2420–2432 (2019).
- ⁵¹S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Sci. Adv.* **3**, e1603015 (2017).
- ⁵²S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nat. Commun.* **9**, 4618 (2018).
- ⁵³V. Botu, R. Batra, J. Chapman, and R. Ramprasad, "Machine learning force fields: Construction, validation, and outlook," *J. Phys. Chem. C* **121**, 511–522 (2016).
- ⁵⁴B. A. Calfa and J. R. Kitchin, "Property prediction of crystalline solids from composition and crystal structure," *AIChE J.* **62**, 2605–2613 (2016).
- ⁵⁵F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, "Machine learning energies of 2 million elpasolite (ABC₂D₆) crystals," *Phys. Rev. Lett.* **117**, 135502 (2016).
- ⁵⁶B. R. Goldsmith, J. Esterhuizen, J. X. Liu, C. J. Bartel, and C. Sutton, "Machine learning for heterogeneous catalyst design and discovery," *AIChE J.* **64**, 2311–2323 (2018).
- ⁵⁷O. Allam, B. W. Cho, K. C. Kim, and S. S. Jang, "Application of DFT-based machine learning for developing molecular electrode materials in Li-ion batteries," *RSC Adv.* **8**, 39414–39420 (2018).
- ⁵⁸A. Frontera, D. Quiñero, and P. M. Deyà, "Cation- π and anion- π interactions," *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **1**, 440–459 (2011).
- ⁵⁹A. C. Mater and M. L. Coote, "Deep learning in chemistry," *J. Chem. Inf. Model.* **59**, 2545–2559 (2019).
- ⁶⁰E. L. Willighagen, R. Wehrens, W. Melssen, R. de Gelder, and L. M. C. Buydens, "Supervised self-organizing maps in crystal property and structure prediction," *Cryst. Growth Des.* **7**, 1738–1745 (2007).
- ⁶¹T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *J. Mach. Learn. Res.* **11**, 743–746 (2010).
- ⁶²X. Ma, Z. Li, L. E. K. Achenie, and H. Xin, "Machine-learning-augmented chemisorption model for CO₂ electroreduction catalyst screening," *J. Phys. Chem. Lett.* **6**, 3528–3533 (2015).
- ⁶³M. H. S. Segler and M. P. Waller, "Neural-symbolic machine learning for retrosynthesis and reaction prediction," *Chem. Eur. J.* **23**, 5966–5971 (2017).
- ⁶⁴J. C. Cole, C. R. Groom, M. G. Read, I. Giangreco, P. McCabe, A. M. Reilly, and G. P. Shields, "Generation of crystal structures using known crystal structures as analogues," *Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater.* **72**, 530–541 (2016).
- ⁶⁵R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Cent. Sci.* **4**, 268–276 (2018).
- ⁶⁶A. Aspuru-Guzik, M.-H. Baik, S. Balasubramanian, R. Banerjee, S. Bart, N. Borduas-Dedekind, S. Chang, P. Chen, C. Corminboeuf, F.-X. Coudert, L. Cronin, C. Crudden, T. Cuk, A. G. Doyle, C. Fan, X. Feng, D. Freedman, S. Furukawa, S. Ghosh, F. Glorius, M. Jeffries-EL, N. Katsonis, A. Li, S. S. Linse, S. Marchesan, N. Maulide, A. Milo, A. R. H. Narayan, P. Naumov, C. Nevado, T. Nyokong, R. Palacin, M. Reid, C. Robinson, G. Robinson, R. Sarpang, C. Schindler, G. S. Schlau-Cohen, T. W. Schmidt, R. Sessoli, Y. Shao-Horn, H. Sleiman, J. Sutherland, A. Taylor, A. Tezcan, M. Tortosa, A. Walsh, A. J. B. Watson, B. M. Weckhuysen, E. Weiss, D. Wilson, V. W.-W. Yam, X. Yang, J. Y. Ying, T. Yoon, S.-L. You, A. J. G. Zarbin, and H. Zhang, "Charting a course for chemistry," *Nat. Chem.* **11**, 286–294 (2019).
- ⁶⁷A. F. de Almeida, R. Moreira, and T. Rodrigues, "Synthetic organic chemistry driven by artificial intelligence," *Nat. Rev. Chem.* **3**, 589–604 (2019).
- ⁶⁸P. S. Gromski, J. M. Granda, and L. Cronin, "Universal chemical synthesis and discovery with 'the chemputer'," *Trends Chem.* **2**, 4–12 (2020).
- ⁶⁹S. Steiner, J. Wolf, S. Glatzel, A. Andreou, J. M. Granda, G. Keenan, T. Hinkley, G. Aragon-Camarasa, P. J. Kitson, D. Angelone, and L. Cronin, "Organic synthesis in a modular robotic system driven by a chemical programming language," *Science* **363**, eaav2211 (2019).
- ⁷⁰R. L. Greenaway, V. Santolini, M. J. Bennison, B. M. Alston, C. J. Pugh, M. A. Little, M. Miklitz, E. G. B. Eden-Rump, R. Clowes, A. Shakil, H. J. Cuthbertson, H. Armstrong, M. E. Briggs, K. E. Jelfs, and A. I. Cooper, "High-throughput discovery of organic cages and catenanes using computational screening fused with robotic synthesis," *Nat. Commun.* **9**, 2849 (2018).
- ⁷¹C. Empel and R. M. Koenigs, "Artificial-intelligence-driven organic synthesis—En route towards autonomous synthesis?," *Angew. Chem., Int. Ed.* **58**, 17114–17116 (2019).
- ⁷²C. W. Coley, D. A. Thomas, J. A. M. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao, R. W. Hicklin, P. P. Plehiers, J. Byington, J. S. Piotti, W. H. Green, A. J. Hart, T. F. Jamison, and K. F. Jensen, "A robotic platform for flow synthesis of organic compounds informed by AI planning," *Science* **365**, eaax1566 (2019).
- ⁷³J. M. Granda, L. Donina, V. Dragone, D.-L. Long, and L. Cronin, "Controlling an organic synthesis robot with machine learning to search for new reactivity," *Nature* **559**, 377–381 (2018).