FRONTIERS OF PHYSICS

REVIEW ARTICLE
Volume 19 / Issue 1 / 13501 / 2024

Advances of machine learning in materials science: Ideas and techniques

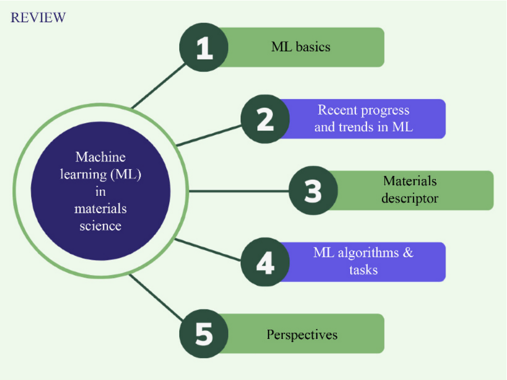
Sue Sin Chong[1,*], Yi Sheng Ng[1,*], Hui-Qiong Wang[1,2,†], Jin-Cheng Zheng[1,2,‡]

1 Department of New Energy Science and Engineering, Xiamen University Malaysia, Sepang 43900, Malaysia
2 Engineering Research Center of Micro-nano Optoelectronic Materials and Devices, Ministry of Education;
Fujian Key Laboratory of Semiconductor Materials and Applications, CI Center for OSED,
and Department of Physics, Xiamen University, Xiamen 361005, China
*These two authors contributed equally.
Corresponding authors. E-mail: †hqwang@xmu.edu.cn, ‡jczheng@xmu.edu.cn

Received July 27, 2022; accepted June 18, 2023

ABSTRACT

In this big data era, the use of large dataset in conjunction with machine learning (ML) has been increasingly popular in both industry and academia. In recent times, the field of materials science is also undergoing a big data revolution, with large database and repositories appearing everywhere. Traditionally, materials science is a trial-and-error field, in both the computational and experimental departments. With the advent of machine learning-based techniques, there has been a paradigm shift: materials can now be screened quickly using ML models and even generated based on materials with similar properties; ML has also quietly infiltrated many sub-disciplinary under materials science. However, ML remains relatively new to the field and is expanding its wing quickly. There are a plethora of readily-available big data architectures and abundance of ML models and software; The call to integrate all these elements in a comprehensive research procedure is becoming an important direction of material science research. In this review, we attempt to provide an introduction and reference of ML to materials scientists, covering as much as possible the commonly used methods and applications, and discussing the future possibilities.


REVIEW
Machine learning (ML) in materials science
1 ML basics
2 Recent progress and trends in ML
3 Materials descriptor
4 ML algorithms & tasks
5 Perspectives

Keywords machine learning, materials science

Contents

Higher Education Press

arXiv: 2307.14032
https://doi.org/10.1007/s11467-023-1325-z

1 Introduction

Commonly recognized as the fourth paradigm of science [1–4], machine learning (ML) has played a crucial role in the development of the data-driven scientific process, shaping the changes in experimental methodology, from measurement to data analysis, assisting the proving of mathematical theorem, and finding new discoveries in areas that were once deemed impossible. In the field of computational material science, the methods have been enriched by the efficient high-throughput ML-aided simulation/data generation and data-driven discovery [5, 6]. In experiments and materials synthesis, the advances in ML have helped researchers to efficiently analyze the data and identify hidden features within the large dataset [7–9].

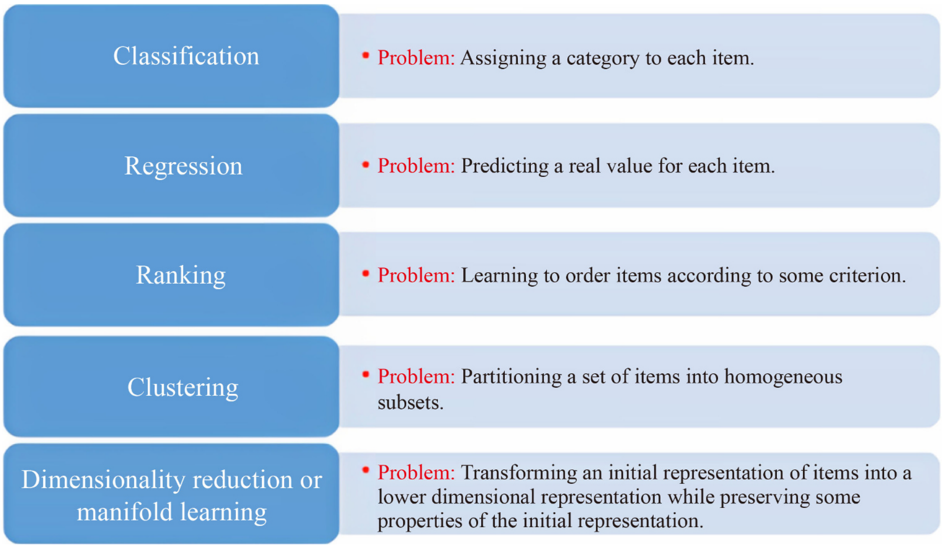
Discovery of new internal logics, patterns, or rules [10–13], and the study of complex systems, including nanostructures [14–29], alloys [30–35], superlattices [22, 36–38], surfaces [39–41], and interfaces [40, 42–46], as well as from materials to devices [47–48], are typical research topics in materials science. These areas could be addressed according to user specifications [49–50] by leveraging ML and big data statistical methods [51], which have advanced to a stage where users can utilize them to achieve large and complicated objectives with complex models. By breaking down broad objectives into smaller tasks, corresponding ML algorithms and objectives that are suitable can be identified and applied.

There are numerous comprehensive surveys on ML in material science [52–60]. In this review, we focus on the application of ML in material science, discussing the recent advances in ML, illustrating the basic principles of applying ML in materials science, and summarizing the current applications and briefly introducing the ML algorithms involved.

2 Basics on machine learning

ML has a long history [61–67]. However, it has only returned to the spotlight recently due to the compounding ability it has gained from the surge in big data and improving data infrastructure and computing power. Stemmed from statistical learning, ML has gained huge successes and popularity in many other tasks and has far-reaching influences in many fields, including physics, chemistry and material science. In this section, the basic ideas and concepts in ML and essential milestones in its illustrious history are covered.

ML can be broadly defined as computational methods using experience (available past information) to improve future performance or to make accurate predictions [68]. Typical ML methods involve three parts: the inputs (previously obtained data), outputs (predictions), and algorithms. The sample size (sample complexity) and the time & space complexity of algorithms are crucial for ML [68]. Therefore, the ML techniques are different from conventional methods such as experimental measurements or computer simulations, but are related to data analysis and statistics. In more general terms, ML techniques are data-driven methods, which combines the fundamental concepts of computer science with ideas from statistics, probability, and optimization. ML can be integrated with other disciplines, resulting in multi-discipline techiques such as quantum ML or physics-informed



Classification	• Problem: Assigning a category to each item.
Regression	• Problem: Predicting a real value for each item.
Ranking	• Problem: Learning to order items according to some criterion.
Clustering	• Problem: Partitioning a set of items into homogeneous subsets.
Dimensionality reduction or manifold learning	• Problem: Transforming an initial representation of items into a lower dimensional representation while preserving some properties of the initial representation.

Fig. 1 List of the conventional machine learning tasks and the problems tackled [68].

neural networks, materials science-based learning.

In ML, the "standard" or conventional learning tasks have been extensively studied, which include classification, regression, ranking, clustering, and dimensionality reduction or manifold learning [68]. The problems related to the above tasks are listed in Fig. 1. The definitions and terminology commonly used in ML for different learning stages are listed in Fig. 2. The typical stages of a learning process are also shown in Fig. 3, which can be briefly described as follows: with a given collection of labeled examples, one can firstly divide the data/samples into three groups, namely, training samples, validation data and test samples, then the relevant features associated to the desired properties are chosen, which are next used to train the pre-determined learning algorithm. This is done by adjusting the hyperparameters Θ in order to ensure that the hypothesis Θ_0 has the best performance on the validation sample. Typical learning scenarios include supervised learning, unsupervised learning, semi-supervised learning, transductive inference, on-line learning, reinforcement learning, active learning, and more other complex learning scenarios. Different from traditional data analysis, ML is fundamentally about generalization [68]. Spectacularly, the neural network-based ML is able to approximate functions in a very high dimension with unprecedented efficiency and accuracy [2], and therefore it can be used for complex tasks in a wide-range of applications.

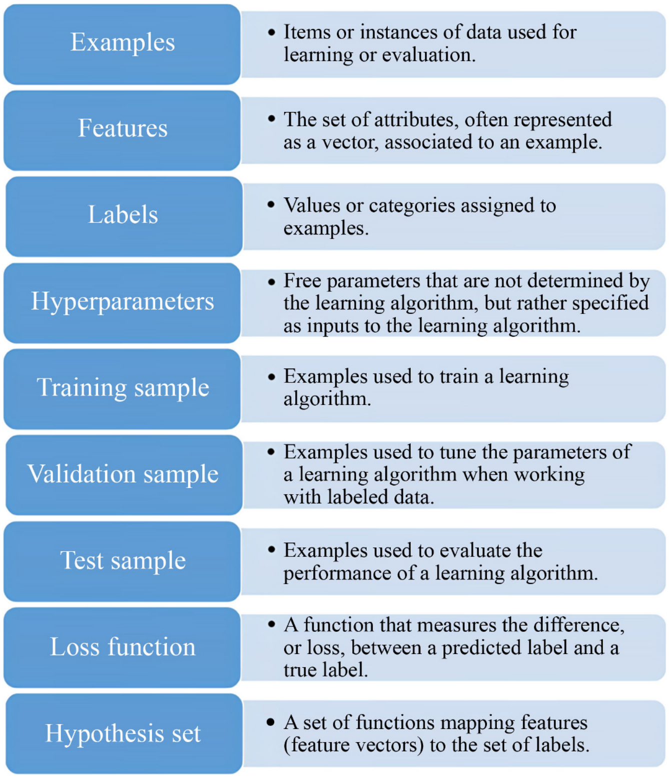
3 Recent progress in machine learning

Recently, the ML community has seen breakthroughs in many traditional AI tasks and classical challenging scientific tasks. This leap of improvement is being powered by both the new grounds in the underlying theory, the overall implementation and architecture, and the massive surge in data and data infrastructures. This section covers the advancement of ideas from various areas of applications of Artificial Intelligence – Natural Language Processing (NLP), Computer Vision (CV), Reinforcement Learning (RL), Explainability Artificial Intelligence (XAI), etc.

3.1 Classical machine learning application areas

In the field of natural language processing and understanding, ML models have made huge progress with Attention Transformer networks [69] and pre-training techniques. SuperGLUE [70] is a natural language understanding benchmark consisting of many tasks, which requires in-depth understanding of short proses and sentences. With superhuman performances at SuperGLUE benchmarks, it has been demonstrated that ML is able to model both understanding of natural language and generation of relevant natural language in context. The technique that has led to this leap in performance is pre-training [71], which refers to "training a model with one task to help it form parameters that can be used in other tasks". Prompt learning is a form of ML that works with large models, to learn knowledge from a language model simply by prompting the learnt model with various types of prompts. BERT-like models have also been extended to process data from realms outside natural language, like Programming Languages, e.g., CodeBERT [72], and Images [73], and had been very successful in these realms too. Table 1 lists works relevant to several main ideas in machine learning for Natural Language Processing (NLP).

Unsupervised learning has made strides in computer vision tasks, with models being able to identify subject in video, or identify poses of objects from point cloud in



Examples
- Items or instances of data used for learning or evaluation.

Features
- The set of attributes, often represented as a vector, associated to an example.

Labels
- Values or categories assigned to examples.

Hyperparameters
- Free parameters that are not determined by the learning algorithm, but rather specified as inputs to the learning algorithm.

Training sample
- Examples used to train a learning algorithm.

Validation sample
- Examples used to tune the parameters of a learning algorithm when working with labeled data.

Test sample
- Examples used to evaluate the performance of a learning algorithm.

Loss function
- A function that measures the difference, or loss, between a predicted label and a true label.

Hypothesis set
- A set of functions mapping features (feature vectors) to the set of labels.

Fig. 2 List of typical machine learning terminologies [68].

a video without labels. In the area of unsupervised learning for time-series data, ML models are able to effectively identify features from time-series data for both classification and prediction. Recently, there has been many work that extends the use of transformers to the characterization and prediction of extra-long timeseries sequences, Informer [99], Longformer [100]. Table 2 lists works relevant to several main ideas in ML for Computer Vision (CV).

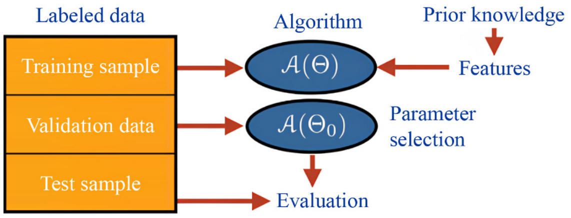


Fig. 3 Illustration of the typical stages of a learning process [68].

Reinforcement learning is the training of agents to make a sequence of reward-optimal decisions in an environment, often modelled as maximizing reward expectation in a partial Markov Decision Process (MDP). In reinforcement learning thrust, there is a huge improvement in the ability of state-of-the-art models to effectively navigate extra large search space to search for sequential actions to maximize task goals. Most notably, models like AlphaGo [116] and AlphaHoldem [117] have been able to navigate extra large search spaces with Monte Carlo search tree methods on latent representation of state space and action. Classical methods like State–Action–Reward–State–Action (SARSA) [118], Q-learning [119], TD-learning explores action space with a reward function, and learns a matching of state space to action, a policy or the q-value of the actions. In 2012, apprenticeship learning [120] initiated by Abbeel, proposes to define the architecture such that agent is able to learn directly from observations of a task being completed, instead of specifying the steps of a task. There is also a trend to integrate reinforcement learning with meta-learning in order to train multi-tasks agents to perform a variety of tasks [121, 122]. Table 3 lists works relevant to several main ideas in ML for Reinforcement Learning (RL).

Many human teaching or learning techniques have been the source of inspiration for advancement of this thrust. With the emergence of effective sampling techniques amongst others, the Efficient Zero [140] models have been able to progress by accumulating experiences through randomly playing against itself repeatedly, which improves the ability of game-playing models. Reinforcement learning has not only made breakthroughs in all-information open games; recently there has also been breakthrough in multi-player partial information games like Texas Hold'em and Alpha Hold'em [117]. For multi-agent reinforcement learning, the common benchmark task StarCraft Multi-Agent Challenge (SMAC) [141] can now be effectively completed by reinforcement

Table 1 Natural language processing (NLP) ideas, techniques and models.

Ideas & technique	Relevant development and models
Pre-training	Ref. [71], Ref. [74], BLIP [75], Pretrained transformers [76], Ref. [77]
Fine-tuning	Ref. [78], Ref. [79]
Bidirectional encoder	BERT [80], Albert [81], Robustly optimized BERT pre-training approach (RoBERTa) [82], CodeBERT [72], BeiT [73]
Transformer	Ref. [69], Ref. [83], Ref. [84], Transfomer memory as search index [85]
Attention prompt	Ref. [86], AutoPrompt [87], OpenPrompt [88]
Learning extra huge models	Open pretrained transfomer (OPT 175B) [89], Jurassic-1 [90], Generative pre-trained transformer 3 (GPT-3) [91], CLD-3 [92]
End-to-end model	Word2Vec [93], Global vectors for word representation (GLoVE) [94], Context2Vec [95], Structure2Vec [96], Driver2Vec [97], wav2Vec [98]

Table 2 Computer vision (CV) ideas, techniques and references.

Ideas & techniques	Relevant literature
Visual models	Visual transformer [101], Flamingo (LM) [102]
Image-text processing	CoCa [103], FuseDream [104], CLIP [105]
Convolutional neural network	DEtection transformer (DETR)[106], LiT [107], Ref. [108]
Image rendering	Dall-E [109], Review [110], Neural radiance field (NeRF) [111]
Point cloud reconstruction	PointNorm [112], Ref. [113], Residual MLP [114], Learning on point cloud [115]

Table 3 Reinforcement learning (RL) ideas, techniques and references.

Ideas & techniques	Relevant literature
Types of RL	Q-learning [119], SARSA [118], Temporal difference (TD)-learning [123]
RL algorithm	Self-training [124], Deep Q-learning (DQN) [125], Deep deterministic policy gradient (DDPG) [126], Offline [127]
Apprenticeship learning efficient RL	SayCan[128], Q-attention [129], Imitation learning [130] Replay with compression [131], Decision transformer [132]
Evolving curriculum	Adversarially compounding complexity by editing levels (ACCEL) [133], Paired open-ended trailblazer (POET) [134], Autonomous driving scene render (READ) [135]
Bandit problem	Bandit learning [136], Batched bandit [137], Dueling bandit [138], Upper confidence bound (UCB) [139]

learning models, which effectively decomposes the cooperation task into role-learning by large neural networks [142] amongst many other techniques [143, 144]. This is a breakthrough for multi-agent reinforcement learning.

3.2 On quantum machine learning

Quantum ML is one of the big next steps of ML [145]. While error-correction still limits our ability to build a fully quantum computer, it is possible to innovate with hybrid algorithms that uses quantum sub-algorithms or components to speed-up, robustify ML or simply to expand the theoretical boundaries of ML with 2 norm probabilities. In quantum computing, we can compute the similarity between feature vectors with state overlaps (denoted by bra and ket) instead of kernels via inner product. Consider a simple quantum ML scenario below:

$$\begin{aligned} K\left(x, x^{\prime}\right) &= \langle\phi(x), \phi\left(x^{\prime}\right)\rangle \rightarrow QKE\left(x, x^{\prime}\right) \\ &= \langle\phi(x) \mid \phi\left(x^{\prime}\right)\rangle . \end{aligned} \tag{3.1}$$

The feature space in quantum ML can be obtained by state preparation. For instance,

$$\phi:\left[x_1 ; x_2\right] \rightarrow\left(x_1|0\rangle+x_2|1\rangle\right) \otimes\left(x_1|0\rangle+x_2|1\rangle\right) . \tag{3.2}$$

The corresponding circuit is denoted by

$$S_x^{AB}\left(|0\rangle_A \otimes|0\rangle_B\right)=S_x^A|0\rangle_A \otimes S_x^B|0\rangle_B . \tag{3.3}$$

We can have quantum kernel estimation [see Eq. (1)] [146], or quantum feature spaces [147] in hybrid ML algorithms or intermediate scale hybrid machines [148]. This offers a new insight to the types of kernels and linear algebra that we can use to improve ML in the classical sense. Quantum physics or chemistry can be more effectively simulated in the primitive sense using quantum ML algorithm. The hybrid ML coupled with quantum material science is potentially an important stepping stone for material scientists and computer scientists alike to innovate and research more efficiently.

3.3 Theory, explainable AI and verification

In classical computer science, the very hard case of Travelling Salesman Problem (TSP), a classical NP problem, has been solved with very satisfactory result based on neural networks, which either blend with pre-training of a solver of a mini-TSP or a reinforcement learning-based [149] strategy selector combined with heuristic. Other prominent NP problems like Maximum Independent Set (MIS) or Satisfiability Modulo Test (SMT) have also been solved satisfactorily with ML-guided heuristic search [150]. This demonstrates that ML models have been able to push through boundaries that have been set forth by traditional theoretical computer science. This breakthrough has been made possible by effective latent representation learning of essential features of the problem itself and the solver.

Explainability XAI techniques like Integrated Gradients (IG) [151], Local Interpretable Model-agnostic Explanations (LIME)[152], Shapley Additive Explanations (SHAP)[153], SimplEx [154] and various others have gained much attention. LIME attempts to identify hot areas in the image responsible for features that result in the prediction. SimplEx [154] is an explainability technique that attempts to explain a prediction with linear combinations of samples drawn from the corpus of training data; the technique returns a combination of training samples that has contributed to the predictions. There are also efforts to incorporate explainability by adding a layer at the end of neural networks for capturing explainability information. Explainable Graph Neural Network NN) techniques that apply specifically to Graph Neural Networks are broadly classified into several classes: Gradients/features based Guided Back-propagation (BP) [155], Perturbation Based GNN Explainer [156], SubgraphX [157], Decomposition Based, Surrogates GraphLIME [158] and Generation [159]. These GNN XAI techniques are well-suited for explaining feature importance for predictions at either the node level, edge level or graph level.

Verification is important for protecting neural network models against adversarial behaviours; adversary behaviours can be characterized by ill-intent shifts of planes of separation in the model so that it is more likely to err on otherwise correctly classified samples or corrupting input samples with noise or otherwise. Neural network robustness verification techniques like Rectified Linear Unit-Plex (ReLUPlex) [160] and alpha-beta CROWN [161] have also made huge progress. It is a numerical bounds back-propagation technique where the score boundaries for each class are back-propagated throughout the network to determine the overlap between class scores. Specifically, in the non-linear portions of the neural network, the ReLU activation functions were bounded with linear functions. Safety-critical applications have also been secured with neural network verification techniques, and the Airborne Collision Avoidance System for Unmanned Aircraft (ACAS Xu) [162] is an ensemble of 45 neural networks whose purpose is to give anti-collision advice to flying planes, and utilize ReLUplex methods to make their advice robust.

3.4 Stack optimizations for deep learning

Graphical Processing Units (GPU) are processors capable of parallel processing instructions. Standard GPU deep learning speedup techniques include convolutional layer reuse, featuremap reuse and filter reuse, and memory access is a common bottleneck [163]. The basic idea is that functions that are computed many times should be optimized on all levels, from high to low, including the instruction set level. The entire software stack, compiler technologies, and code generation have been optimized for deep learning computations on GPU. Deep learning GPU is known for its high energy usage; reducing energy usage is an essential objective for GPU optimization research [164]. The requirement for the scale of hardware architecture for ML is also loosening up, as engineers are packing engineering insights from large systems into smaller and energy-conserving systems, TensorFlow Lite Micro [165].

ML theory and practice have made massive progress in recent years. It is now transforming the scientific methods and has become deeply integrated with many scientific and humanities [166] fields. Application-wise, ML models have been trusted to make more and more crucial decisions for the well-functioning of society. For instance, in the criminal justice setting [167], ML models have been used to set bail for defendants; in the finance sector, models can help make decisions [168]; in the energy sector, they predict power generation efficiency for wind power stations. While neural network might still be a black box and can be hard to verify at times, its effectiveness as a predictor and sometimes generator has already been relied upon by many societal sectors for greater efficiency and effectiveness.

4 Development trend of machine learning for materials science

ML has helped material scientist achieve their study aims in a wide variety of tasks, most prominently as a screening tool in the design of a large variety of materials, which include energy materials, semiconductors, polymer design, catalyst, high entropy alloy, etc. The trend of going from processing a single dataset to achieving a specific aim to learning a latent representation of the underlying structure, which can later be finetuned to perform specific tasks, such as predicting the energetically stable structure across datasets, is rather prominent.

4.1 From numerical analysis to feature engineering

Traditionally, ML has been used as an advanced numerical regression tool to analyse experimental data in material science and many other fields [169, 170]. The remarkable ability of ML to interpolate data has allowed scientists to explain phenomena and verify hypotheses effectively.

Traditional material science ML practitioners often concern themselves with explicit feature engineering of specific materials [171]. Bhadeshia [171] has outlined four categories of models in material science; traditionally ML models are “models used to express data, reveal patterns, or for implementation in control algorithm”. The classical works that involve material property prediction mostly fall into the fourth category. Figure 4 illustrates the feature engineering process for material science, which encompasses four stages: feature extraction; feature analysis; correlation and importance analysis; and feature selection [172].

In material space, there are many degrees of freedoms, such as the atomic coordinates, coordination numbers, interatomic distances, and the position of the various species. Often, they are impractical to be used as the direct inputs to the algorithms, as they are not invariant under translation and rotation. In feature extraction [see Fig. 4(a)], we seek to convert them into descriptors,

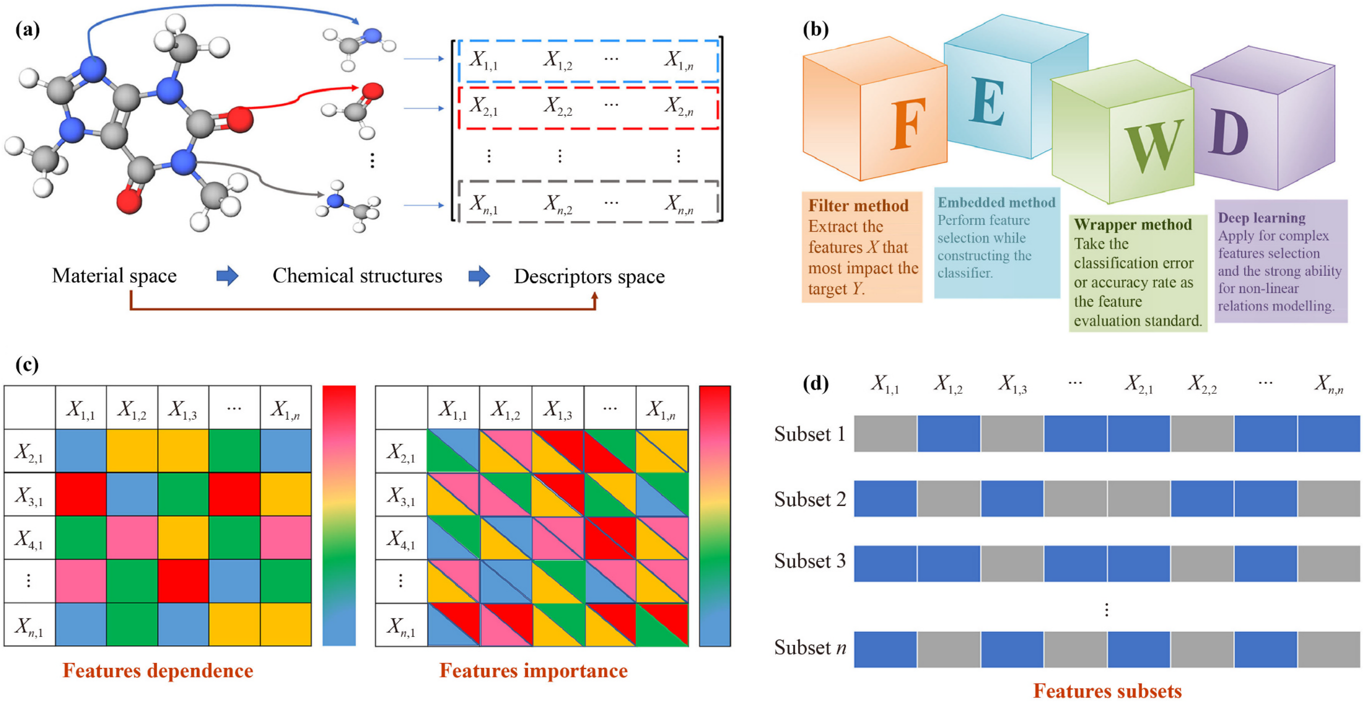

(a)
$X_{1,1}$ $X_{1,2}$ ⋯ $X_{1,n}$
$X_{2,1}$ $X_{2,2}$ ⋯ $X_{2,n}$
$X_{n,1}$ $X_{n,2}$ ⋯ $X_{n,n}$
Material space
Chemical structures
Descriptors space
(b)
F
E
W
D
Filter method
Extract the features X that most impact the target Y.
Embedded method
Perform feature selection while constructing the classifier.
Wrapper method
Take the classification error or accuracy rate as the feature evaluation standard.
Deep learning
Apply for complex features selection and the strong ability for non-linear relations modelling.
(c)
$X_{1,1}$ $X_{1,2}$ $X_{1,3}$ ⋯ $X_{1,n}$
$X_{2,1}$ $X_{3,1}$ $X_{4,1}$ ⋮ $X_{n,1}$
Features dependence
Features importance
(d)
$X_{1,1}$ $X_{1,2}$ $X_{1,3}$ ⋯ $X_{2,1}$ $X_{2,2}$ ⋯ $X_{n,n}$
Subset 1
Subset 2
Subset 3
Subset n
Features subsets

Fig. 4 Feature engineering for ML applications. **(a)** Feature extraction process. Starting from material space, one can extract information from material space into chemical structures then to descriptors space. **(b)** Typical ML feature analysis methods. "FEWD" refers to Filter method, Embedded method, Wrapper method, and Deep learning. **(c)** Correlation and importance analysis of selected features. The feature correlations is visualized in the diagram on the left. Diagram on the right is normalized version of left diagram, where the colors indicate the relative correlation of every other feature for prediction of the row/column feature. **(d)** Various feature subsets obtained from feature engineering analysis. One can construct features with linearly independent combination of subsets, in other words, subsets of features are basis. Reproduced with permission from Ref. [172].

which extract the underlying symmetry and distinguish systems that are truly different and not just a product of translations and/or rotations.

After the features are extracted, they undergo a series of analysis to fine tune and reduce the dimensionality of the descriptors space. The four commonly used methods, shown in Fig. 4(b), are the filter method, embedded method, wrapper method, and deep learning method. With the analysis process completed, a mapping, as illustrated in Fig. 4(c), which relates the importance and correlations among the selected features, can be used to visualize their dependence. In turn, this aids the process of feature selection, in which many suitable subsets of features [see Fig. 4(d)] are chosen to proceed to the next stage – fed into the ML algorithm and compared to obtain the best performing minimal subset.

4.2 From feature engineering to representation learning

While explicit feature engineering is a practical and valuable task, it often restricts the type of task that ML can perform and does not fully use its ability to learn a generalized representation or sound separation of features and ability to interpolate or extrapolate along those dimensions. Moreover, the task of sifting through a vast dataset is laborious and hard to manage for individuals. Furthermore, with the ever-expanding computing power, the dimensionality of the features that is computationally feasible also rapidly scales up, allowing the consideration of more factors, which ultimately improves the accuracy of the prediction while also widening the coverage of material types screened. Thus, there is a push towards representation learning, an automation of feature engineering of a large material dataset [173], which better captures the internal latent features [174]. This trend encouraged a deeper integration in both development trends in ML and material science, coupled with a concise selection of ML tools, which require an intuitive understanding of mathematical and theoretical computer science ideas behind these tools.

In representation learning, the features are automatically discovered and extracted from the raw data, and thus complicated patterns that are hidden from the human user but are highly relevant could boost the accuracy and effiecieny of the ML model, which is highly dependent on the quality of the selected features. Therefore, representation learning excels in applications where the data dimensionality is high and features extraction is difficult,

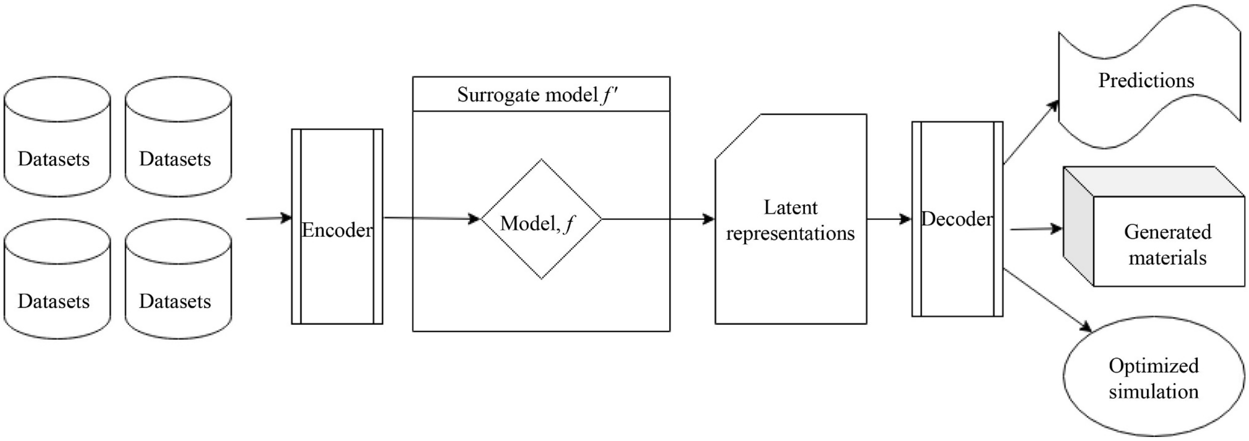


Fig. 5 Infographic of End-to-End Model. End-to-End models take multi-modal dataset as inputs, and encodes them into vectors for the surrogate model. The surrogate model then learns the latent representation, which makes the internal patterns of these datasets indexable. One is then able to decode the latent representation into an output form of our choice, which includes property predictions, generated novel materials and co-pilot simulation engines.

such as speech recognition and signal processing, object recognition, and natural language processing [175].

Neural networks can be packed into layers or attention blocks that can be integrated into a single neural network. Effective embedding of information that is a dimensional reduction tool reduces the complexity of the model, when upended upon the training pipeline, brings us to end-to-end learning. Figure 5 shows a simplified pipeline for material science end-to-end model, where datasets are turned into vectors by the encoder to use as the input for the surrogate model, which attempts to identify the latent representation that can be decoded to generate predictions.

Representation learning has been applied in materials science. By using the raw experimental X-ray absorption near edge structure (XANES) spectra, Routh *et al.* [176] managed to obtain latent features after performing unsupervised ML methods. The raw experimental data are fed into an autoencoder that includes the encoder and decoder, which uses the input data as the output data, while information is passed through a bottleneck layer, as illustrated in Fig. 6.

4.3 From representation learning to inverse design

After learning the representations that are critical in influencing the functionality of the materials, we ought to think: could we use them inversely, to generate novel and maybe better materials? This question had been sought in 1999 by Franceschetti and Zunger [177], where they successfully searched for the alloy of fixed elements with targeted electronic structure, using Monte Carlo

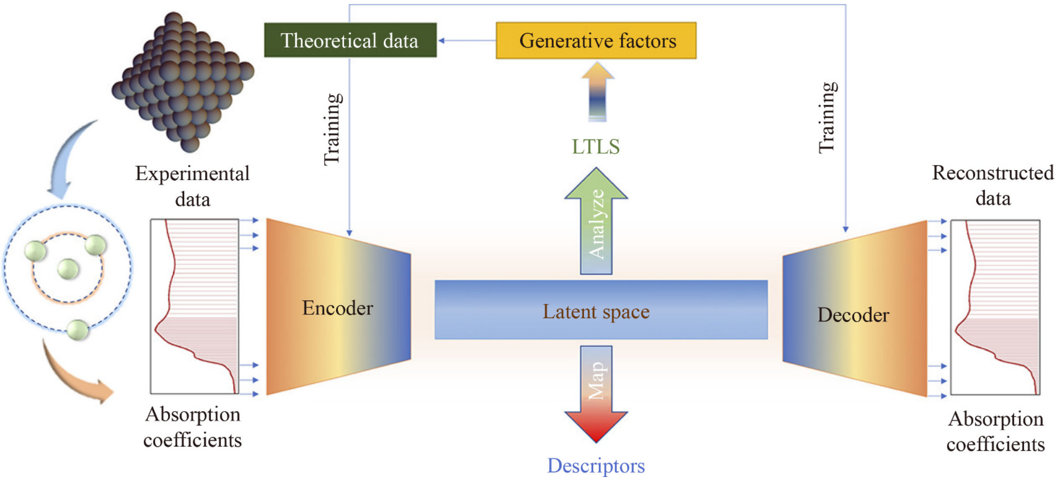


Fig. 6 Schematic of the representation learning methods used in the structural characterization of catalysts, where the autoencoder, which includes the encoder and decoder, is used, with the input and output data being the same. Reproduced with permission from Ref. [176].

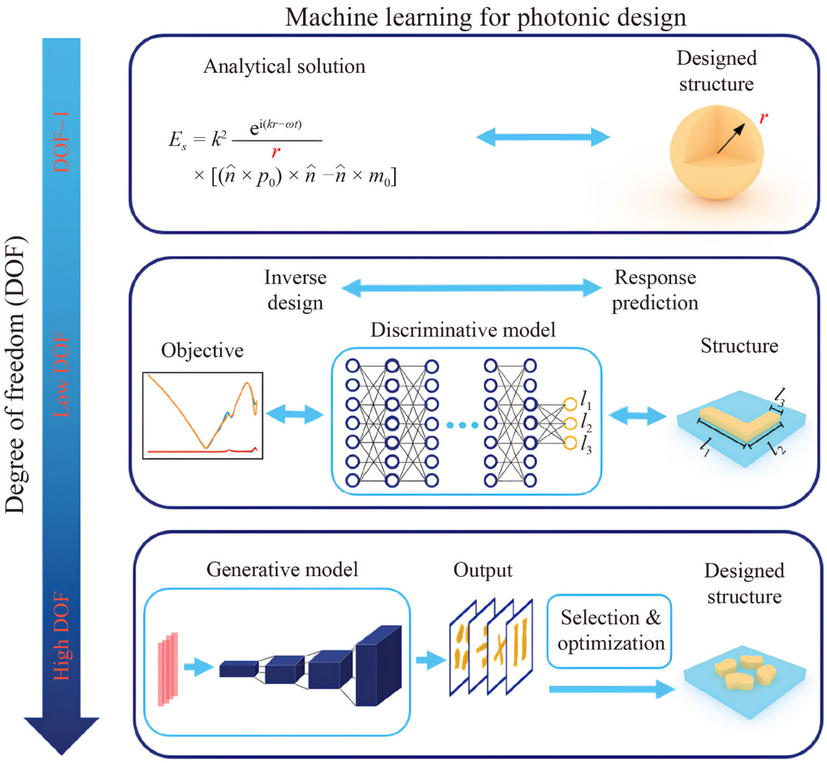


Fig. 7 Depending on the degree of freedom (DOF) involved, the machine learning methodologies of the photonic design vary. The analytical methods that are suitable for DOF of order unity are replaced by the discriminative model of ML. As DOF increases, generative model is leveraged to bring down the dimensionality. Reproduced with permission from Ref. [178].

method only. This limited yet profound results show us the vast usefulness of solving the inverse problem. Now, armed with the computation power and advancement in ML, we are in a better position to answer this question. Generative models like Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) have been applied in the inverse design of molecules and solid-state crystal.

By combining the power of representation learning and generative models into a single extensive model, that is the joining of neural networks from several parts of the workflow into a single network, many benefits can be reaped. First of all, the ability of an extensive network to counter noise levels in the training dataset, resulting in better predictions or better-generated solutions. Secondly, the latent representation learnt from each part of the pipeline is more consistent with the final goal of experimentation or design. Thirdly, the absence of human error-prone non-ML intervention helps experimenters focus on the overall goal and architecture.

By using discriminative models, generative models, and rapid simulation, whether standalone or in combination, one can construct sophisticated models that tackle problems ranging from predicting density functional theory (DFT) properties to inverse device design with confidence. One can also explore material design at different scale and granularity with ML model as an aide. An example is shown in Fig. 7, where both discriminative and generative models are used jointly to design photonics. When the dimensionality of the photonic structures involved is very low, at the order of 1, analytical methods are well-suited. However, as the dimensionality increases, the analytical methods are no longer feasible, and the ML methods are required. On its own, discriminative models are suitable at slightly larger parameters space, but when the degree of freedom scales up considerably, generative model can be employed to reduce the dimensionality.

5 Databases in material science

Data is prevalent in material science; data which originates from every aspect and process of material science research endeavour have varying types, accuracy and genre. Table 4 lists the typical data types and database that are used in ML models. A material science task often includes processing a combination of data types listed.

The broad spectrum of data types and multi-modules of input data dictates that material science models need to learn to integrate multi-modal data to produce meaningful research results. This trend also means that the

Table 4 Typical material science databases.

Data type	Database
Computational data	OQMD: Materials properties calculated from DFT [179,180], Materials project [181], Joint automated repository for various integrated simulations (JARVIS) [182], AFLOW [183], MatCloud [184], MPDS [185], NOMAD [186], C2DB [187], 2DMatPedia [188]
Crystallographic data	ICSD [189], Crystallography open database (COD) [190], The NIST surface structure database (SSD4) [41], Aspherical electron scattering factors [191], AlphaFold [192]
Imaging/spectra data	MatBench [193], TEMImageNet [194], Single-atom library [195]
Other types	Knowledge graph, e.g., propnet [196]

Table 5 Machine learning libraries. All descriptions were adapted from the references therein.

Library	Library	Description
General deep learning libraries (APIs)	Deepmind Jax [201]	Open ML codebase by Deepmind. With its updated version of Autograd, JAX can automatically differentiate native Python and NumPy code.
	Keras [202]	Free open source Python library for developing and evaluating deep learning models.
	PyTorch [203]	PyTorch is an open source machine learning framework based on the Torch library.
	TensorFlow [204]	Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning.
Useful libraries for machine learning tasks	HuggingFace [205]	Open NLP Library with Trained Models, API and Dataset Loaders.
	OpenRefine [206]	OpenRefine is an open-source desktop application for data cleanup and transformation to other formats, an activity commonly known as data wrangling.
	PyTorch Geometric [207]	PyG (PyTorch Geometric) is a library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data.
	PyTorch lightning [208]	PyTorch lightning is the deep learning framework for professional AI researchers and machine learning engineers who need maximal flexibility without sacrificing performance at scale.
	VectorFlow [209]	Optimized for sparse data in single machine environment.
	Weights & Biases [210]	W&B for experiment tracking, dataset versioning, and collaborating on ML projects.
Tools that might be useful to material science	Dscribe [211]	Provides popular feature transformations (“descriptors”) for atomistic materials simulations, including Coulomb matrix, Ewald sum matrix, sine matrix, Many-body tensor representation (MBTR), Atom-centered symmetry funsction (ACSF) and Smooth overlap of atomic positions (SOAP).
	Open graph database [212]	The open graph benchmark (OGB) is a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs. OGB datasets are automatically downloaded, processed, and split using the OGB Data Loader.
	RDKit [213]	Opensource library for converting molecules to SMILES string.
	Spektral [214]	Spektral is a Python library for graph deep learning, based on the Keras API and TensorFlow 2.

material science community needs to embrace the software and statistical revolution that will propel the field forward.

In order to use computer systems to process material information, material-related nomenclatures have to adapt to computer processing norms, like string. Both atomic and structure of molecules should be evident by parsing strings, e.g. Simplified molecular-input line-entry (SMILES), BigSMILES [197], Self-referencing embedded strings (SELFIES) [198], Physical Information File (PIF) [199]. Material Science datasets are often implemented in neural network data loaders like Deep Graph Library [200]. The ML community’s Datasets have codebases that organize information that eases software engineers to call and process with a library. Most quantum chemistry software is softwareengineering based Application Programming Interface (API) to share and process Quantum Chemistrydata; it is written to store quantum calculation data with well-tested and scalable database norms (like schema) and eases or speedup data batch processing. The basis for quantum chemistry libraries is standardized; typical ones include Gaussian Orbital Basis (GTO), Plane Wave Basis (PW), and Numerically Tabulated Atom-centered Orbitals (NAO). Table 5 lists softwares which might be useful. The first portion lists general deep learning libraries (APIs), second portion lists useful libraries for machine learning tasks, third portion lists tools that might be useful to material science.

6 Machine learning descriptors for material science

The material science datasets are often comprised of atomistic information with the coordinates of atoms, the

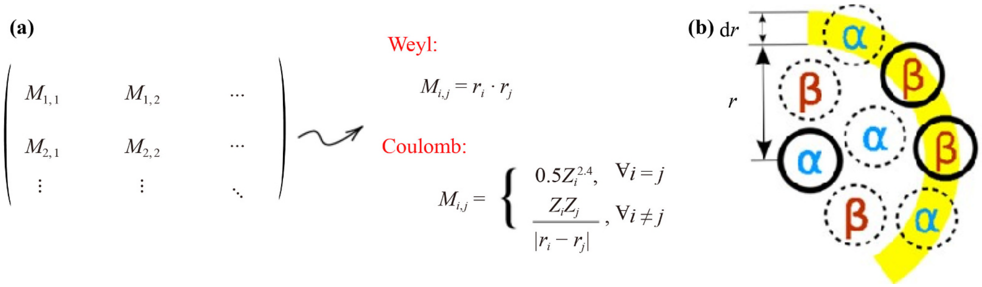

(a)
$M_{1,1}$
$M_{1,2}$
$\cdots$
$M_{2,1}$
$M_{2,2}$
Weyl:
$M_{i,j} = r_i \cdot r_j$
Coulomb:
$M_{i,j} = \begin{cases} 0.5Z_i^{2.4}, & \forall i = j \\ \frac{Z_i Z_j}{|r_i - r_j|}, & \forall i \neq j \end{cases}$
(b) dr
r
α
β

Fig. 8 **(a)** The mathematical description of the Weyl and Coulomb matrices. **(b)** The construction of the PRDF sums, where atoms covered by the yellow strip covering the radius $(r, r + \mathrm{d}r)$ are considered. (b) Reproduced with permission from Ref. [221].

charges on the atoms, and their compositions. To capture the spatially invariant information, the local environment on atomic scale is oftenly extracted, such as the list of neighbouring atoms and their relative spatial positions. They are then compactified and propagated as descriptors in the form of a vector, in a neural network which maps this information to their properties that are of interests: total energy, mass density, bulk moduli, etc. In general, a good descriptor needs to have the following qualities:

i) Invariant under spatial transformation (arbitrary translations, rotations, and reflections)

ii) Invariant under permutation/exchange of atoms of identical species, i.e., only a unique representation for each arrangement of atoms.

iii) Computationally cheap and easy to implement.

iv) Minor deviation under small perturbations in the atomic structure.

Clearly, the Cartesian coordinates of the atoms do not satisfy the points i) and ii), even though it is the easiest imaginable method. There are many different descriptors that have been tried and tested in material science, which we will attempt to briefly summarize in this section, but it is by no means exhaustive. For further information and use examples on descriptors, the reader is recommended to the articles of Li *et al.* [215] and Schmidt *et al.* [216].

6.1 Pair-wise descriptor

Pair-wise descriptor is a type of descriptor that considers each and every possible pair of atoms in the system. Examples include Z-matrices, Weyl matrices, and more recently, the Coulomb matrices [216]. A figure briefly describing the Weyl matrices and Coulomb matrices are shown in Fig. 8(a). In the work of Rupp *et al.* [217], Coulomb matrices were constructed for a set of organic molecules that are numerically extracted and sorted descendingly, then the Euclidean difference between the vectors of eigenvalues are computed and defined as the distance between two molecules (with different dimensions accounted for by adding trailing zeroes to vectors). Using this as the sole descriptor, they developed a ML approach for fast and accurate prediction of molecular atomization energy. The same eigenvalue-based method has also been used in a number of recent studies [218, 219]. The downsides of this method are the inability to differentiate enantiometer [220] and the loss of information, as the dimensions are reduced from N^2 to N, which can sometimes be an advantage [219].

As described, the Coulomb matrices methods are only viable for finite system. To extend the pairwise descriptor to infinite periodic system, Faber *et al.* [220] proposed three different methods: Ewald sum matrices, Sine matrices, and Extended coulomb-like matrix, and their results show that Sine matrix is the most efficient and outputs the smallest error.

Another alternative, the partial radial distribution function (PRDF) was proposed by Schütt *et al.* [221] and used in their work to perform fast prediction of density of states at Fermi level for different types of solids. The pairwise distances $d_{\alpha\beta}$ between two atoms type are considered, in the following equation for PRDF:

$$g_{\alpha\beta}(r) = \frac{1}{N_\alpha V_r} \sum_{i=1}^{N_\alpha} \sum_{j=1}^{N_\beta} \theta\left(d_{\alpha_i\beta_j} - r\right) \theta\left(r + \mathrm{d}r - d_{\alpha_i\beta_j}\right), \tag{6.1}$$

where $\theta(x)$ is the step function, V_r is the volume of the primitive cell, while N_α and N_β are the number of atoms of types α and β. Only the atoms in the primitive cell are considered as the shell center, i.e., the atoms α_i, see Fig. 8(b). This function is invariant under translation, rotation, and different choice of the unit cell.

6.2 Local descriptor

The most intuitive methods to describe a system of atoms that also take into the geometrical aspect into account is the neighbour-based or local descriptor, as the electron density is only weakly affected by distant atoms. By considering the neighbouring atoms of a selected atom within a pre-determined cutoff radius, we can store the information about their bonds, such as the bond distance and angle.

Behler and Parinello [222] proposed the use of two symmetry functions, the radial symmetry function G_i^1 and angular symmetry function G_i^2:

$$G_i^1 = \sum_{j \neq i}^{all} \mathrm{e}^{-\eta (R_{ij} - R_s)^2} f_c\left(R_{ij}\right), \tag{6.2}$$

$$\begin{aligned} G_i^2 = & 2^{1-\zeta} \sum_{j,k \neq i}^{all} \left(1 + \lambda \cos\theta_{ijk}\right)^{\zeta} \\ & \times \mathrm{e}^{-\eta\left(R_{ij}^2 + R_{ik}^2 + R_{jk}^2\right)} f_c\left(R_{ij}\right) f_c\left(R_{ik}\right) f_c\left(R_{jk}\right), \end{aligned} \tag{6.3}$$

where R_{ij} is the distance between atom i and j, θ_{ijk} is the angle between the three atoms i, j, k. There are four free parameters in total, $\lambda (= +1, -1)$, η, ζ, and the implicit R_c in f_c, defined as

$$f_c\left(R_{ij}\right) = \begin{cases} 0.5 \times \left[\cos\left(\dfrac{\pi R_{ij}}{R_c}\right) + 1\right] & \text{for } R_{ij} \leq R_c; \\ 0 & \text{for } R_{ij} > R_c. \end{cases} \tag{6.4}$$

The symmetry functions capture the local environment of an atoms, are invariant to permutation, translation, rotation, and changes in coordination number. They have been used in reproducing potential energy surface (PES) at DFT accuracy. This formalism was extended and studied in extensive details in Behler [223], where the set of symmetry functions are coined the "Atom-centered Symmetry Functions (ACSFs)". A further generalization was done by Seko *et al.* [224], which included basis functions other than the Gaussian in Eq. (2), such as Neumann functions and Bessel functions. They also introduced the use of the Least Absolute Shrinkage and Selection Operator (LASSO) technique to optimize the basis set and find the sparsest representation to speed up computation. This was successfully used to reproduce almost DFT-accuracy phonon dispersion and specific heat for hcp Mg. A more recent work to reduce the undesirable scaling in ASCF has also been discussed [225].

Another approach using bispectrum, a three-point correlation function, was introduced by Bartók *et al.* [226]. In this approach, they first construct local atomic density function for each atom i, as

$$\rho_i\left(\boldsymbol{r}\right) = \delta\left(\boldsymbol{r}\right) + \sum_j \delta\left(\boldsymbol{r} - \boldsymbol{r_{ij}}\right) f_c\left(\left|\boldsymbol{r_{ij}}\right|\right), \tag{6.5}$$

where the $\delta\left(\boldsymbol{r}\right)$'s are the Dirac Delta function. This atomic density is then projected onto the surface of a 4D sphere, by expanding the atomic density using 4D spherical harmonics, $U_{m'm}^j$ (index i omitted):

$$c^{j'}_{m'm} = \langle U_{m'm}^j | \rho \rangle, \tag{6.6}$$

and the bispectrum is then built from these coefficients, defined as

$$\begin{aligned} B_{j_1,j_2,j_3} = & \sum_{m_1',m_1=-j_1}^{j_1} \sum_{m_2',m_2=-j_2}^{j_2} \sum_{m',m=-j}^{j_1} \left(c_{m'm}^j\right)^* \\ & \cdot C_{j_1 m_1 j_2 m_2}^{jm} C_{j_1 m_1' j_2 m_2'}^{jm'} c_{m_1' m_1}^{j_1} c_{m_2' m_2}^{j_2}, \end{aligned} \tag{6.7}$$

where the $C_{j_1 m_1 j_2 m_2}^{jm}$'s are the ordinary Clebsch–Gordan coefficients of the $SO(4)$ group.

The Smooth Overlap of Atomic Positions (SOAP) descriptor [226] uses the atomic density defined in Eq. (5), but with the Dirac Delta function replaced by the Gaussians, expanded in terms of spherical harmonics:

$$\begin{aligned} \exp\left(-\alpha |\boldsymbol{r} - \boldsymbol{r_i}|^2\right) = & 4\pi \exp\left[-\alpha\left(r^2 + r_i^2\right)\right] \\ & \cdot \sum_{lm} h_l\left(2\alpha r r_i\right) Y_{lm}\left(\widehat{\boldsymbol{r}}\right) Y_{lm}^*\left(\widehat{\boldsymbol{r_i}}\right), \end{aligned} \tag{6.8}$$

where h_l's are the modified spherical Bessel functions of the first kind and Y_{lm} is the spherical harmonics. A similarity kernel $k\left(\rho, \rho'\right) \equiv \int \mathrm{d}\widehat{R} \left| \int \rho\left(\boldsymbol{r}\right) \rho'\left(\widehat{\boldsymbol{R}}\boldsymbol{r}\right) \mathrm{d}\boldsymbol{r} \right|^n$ was introduced to compare two different atomic environments, where $n = 2$ is used in their study. The normalized kernel or SOAP kernel

$$K\left(\rho, \rho'\right) = \left(\frac{k\left(\rho, \rho'\right)}{\sqrt{k\left(\rho, \rho\right) k\left(\rho', \rho'\right)}}\right)^{\xi}, \tag{6.9}$$

where ξ is any positive integer, chosen to control the sensitivity, goes into the PES of the form

$$\varepsilon\left(\boldsymbol{q}\right) = \sum_{k=1}^{N} \alpha_k K\left(\boldsymbol{q}, \boldsymbol{q}^{(k)}\right), \tag{6.10}$$

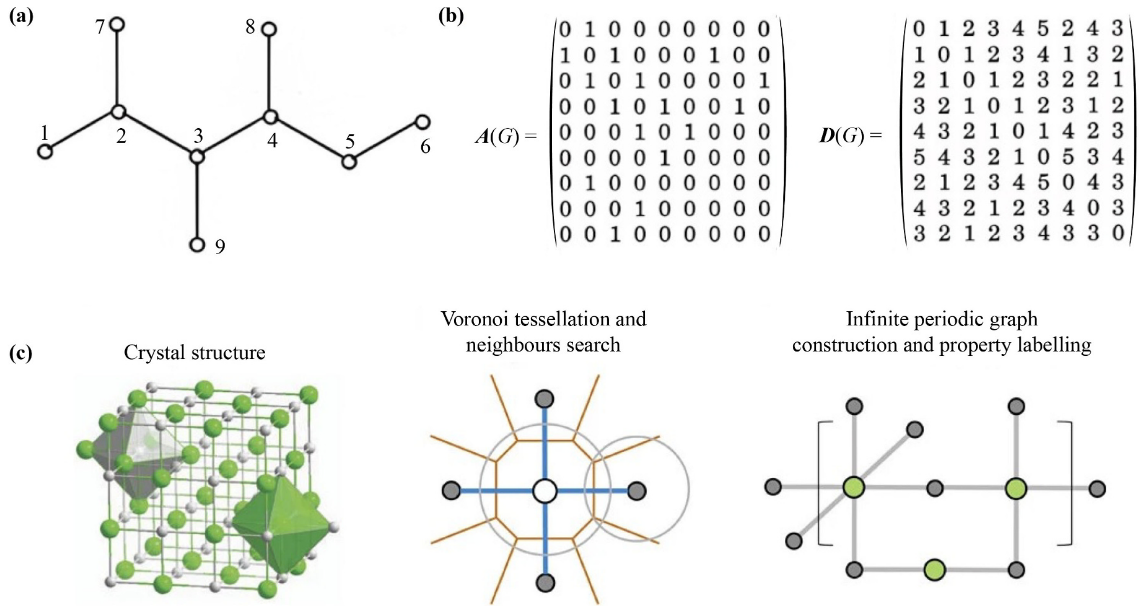
where the $\boldsymbol{q}^{(k)}$ is the training set configurations. The SOAP descriptor is now widely adopted, especially in the machine-learning of potentials [227–230].

Based on the SOAP approach, Artrith *et al.* [216] introduced another descriptor for machine-learnt potentials, which does not scale with the number of chemical species, a feature that SOAP lacks. This is carried out by taking the union of a set of invariant coordinates which maps the atomic structure and another one that maps the chemical composition, which are both described by the radial and angular distribution functions:

$$\mathrm{RDF}_i\left(r\right) = \sum_{\alpha} c_a^{(2)} \phi_\alpha\left(r\right), \ 0 \leq r \leq R_c, \tag{6.11}$$

$$\mathrm{ADF}_i\left(r\right) = \sum_{\alpha} c_a^{(3)} \phi_\alpha\left(\theta\right), \ 0 \leq \theta \leq \pi, \tag{6.12}$$

where R_c is the cutoff radius and the ϕ_α is a complete basis set, which in their work is the Chebyshev polynomials


(a)
7
8
1
2
3
4
5
6
9
(b)
(c)
Crystal structure
Voronoi tessellation and neighbours search
Infinite periodic graph construction and property labelling

$$A(G)=\begin{pmatrix}0&1&0&0&0&0&0&0&0\\1&0&1&0&0&0&1&0&0\\0&1&0&1&0&0&0&0&1\\0&0&1&0&1&0&0&1&0\\0&0&0&1&0&1&0&0&0\\0&0&0&0&1&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\end{pmatrix}\qquad D(G)=\begin{pmatrix}0&1&2&3&4&5&2&4&3\\1&0&1&2&3&4&1&3&2\\2&1&0&1&2&3&2&2&1\\3&2&1&0&1&2&3&1&2\\4&3&2&1&0&1&4&2&3\\5&4&3&2&1&0&5&3&4\\2&1&2&3&4&5&0&4&3\\4&3&2&1&2&3&4&0&3\\3&2&1&2&3&4&3&3&0\end{pmatrix}$$

Fig. 9 **(a)** Structure graph for 2,3,4-trimethylhexane and **(b)** the related adjacency and distance matrix. Reproduced with permission from Ref. [231]. **(c)** The Universal fragment descriptors. The crystal structure is analysed for atomic neighbours via Voronoi tessellation with the infinite periodicity taken into account. Reproduced with permission from Ref. [232].

of the first kind.

6.3 Graph-based descriptor

By converting the atoms and bonds in a molecule into vertices and edges, we can turn the molecule into a graph as depicted in Fig. 9(a). The information about the edges and the edge distance between vertices can then be encoded into the adjacency and distance matrices [231], shown in Fig. 9(b). This graph-theoretic approach is known as structure graph, which has been devised long ago in 1863. Despite the simplicity and apparent loss of 3D information, structure graphs have seen widespread uses in comparing the structure of molecules.

The generalization of structure graph to periodic systems is the Universal Fragment Descriptor (UFD) [232], which uses the Voronoi tessellation [see Fig. 9(c)] to determine the connectivity of atoms, in the following two steps:

i) The crystal is partitioned into atom-centered Vornoi–Dirichlet polyhedral;

ii) Atoms that share a perpendicular-bisecting Voronoi face with interatomic distance smaller than the Cordero covalent radii (with 0.25 Å tolerance) is determined to be connected. Periodic atoms are considered.

which defines the graph. Subgraphs are also constructed corresponding to the individual fragments, which include linear paths connecting at most 4 atoms and circular fragments, representing the coordination polyhedral of an atom. Then, an adjacency matrix $\boldsymbol{A}$ is constructed based on the determined connectivity, along with a reciprocal distance matrix ($D_{ij}=1/r_{ij}^2$), which when multiplied together gives the Galvez matrix $\boldsymbol{M}\equiv\boldsymbol{A}\cdot\boldsymbol{D}$. The information about the atomic/elemental reference property $\boldsymbol{q}$ (could be Mendeleev group and period number, number of valence electron, electronic affinity, covalent radii, etc.) is then incorporated in the pair of descriptors for a particular property $\boldsymbol{q}$,

$$T^E=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}|q_i-q_j|\,M_{ij},\tag{6.13}$$

$$T^E_{bond}=\sum_{\{i,j\}\in\text{bonds}}|q_i-q_j|\,M_{ij},\tag{6.14}$$

where the former runs over all pairs of atoms while the latter only considers bonded pairs of atoms.

Xie *et al.* [233] proposed a framework, the generalized crystal graph convolutional neural networks (CGCNN) which introduced another graph-based descriptor that is inspired by the UFD. Their construction of the crystal graph is illustrated in Fig. 10, where the connectivity determination is the same as in UFD, but they used the one-hot encoding to encode the atom and bond properties in two separate feature vectors: node feature vectors and edge feature vectors. They are the descriptors, which are then sent through convolutional layer, which further extracts critical features while reducing the dimensions. Convolutional neural network is discussed in Section 7.

6.4 Topological descriptor

Topology famously does not differentiate between a donut and a coffee mug, as they both have a hole. This

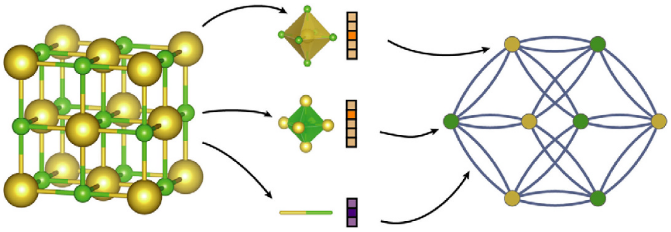


Fig. 10 Crystal graph construction proposed used in the generalized crystal graph convolutional neural networks. Reproduced with permission from Ref. [233].

is because in topology, most of the geometrical features are stripped away, leaving only quantities that are invariant under continuous deformation. This seemingly bizarre concept has had deep impacts in physics and was the main theme of the works that won the 2014 Nobel Prize. A branch of topology, persistence homology, measures the topological features which persist across different scales or granularity, and encodes them into diagrams. This idea has already been used to classify and describe proteins [234, 235], and used as ML descriptor for crystalline [236] and armophous solids [237]. However, it has not been widely used due to its mathematically complicated nature, and the lack of physical and chemical intuition also further hinders the ability to interpret the results [216]. Here, we will attempt to provide a simplified and non-rigorous overview of persistent homology and the crystal topological descriptor of Jiang *et al.* [236]. For a rigorous and detailed introduction to persistent homology, Ref. [238] is recommended along with other works cited in this paragraph, especially Ref. [235].

The basic building blocks of persistent homology are the *simplices* (see Fig. 11): a 0-simplex is a point, a 1-simplex is two connected points, a 2-simplex is a filled triangle, and a 3-simplex is a filled tetrahedron. A *face* can be a point, a line, or a 2D surface, depending on the number of points. A *simplicial complex* K is a collection of simplices which satisfies two conditions:

i) Faces of a simplex in K are also in K.

ii) Any intersection of two simplices in K is a face of both the simplices.

In a simplicial complex, holes are considered as voids that are bounded by simplices of different dimensions. In dimension 0, a connected component is counted as a hole; in dimension 1, a hole is a loop bounded by 1-simplices or edges; in dimension 2, hole is bounded by 2-simplices or triangles. The number of i-dimensional holes or voids in a simplicial complex is basically described by the i-th Betti numbers, $\beta_i(K)$, e.g., β_0 is the number of connected components, β_1 is the number of loops, and β_2 is the number of cavities. An example is shown in Fig. 11(b), where there are five 0-simplices or vertices, forming six 1-simplices, and one 2-simplices. Since all the points are connected, $\beta_0 = 1$; there is a square-shaped hole enclosed by the 1-simplices A, B, C, and F, giving $\beta_1 = 1$. Note that the face of the triangle is filled, thus it is not counted as a hole.

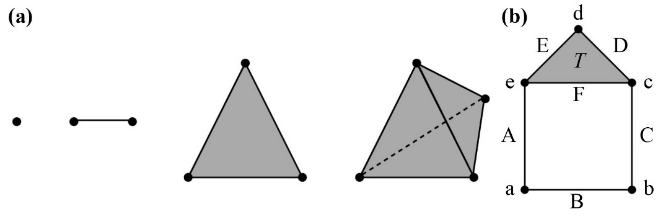


Fig. 11 **(a)** Left to right: 0-, 1-, 2-, 3-simplex. **(b)** An example of a simplicial complex, with five vertices: a, b, c, d, and e, six 1-simplices: A, B, C, D, E, and F, and one 2-simplex T. The Betti numbers for this complex are $\beta_0 = \beta_1 = 1$. Reproduced with permission from Ref. [238].

To generate simplicial complex from the crystal data, we use the Vietoris–Rips (VR) filtration process, giving VR complex. This is carried out by increasing a filtration parameter, commonly the Euclidean distance cutoff between points, where points that are within cutoff distance of each other are connected. The filtration parameter used in the crystal descriptor is the radius measured from each atom, d, which is increased from 0 to 8 Å. As d increases, the simplicial complex also undergoes changes, where the Betti numbers of “holes” change. This can be quantitatively plotted using persistence barcodes for each of the Betti numbers, as can be seen in Fig. 12, where each barcode represents each of the “hole” for each Betti number. As d reaches 4 Å, all the Betti 0 barcodes except one suddenly terminate, indicating that the points are now all connected as the Na atoms are separated by 4 Å. There is no Betti 1 barcode because the distances between any two Na atoms are the same, reflecting the structure symmetry.

To embed different elemental compositions, atom-wise chemical information is used, where a chosen atom is surrounded by atoms of a chosen type, such as in Figs. 12(a) and (b), where the selected Na atom is surrounded by only Na atoms and Cl atoms, respectively. The birth, death, and the persistent length of the Betti barcodes are then encoded in a vector known as ASPH feature vector.

6.5 Reciprocal space-based descriptor

The reciprocal space is linked to the real space by the means of Fourier transform and can be mapped using X-ray diffraction (XRD) into 2D diffraction pattern [see Fig. 13(a)], either experimentally or computationally. 2D XRD data has first been applied as descriptor by Ziletti *et al.* [239] to automatically classify crystal structures. In their work, they rotated the crystal 45° clockwise and counterclockwise about a chosen axis and superimposed the obtained XRD patterns. This is then carried out for the other two axes, with different colours of the RGB

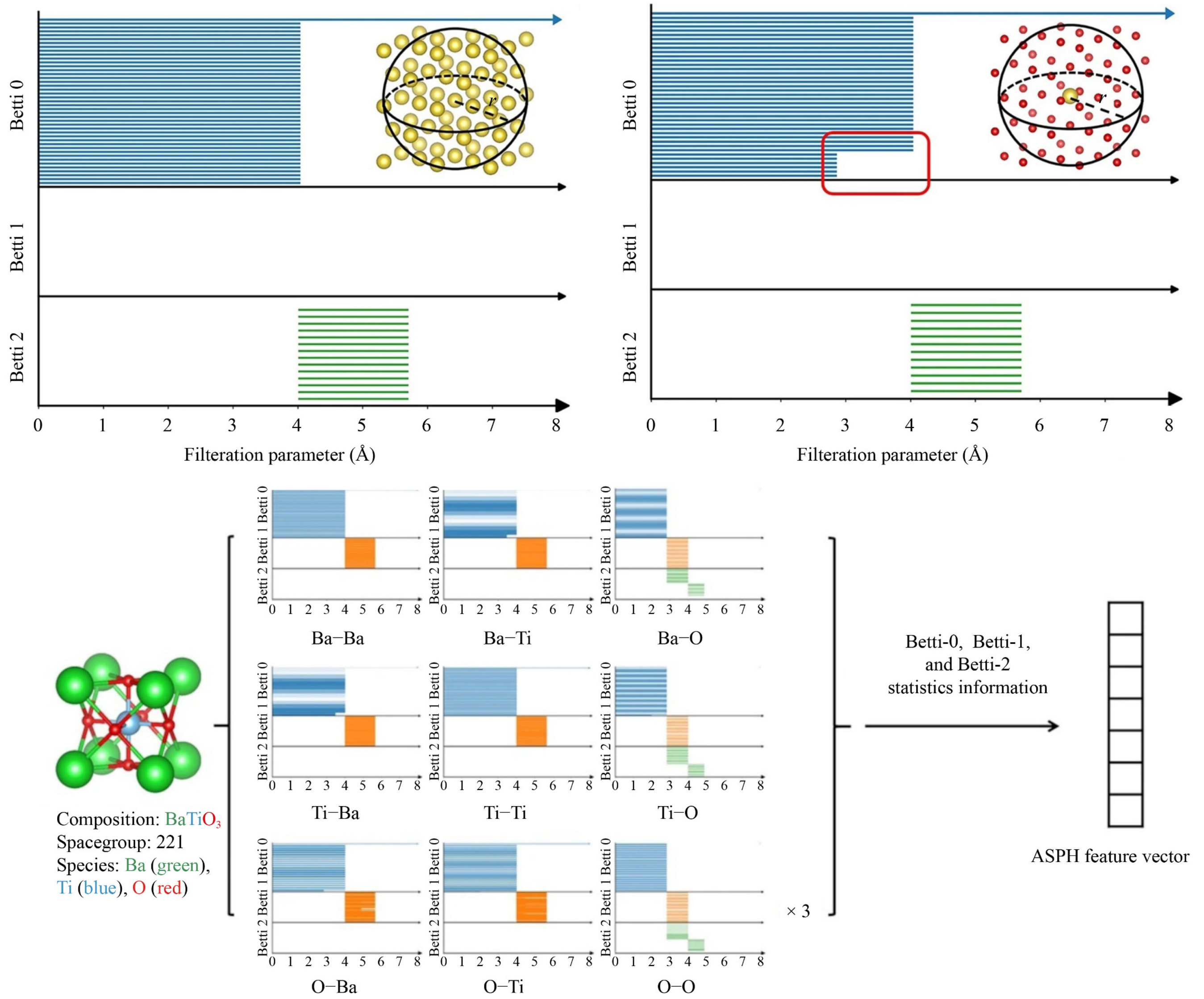


Fig. 12 Persistence barcode plot for the selected Na atom inside a NaCl crystal, surrounded by only **(a)** Na atoms and **(b)** Cl atoms. **(c)** Construction of crystal topological descriptor, taking into account different chemical environmen *t*. Reproduced with permission from Ref. [236].

palette chosen for the patterns obtained from the rotation of different axis [e.g., red for x-axis, green for y-axis, and blue for z-axis, see Fig. 13(b)]. The final obtained pattern is then used as the descriptor, fed into a convolutional network, similar to image-based object recognition. The benefits of the XRD descriptor are that the dimension is independent of the system size and very robust to defects [compare Fig. 13(b) and Fig. 13(c)].

The more conventional XRD is the 1D XRD, shown in Fig. 13(d) for different crystals, which is obtained based on Bragg's law, mapping the 3D structures into 1D fingerprints. 1D XRD based descriptor has been used to classify crystal structure [240] and predict their properties [241]. In the latter, the group used modified XRD, where only the anions sublattice is considerd with the cations removed, and the *pymatgen* package is used to generate the XRD computationally. They successfully distinguished solid-state lithium-ion conductors with this descriptor using unsupervised learning.

6.6 Reduction of descriptor dimension

In materials science, there are many possible combinations of various properties that can be used as descriptors. It is often difficult to select and fine-tune the descriptor space manually. This is a common problem in the field of ML, and several methods have been developed to tackle this issue: principal component analysis (PCA) [242, 243], least absolute shrinkage and selection operator (LASSO) [244], and sure independence screening and sparsifying operator (SISSO) [245]. However, they mainly work for models that are linear, hence not

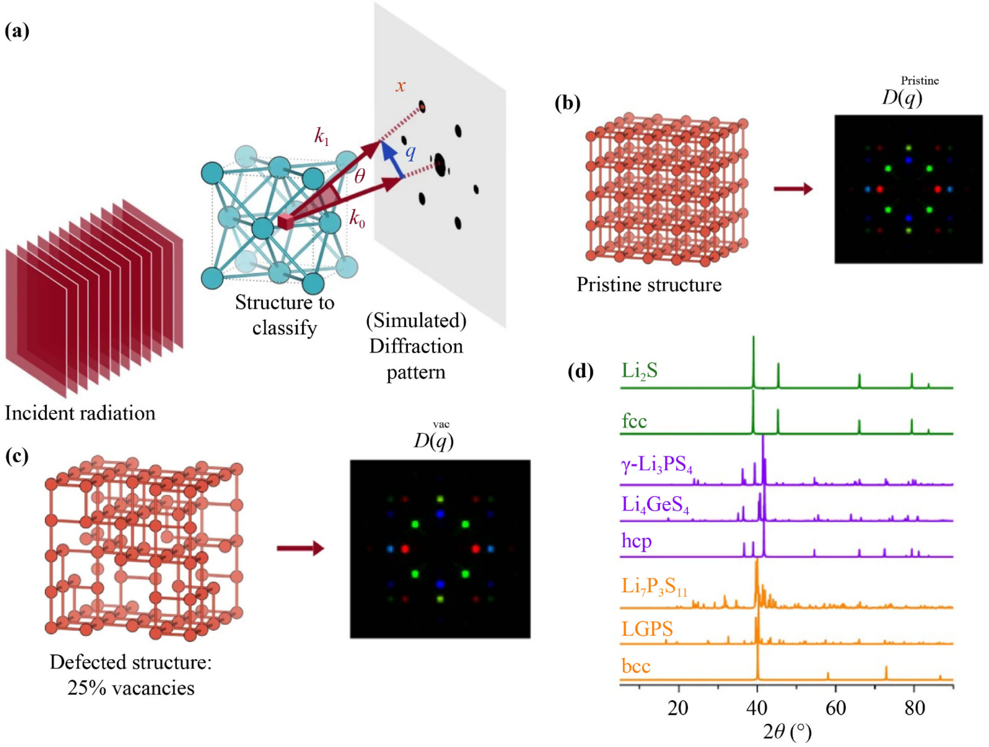


Fig. 13 **(a)** Experimental XRD method, where X-ray plane wave incidents on a crystal, resulting in diffraction fingerprints. **(b, c)** XRD-based image descriptor for a crystal where each RGB colour corresponds to rotation about the x, y, z axes. The robustness of the descriptor against defects can be observed by comparing (b) to (c). **(d)** Examples of 1D XRD. (a–c) Reproduced with permission from Ref. [239], (d) Reproduced with permission from Ref. [241].

Table 6 List of Machine Learning (ML) algorithms used by various tools or framework developed in materials science.

ML algorithms	Tool
Support vector machine (SVM)	Refs. [260, 261, 262, 246]
Kernel ridge regression (KRR)	Refs. [237, 263, 247, 264]
Deep neural network	VampNet [257], DTNN [265], ElemNet [266], IrNet [267], PhysNet [268], DeepMolNet [269], SIPFENN [270], SpookyNet [250]
Convolutional neural network (CNN)	SchNet [271], Refs. [239, 240, 272, 273]
Graph neural network (GNN)	CGCNN [274], MEGNet [275], GATGNN [276], OrbNet [277], DimeNET [278], ALIGNN [279], MXMNet [280], GraphVAMPNet [281], GdyNets [282], NequIP [283], PaiNN [284], CCCGN [285, 286], FFiNet [287]
Generative adversarial networks (GAN)	Ref. [288], CrystalGAN [246], MatGAN [289]
Variational auto encoder (VAE)	FTCP [290], CDVAE [291], Refs. [292, 263]
Random forest/ decision tree	Refs. [236, 293, 294, 251, 295, 296]
Unsupervised clustering	Refs. [241, 282, 252, 297, 298]
Transfer learning	Roost [299], AtomSets [288], XenonPy.MDL [289], TDL [290], Refs. [256, 291, 292, 300, 301]

directly applicable for neural network-based models [239].

7 Machine learning algorithms for material science

In this section, we collate the recently developed ML-based tools and frameworks in materials science, grouping them together by the ML algorithms used. We then briefly describe the commonly used algorithms and also introduces some of the emerging algorithms, which could further unlock the potential of materials science ML applications.

7.1 Currently utilized algorithms

Table 6 enumerates the ML algorithms used in relatively

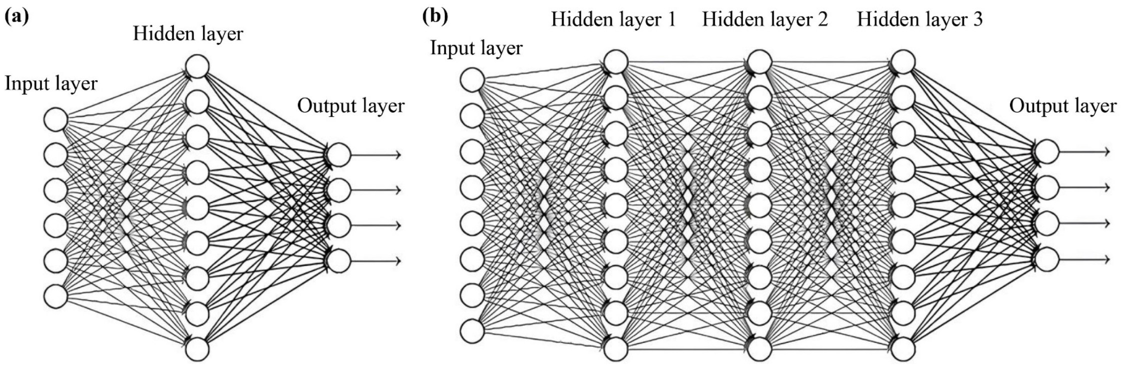


Fig. 14 **(a)** Neural network (NN) with 3 layers: input, hidden, and output. **(b)** Deep NN with 3 hidden layers. Reproduced with permission from Ref. [257].

recent developed tools or framework in materials science. It can be seen that the convolutional and graph neural networks are the popular algorithms, with transfer learning also picking up pace. We will briefly introduce the algorithms, drawing examples from the materials science implementation.

7.1.1 Kernel-based linear algorithms

Support vector machine (SVM) and kernel ridge regression (KRR) are kernel-based ML algorithms, which utilize kernel functions $K(\boldsymbol{x},\boldsymbol{x}')$ that allow high-dimensional feature to be used implicitly, without actually computing the feature coordinates explicitly, hence speeding up computation. Furthermore, it allows non-linear problem to be solved using linear algorithms by mapping the problem into a higher-dimensional space. Examples of commonly used kernel functions include

Linear kernel:

$$K\left(\boldsymbol{x},\boldsymbol{x}'\right)=\left(\boldsymbol{x_i}\cdot\boldsymbol{x_j}+\theta\right); \tag{7.1}$$

Polynomial kernel:

$$K\left(\boldsymbol{x},\boldsymbol{x}'\right)=\left(\boldsymbol{x_i}\cdot\boldsymbol{x_j}+\theta\right)^d; \tag{7.2}$$

Gaussian kernel/radial basis function (RBF):

$$K\left(\boldsymbol{x},\boldsymbol{x}'\right)=\exp\left(\frac{-||x_i-x_j||^2}{\sigma^2}\right), \tag{7.3}$$

where $\boldsymbol{x}$ is the input data, while θ and σ are adjustable parameters. SVM is used for both classification and regression problem, denoted as SVC and SVR, respectively. On the other hand, KKR is used only for regression problems and it is very similar to SVR, except for the different loss functions.

Applications of both types of SVM are demonstrated in the work of Lu *et al.* [246]. Using atomic parameters such as electronegativity, atomic radius, atomic mass, valence, and functions of these parameters, they constructed classifier for the formability of perovskite structure, and regression models to predict the band gaps of binary compounds, using the polynomial kernel [Eq. (7.2)] with $d=2$ and $\theta=1$.

Wu *et al.* [247] used KRR to assist in non-adiabatic molecular dynamics (NA-MD) simulations, particularly in the prediction of excitation energy and interpolate nonadiabatic coupling. KRR was chosen over neural networks because of the fewer hyperparameters KRR possessed, and KRR requires only the use of simple matrix operation to find the global minimum. By only providing a small fraction (4%) of sampled points, KKR gives a reliable estimate while saving MD computational effort of over an order of magnitude.

7.1.2 Neural network

Artificial Neural Networks (ANNs, shortened to NNs) is a type of ML architecture that aims to mimic the neural structure of the brain. In NNs, there are 3 types of layers consists of interconnected nodes: input layer, hidden layer(s), and output layer, as shown in Fig. 14(a). The input layer receives the input raw data, which is then propagated to the hidden layer(s), where the nodes are functions of the backward-connected nodes and each connection is weighted. The function of a hidden layer node m with x being the node values of previous layer, takes the form:

$$h_m=\sigma\left(b+\sum_i^n\omega_i\cdot x_i\right), \tag{7.4}$$

where $\sigma\left(z\right)$ is known as the activation function, where a common choice is the sigmoid function $\sigma\left(z\right)=1/(1+\mathrm{e}^{-z})$ and b is bias term, and the ReLU (Rectified Linear Unit) function, simply defined as $\sigma\left(z\right)=\max(0,z)$. After the hidden layer(s), the information is then passed toward to the output layer, which is another function of the nodes of the final hidden layer. The outputs are measured against true value using a pre-defined cost function, with the simplest example for regression problem being the sum of squared error

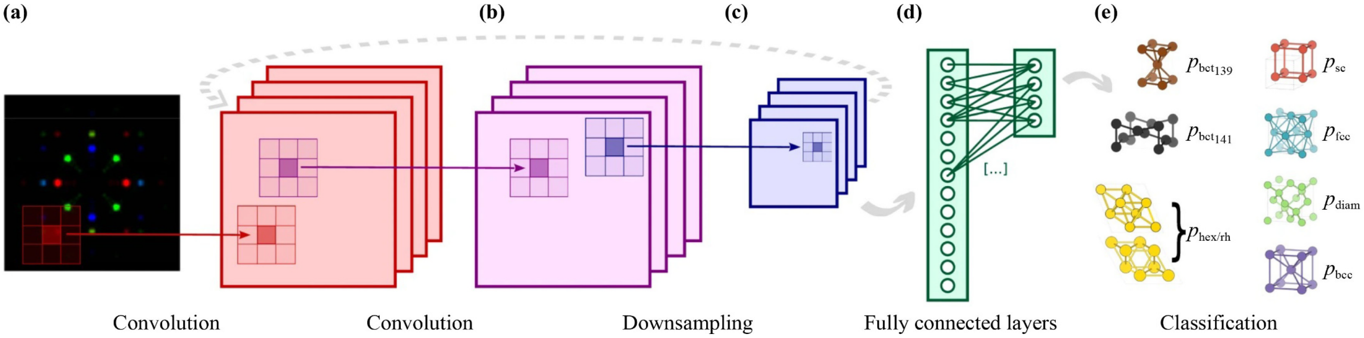


Fig. 15 The CNN architecture used in the work of Ziletti *et al.* [239]. **(a)** A kernel or learnable filter is applied all over the image, taking scalar product between the filter and the image data at every point, resulting in an activation map. This process is repeated in **(b)**, which is then coarse grained in **(c)**, reducing the dimension. The map is then transferred to regular NNs hidden layers **(d)** before it is used to classify the crystal structure **(e)**. Reproduced with permission from Ref. [239].

$$J\left(\omega\right)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(y_i-\widehat{y}_i\right)^2,$$

where y_i and $\widehat{y}_i$ are the true and predicted values respectively, and the sum is taken over the whole training set with n being the size of the training set. The weights are then optimized iteratively using the backpropagation method, which is a function of the gradient of the cost function. For a detailed discussion, the book [248] is recommended.

Deep NNs are NNs with more than one hidden layer [see Fig. 14(b)]. By having more hidden layers, the model is better positioned to capture the nonlinearities in the data. However, having too many hidden layers can cause the convergence or learning to be slow and difficult, because the gradients used in backpropagation will tend to become vanishingly small. To overcome this issue, residual block has been devised [249], which introduces shortcut between layers.

SpookyNet [250] is a DNN-based model built to construct force fields that explicitly include nonlocal effects. In their DNN architecture, the generalized sigmoid linear unit (SiLU) activation function is used, which is given by $\mathrm{silu}\left(x\right)=\frac{\alpha x}{1+\mathrm{e}^{-\beta x}}$, where both α and β are learneable parameters. They noted that a smooth activation function is crucial for the prediction of potential energies, as discontinuities in the atomic forces would be introduced otherwise. They introduced a loss function that has 3 components: energy, forces, and dipole moments, which is minimized by optimizing the weights using mini-batch gradient descent. They also incorporated residual block which allowed them to use a large number of hidden layers.

Convolutional NNs (CNNs) is primarily used in image pattern recognition, and is different from deep NNs by having a few extra layers, which are the convolutional and pooling layers. The extra layers filter and convolute the data to capture crucial features in the data and also reduce the input dimension, which scales quickly with resolution in image recognition problems. The work of Ziletti *et al.* [239] used CNN architecture, as depicted in Fig. 15. The convolution layers capture elements that are discriminative and discard unimportant details.

Graph NNs is specifically designed for input data that are structured as graph, which contains nodes and edges, and can handle inputs of different sizes. There are several differrent types of graphs NNs, such as graph convolutional network (GCNNs), graph attention network, and Message Passing Neural Network.

7.1.3 Decision tree and ensembles

Decision tree is a supervised method for solving both classification and regression problems, which resembles a tree. A typical decision tree is shown in Fig. 16(a), where each internal node represents a feature or attribute, each branch contains a decision rule, and each leaf node is a class label or a numerical value, depending on the type of problem solved. The number of node layers a decision tree contains is known as depth, which needs to be tuned. An important metric used in measuring the performance of a decision tree in classification is the information gain, which is defined as the difference of the information entropy between the parent and child node; while for regression problem, the variance reduction is the performance evaluation metric for a decision tree. Decision tree is advantageous when it comes to interpretability, but it suffers from overfitting, especially when the tree is too deep and complex. It can also be overly-sensitive to data changes.

Random forest is an algorithm that combines multiple decision tree, with each of them trained on randomized subsets of samples, where both training data and features are chosen random with replacement in a process known as bootstrapping. The final decision is then made by aggregating the results from each decision tree and taking the majority vote for classification or the average for regression. The steps taken above are collectively known as bagging, which help ensure that the random forest algorithm is less sensitive to changes in

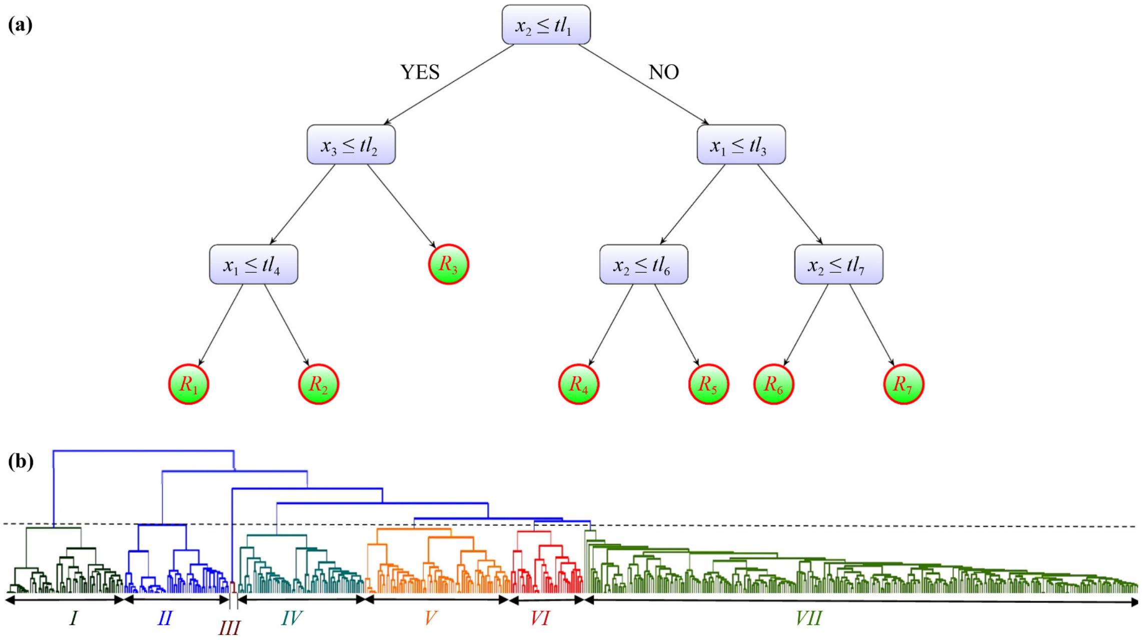


Fig. 16 **(a)** An example of a decision tree, where each square represents internal node or feature, each arrow represents branch or decision rule, and the green circles are leafs representing class labels or numerical values. **(b)** Dendogram obtained via agglomerative hierarchical clustering (AHC) where the dashed line indicates the optimal clustering. (a) Reproduced with permission from Ref. [258], (b) Reproduced with permission from Ref. [241].

the dataset and more robust to overfitting.

Gradient Boosting Decision Tree (GDBT) is another method that uses ensembles of decision trees but in sequence rather than in parallel. GBDT works by adding decision trees iteratively, with each one attempts to improve upon the errors of the previous tree. The final output from the trees ensembles is then taken by using weighted average of the decision trees outputs.

Random forest models are used in the work of Zheng *et al.* [251], which predicts the atomic environment labels from the X-ray absorption near-edge structure (XANES). Using the random forest classifier of scikit-learn package, they found that 50 trees ensemble gave the best performance, even better than other classifiers, such as CNN and SVC. On the other hand, GBDT has been used for regression in the topology-based formation energy predictor [236]. Also using the scikit-learn package, they added a tree to their model one at a time and used bootstrapping to reduce overfitting. This topology-based model is able to achieve a high accuracy in cross-validation, with mean absolute error of only 61 meV/atom, outperforming previous works that uses Voronoi tessellations and Coulomb matrix method.

7.1.4 Unsupervised clustering

K-Means clustering is a popular unsupervised classification algorithm which aims to group similar data points together in K different clusters. K numbers of points that are known as cluster centroids are initialized randomly, and each data point is assigned to a cluster centroid that is closest in Euclidean distance to the data point. The centroids are then moved to a new location that is the arithmetic mean of the assigned data points. This repeats until convergence, i.e., there is no more movement among the centroids. The number K determines the number of classes in the data, which can be known before hand if the dataset has clear distinction, e.g., metal vs. non-metal, or can be optimized using the elbow method, which has an associated cost function J that is optimized by the best choice of K. Despite its popularity, K-Means clustering has some limitations, such as sensitivity to outliers, dependence on the centroids position initialization, ineffective for dataset with uneven distribution, and predetermined number of clusters.

Several alternatives have been: proposed which improves upon the limitations of K-Means clustering. Agglomerative hierarchical clustering (AHC), used in the work of Zhang *et al.* [241], is initialized by using each data point as a single cluster, then iteratively merged the clusters of the closest points until one big cluster is left. Then, a dendogram or a bottom-up hierarchical tree diagram, as show in Fig. 16(b), which can be cut at a desired precision, as indicated in the figure via a dashed line, where 7 groups are obtained. To verify the results, they performed spectral clustering, which splits the samples into chosen K groups, based on the eigenvalues of the similarity matrix constructed from the data. This process is recursively applied bisectionally,

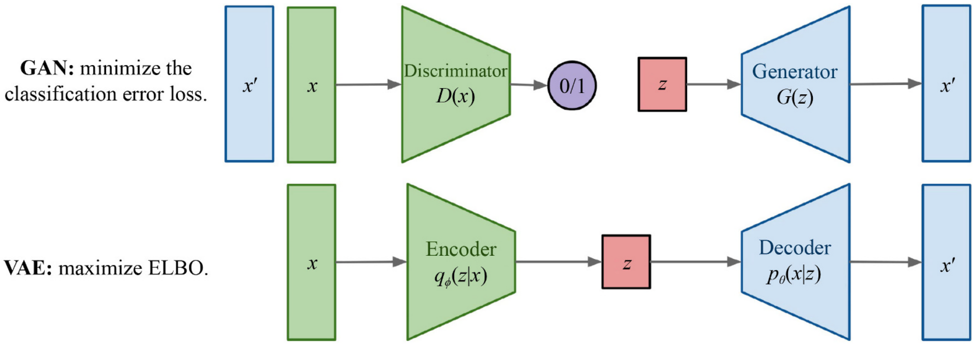


Fig. 17 The architectures of the two generative models, Generative adversarial networks (GAN) and Variational auto encoders (VAE). Reproduced with permission from Ref. [259].

and they obtained similar clusters as the AHC. There is also the mean-shift algorithm, utilized in Ref. [252].

7.1.5 Generative models (GAN and VAE)

Generative models attempt to learn the underlying distribution of a training dataset, and use that to generate new samples that resemble the original data. Two popular types of generative models are Generative Adversarial Networks (GAN) and Variational Autoencoders (VAE). As can be seen from Fig. 17, there are two different neural networks in both of the models: GAN contains a discriminator and generator network, while VAE has a decoder and encoder network. In GAN, random noise is injected into the generator network and subsequently outputs a sample that is then fed to the discriminator network, which is then classified as real or generated sample. The networks are trained together until the generated samples are able to convince the discriminator that the samples are real and not generated. On the other hand, VAE tries to learn the latent representations from the training data and generate new samples based on them using probabilistic approach.

A variant of GAN, Wasserstein GAN, has been applied in the work of Kim *et al.* [253], which generate Mg–Mn–O ternary materials which can potentially be used as potential photoanode materials. The overview of their GAN architecture is shown in Fig. 18(a), which after training, takes in random Gaussian noise vector Z and encoded composition vector, and spits out new unseen crystals. The new crystals are then passed to a critic and a classifier, where the former computes the Wasserstein distance that measures the dissimilarity between the generated and true data distributions, which are used to improve the realism of the generated materials, while the latter ensure that the generated materials meet the composition condition. Using this model, they found 23 previously unknown new crystals with suitable stability and band gap.

An example of inverse design using VAE was demonstrated in the work of Noh *et al.* [254], where their proposed a two-step VAE-based generator is shown in Fig. 18(b). In the first step, the materials data is passed to a convolutional autoencoder, which contains 4 convolutional layers, outputting a compressed intermediate vector, which is then fed to a decoder that aims to maps the vector back to the input. The intermediate vector is fed into the VAE in 2nd step to learn about the latent materials space. To generate completely novel polymorphs, the materials space around known stable structure is sampled using random Gaussian distributed vectors and the resulting latent vectors are decoded in a series of steps to output new stable structures. The model is able to recover 25 out of 31 known structures that are not included in the training, and 47 new valid compositions are discovered that have eluded genetic algorithms.

7.1.6 Transfer learning

In materials science, high quality data for a specific type of materials is usually scarce, which severely impedes the applications of ML in generating high quality predictions [255]. Transfer learning (TL) is a method that can be applied to overcome this data scarcity issue. In transfer learning, the parameters of a model that has been pre-trained on a large data set but with different task/purpose, are used to initialize training on another data-scarce task, such as the parameters of the models used for predicting formation energy is later used to train another task of predicting band gaps.

Chang *et al.* [256] combined pairwise transfer learning and mixture of experts (MoEs) framework in their model. In pairwise transfer learning, a model is pre-trained on a source task (task designed for the large dataset) and a subset of the pre-trained model parameters is used to produce generalizable features of an atomic structure, defined as a feature extractor. This extractor extracts a feature vector from an atomic structure, which can be used to predict a scalar property after passing thorugh a neural network. On the other hand, MoE contains multiple neural network models that specialize in different regions of the input space, known as "experts", and each

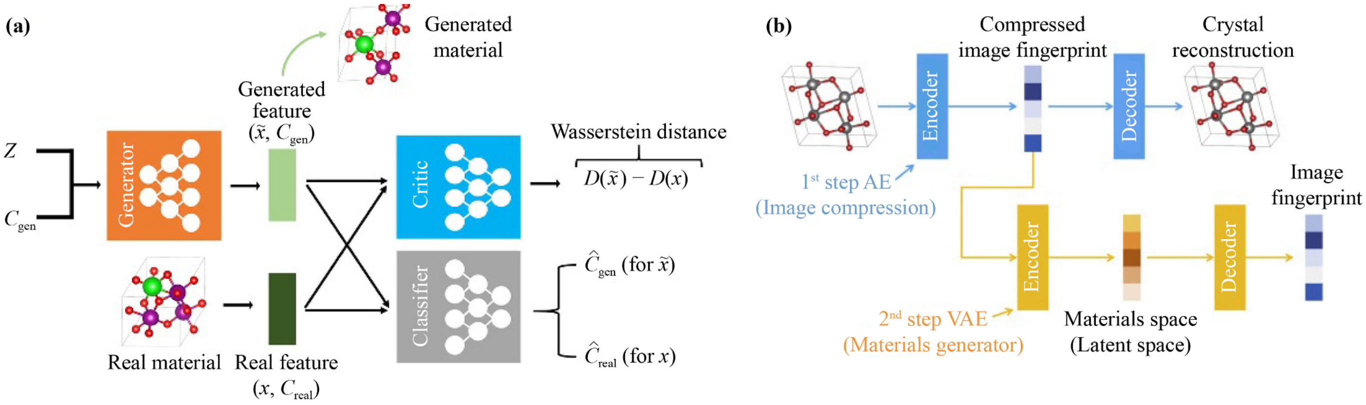


Fig. 18 **(a)** Composition-conditioned crystal GAN, designed to generate crystals that can be applied in photodiode. **(b)** Simplified VAE architecture used in the inverse design of V_xO_y materials. (a) Reproduced with permission from Ref. [253], (b) Reproduced with permission from Ref. [254].

of them are activated and controlled by a gating function. The outputs of the "experts" are then aggregated through an aggregation function. Using this architecture, the authors have performed many downstream, data-scarce tasks, such as predicting band gap, poisson ratio, 2D materials exfoliation energy, and experimental formation energies.

7.2 Emerging ML methods

7.2.1 Explainable AI (XAI) methods

The DNNs-based approaches discussed have proved to be of great help in assisting and speeding up materials research, but their black-box nature has made understanding and explaining the results difficult, which has also plagued the general ML community [302]. In systems that trust, fairness, and moral are highly critical, such as in healthcare, finance, and autonomous driving, the decisions made by AI cannot be blindly trusted without understanding the motivation and reasoning behind the choice. Furthermore, when the black box returns results that are erroneous and puzzling, it can be difficult to diagnose and correct without knowing what exactly went wrong. To overcome these issues, the XAI techniques were introduced, which try to explain the reasonings and connections behind a prediction or classification.

There are many post-hoc (i.e., applied after model fitting) XAI methods proposed for the general ML community [303], including gradient-based attribution (Gradients, Integrated gradients, and DeepLIFT), deconvolution-based methods (Guided backpropagation, Deconvolution, Class activation maps (CAM), Grad-CAM), model-agnostic techniques (Shapley additive explanations (SHAP), local interpretable model-agnostic explanations (LIME), Ancors).

Another type of XAI is the use of models that are inherently interpretable or explainable, which have one or more of these features [304]: sparsity, simulatability, and modularity. A model that has limited number of nonzero parameters is known as sparse, and this can be obtained by the LASSO method, whereas if a model can be easily comprehend and mentally simulate by the human user is simulatable, such as decision trees-based model. A modular model is a model that combines several modules which can be interpreted independently. In the field of materials science, the understanding of the physical and chemical intuition is paramount as it opens the door to understanding hidden connection and physics, and improve the efficiency of future studies by providing insights from previous work. The importance and implementation of XAI in materials ML tools (refer to Table 6) have been discussed in the review of Oviedo *et al.* [58] and Zhong *et al.* [302]. Zhong *et al.* [302] presented an overview of DNNs-based XAI as shown in Fig. 19(a), which highlight two fundamental motivations for XAI, which is the need for explaining how the results are obtained from the input (model processing), and what information is contained in the network (model representations). The design of an intrinsically explainable DNNs will prove important in answering the questions posed, but is itself a highly difficult task. In the following, we will illustrate some of the materials science XAI implementation, which is still in its infancy.

Kondo *et al.* [305] used heat maps to highlight the feature importance, particularly in identifying the positive and negative features that affect ionic conductivity in ceramics, using scanning electron microscope (SEM) images. Their CNN-based model used feature visualization method that is very similar to the deconvolution method used in CAM and Grad-CAM. By defining mask map, they obtained masked SEM images [see Fig. 19(b)] that show features that determine low and high ionic conductivities.

A recent implementation of XAI for crystals is the CrysXPP [306] which is built upon an auto-encoder-based architure, CrysAE, that containing deep encoding

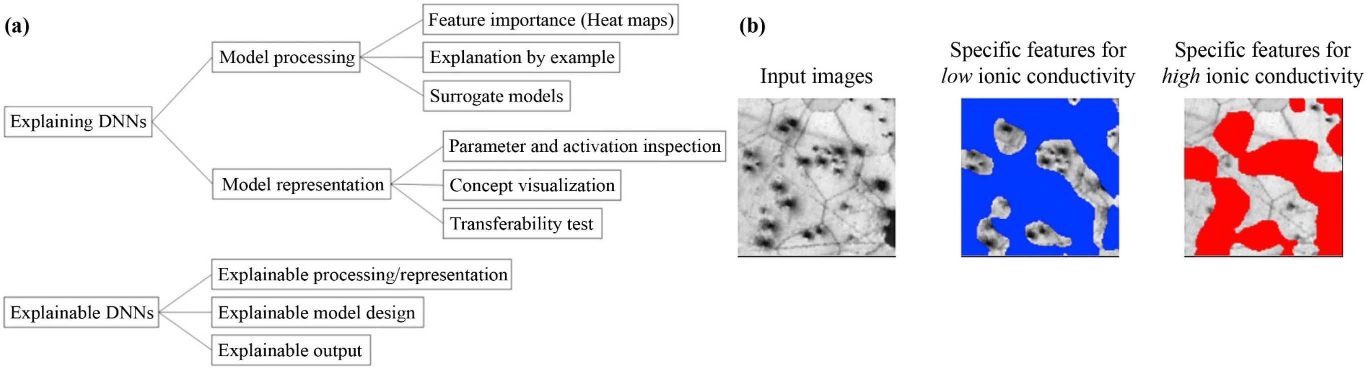


Fig. 19 **(a)** Overview of explainable DNNs approaches. **(b)** Feature visualization in the form of heat map used in determining the ionic conductivity from SEM images. (a) Reproduced with permission from Ref. [302], (b) Reproduced with permission from Ref. [305].

module which is capable of capturing the important structural and composition information in crystal graph. The information learnt is then transferred to the GCNN contained within CrysXPP [shown in Fig. 20(a)], which takes in feature selected from crystal graph. The feature selector contains trainable weights that selects weighted subset of important features, which is fine-tuned with LASSO to improve the sparsity of the features. An example of the explainable results obtained is shown in Fig. 20(b), where features that affect the band gap of GaP crystal are weighted and compared.

Compositionally restricted attention-based network (CrabNet) [307, 308] is an example of explainable DNN in materials science that is based on the Transformer-

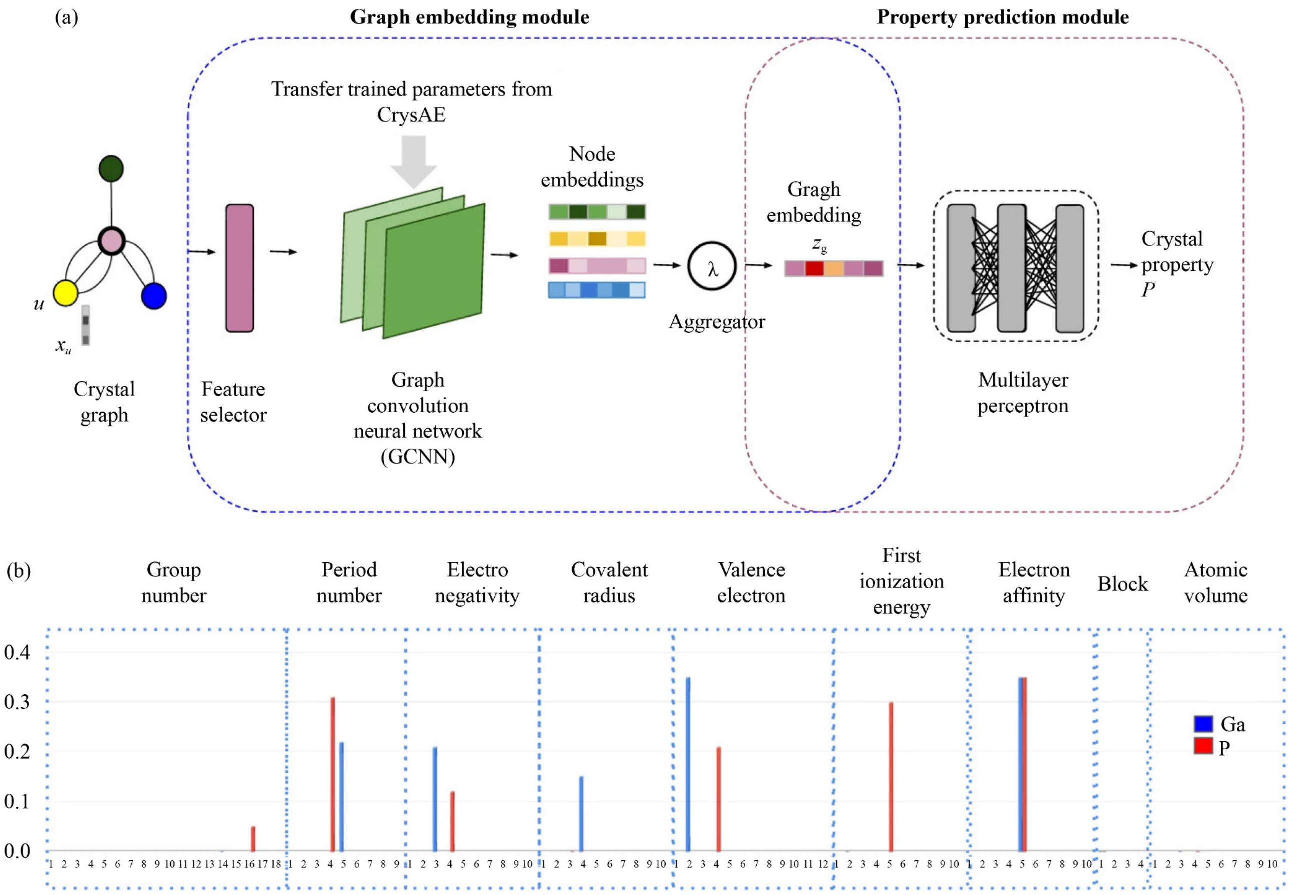


Fig. 20 **(a)** The architecture of CrysXPP, which is capable of producing explainable results, as seen in **(b)** the bar chart of features affecting the band gap of GaP crystal. Reproduced with permission from Ref. [306].

based self-attention mechanism [69], a type of algorithm initially intended for NLP, but has exploded in popularity recently. Briefly, the transformer self-attention mechanism allows the model to focus on the different parts of the input and relate them with weights to encode a representation. In this way, the dependencies between the elements are better captured, even when some of the elements are present in very small amount, e.g., dopants, which even in small quantity can have tremendous effect on the properties of the materials.

7.2.2 Few-shot learning (FSL)

As mentioned above, high-quality data in materials science which are complete with proper labels are scarce. This issue is exarcebated when we look at experimental data, which unlike computationally-produced data, is plagued with issues stemming from the different experimental equipments and variable environments. Therefore, the few-shot learning (FSL), which specifically targets situation where data is limited, has enticed materials scientists, especially the experimentalists. There are several approaches to FSL, as discussed in Refs. [309, 310], such as metric-based, optimization-based, and model-based approach. FSL is still a relatively young and unrefined method, but it has already attracted a lot of attentions. FSL has been implemented in the prediction of molecular properties [311, 312], classification of space group from electron backscatter diffraction (EBSD) data [313], and segmenting electron microscopy data [314].

8 Machine learning tasks for material science

This section will discuss the coverage of materials science tasks that ML tools have been utilized to assists in tackling. The common ML tasks in material science often coincide with the traditional ML tasks, which have been extensively studied and optimized. The tasks of inference of material property given structural and compositional data, generative modelling from a latent representation of desired properties, and the generation of DFT functionals, are analogous to the tasks that ML has traditionally performed well, including object classification, image and text generation using text cues, and natural language processing (NLP).

8.1 Potentials, functionals, and parameters generation

Traditionally, the XC functionals used in DFT are generated through mathematical approaches, guided by empirical data, such as the Perdew–Zunger exchange, with the exact XC functional remains elusive. The search for an improved XC functional above the currently popular GGA on the Jacob's ladder of Perdew [315] is desirable. The techniques of ML have been started to be utilized in the generation of new XC functionals [316–320], with the aim of improving the calculated accuracy while maintaining the efficiency. Transferability remains a huge challenge, which will need a huge and diverse dataset to achieve.

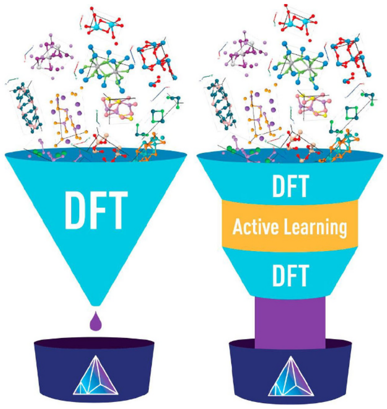


Fig. 21 High-throughput screening with learnt interatomic potential embedding from Ref. [330]. With the integration of active learning and DFT in the screening pipeline, the throughput efficiency or the quality of the output obtained from calculation can be improved. Reproduced with permission from Ref. [330].

The potentials and force fields used in molecular dynamics (MD) are critical in determining the reliability and accuracy of the output [321]. MD that does not involve first-principles approach but rather fixed potentials are in general less accurate the *ab initio* MD (AIMD), but they can be applied on a large system and long time scale, where AIMD is too costly. As such, one would hope that the standard MD can bring about results similar to AIMD. Developed in 2017, DeePMD [322] accurately reproduced the water model obtained from DFT. The same team developed an open-source tool for the on-the-fly generation of MD potentials, known as DP-GEN [323], available on available on URL: github.com/deepmodeling/dpgen. In 2020, the team won the ACM Gordon Bell Prize for the DeePMD work, as it can be scaled efficiently on the best HPC. Similar works have also been carried out by other teams [324, 325].

Another material modeling technique is the Density Functional Tight Binding (DFTB) method, which is less computational expensive than DFT-based first principles calculations. Efforts have been carried out on applying ML to obtain the TB parameters [252, 326].

8.2 Screening of materials

There are many methods to compress design space. To name few, one could train a model that predicts material property given material information or performs ML guided simulation of new materials to predict material

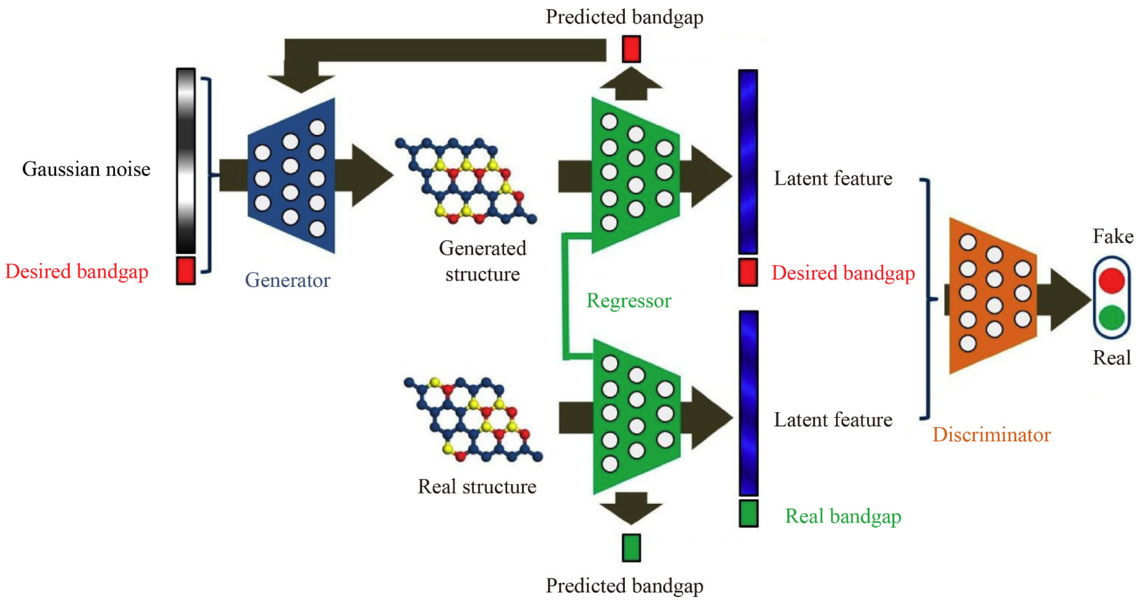


Fig. 22 Schematics of generative adversarial network. Reproduced with permission from Ref. [332].

behaviour under certain circumstances. High-throughput screening eliminates most potential materials without actually performing actual experiments to verify their property and provide experimentalists with a minimal set of candidate materials to try out. Pivoting to a generative model perspective, one could also specify material properties and generate stable materials that are likely to fit the specified property [327–329].

There are many properties that are of interests, including band gaps, bulk and shear moduli, crystal structures, conductivity, and topological states, as discussed in details in Ref. [216]. These properties are usually computed via DFT, which could be computationally expensive depending on the system setup. Properly setup and trained ML models can produce DFT-level accuracy properties predictions, while at far lower computational time. Isayev *et al.* [232] managed to obtain prediction at 0.1 s for each structure, which amounts to 28 million structure in a day, as pictured in Fig. 21. However, a well-trained and fully-transferable ML model requires the existence of high-quality large database and heavy computational power to optimize the model.

8.3 Novel material generation

The latent representations of common desired properties are of high interests among the community [331]. Based on the learnt latent representation, we can generate structure with similar desired properties at will. This is often carried out using the generative models, such as GAN and VAE, as demonstrated in the work of Dong *et al.* [332] and Pathak *et al.* [333].

8.4 Imaging data analysis

There are many imaging methods to capture the structure and fingerprints of a material experimentally at the atomic level, such as X-ray Diffraction (XRD), Fourier Transform Infrared Spectroscopy (FTIR), Atomic Force Microscopy (AFM), Transmission Electron Microscopy (TEM), and many others. Typically, they require laborious human interpretation to understand the meaning of the signals and whether they are due to noises and errors. This can be remedied with the help of ML, and thus far ML has been applied to:

- Identify symmetry and space group from XRD [334];
- Discover hidden atomic scale knowledge from XRD [297];
- Identify functional group in gaseous organic molecule [335];
- Analyze patterns and feature in AFM images, including domain wall and grain boundaries [336];
- Quantify nanoscale forces in dynamic AFM [337];
- Perform structural analysis and reconstruction in TEM [338];
- Identify chemistry and processing history from microstructure image [298];
- Characterize and analyze mechanical behaviour in microstructure image [339].

8.5 Natural language processing of material science literature

Nature language processing (NLP) refers to the ability of computer algorithm to understand spoken and written language. This technology has seen explosive development, with the recent GPT-3 [340] and now GPT-4 [341] models making strides not just academically, but used in just about everywhere. The trained AI models are able to hold realistic conversation with humans, take standardized exams [342], write codes in various programming language, and so on.

Most of the published results on materials science are not stored in a centralized database, which hinders the overall effort of ML applications. The NLP techniques can help this by scraping information from published literatures, such as the materials structure and properties. Tshitoyan *et al.* [343] demonstrated that an unsupervised learning modethat can capture complicated underlying knowledge from the literature and recommend materials for functional applications before their actual discovery. NLP also can help hypothesis formation and provide knowledge on the current trends in the field [344]. NLP methods can also serve as an efficient knowledge extractor from vast amount of material science literature, making the literature review process more efficient and thorough for researchers [345].

9 Perspectives on the integration of machine learning in materials science

In the following we will list perspectives on the integration of machine learning in materials science with materal science point of view and with machine learning point of view, respectively.

9.1 Perspectives from machine learning viewpoint

As ML techniques and ideals become ever more prevalent, we believe algorithmic templates and ML ideas will eventually become either the target modes of computation or the mode of guidelines which decides the permutation to which areas of material science garner attention and gain resources. Machine translation has evolved from a rule-based coupled with statistical model to a very data-driven approach, and researchers are discussing the translation task with less and less reference to a specific source and target languages, pivoting towards advancing mode of computation for the task as a whole.

9.1.1 More deep integrations

We might also observe the trend of attempting to learn descriptors for parts of complex systems with ML models to be either more computationally efficient or more human interpretable or editable. Instead of scientists attempting to describe a system with equations from first principles, ML models can help scientists discover a better set of descriptors for systems across all datasets. For example, descriptors could be descriptors for input data (atomistic information/reaction space) or labels (crystal structures). These discoveries can hugely impact physics and chemistry theory [346], experiments and research methods. Physics-informed neural networks [347] can both improve neural network performance and physics research efficiency.

Increasingly, it could be more and more about the mapping of descriptors. We can imagine that with the emergence of more sophisticated models, it is possible to advance a particular segment of study in material science, such as polymer design, by completing a well-defined sophisticated task with a model/model of computation, where the lack of relevant databases will limit its advancement. Tracing the development of computer science, sophisticated models which perform generic tasks in material science well will again be integrated into a giant multi-purpose model much like a generic processing chip, to which we can prompt for insights which was previously only gained by human experimentation at a much slower rate. ML models will bring material scientists closer to the many possibilities already inherent in big data itself, allowing us to explore and exploit the possibilities with greater efficiency. The task material scientist will be able to complete with the help of machine learning will become more integrated and sophisticated, from the screening of material to the design of material as a complete task. Then with the design of material as an atomic/primitivetransaction, we will be able to come out with new science on top of the material design as a whole.

9.1.2 Systematic generalization

In our stride towards autonomous general intelligence (AGI), researchers have drawn many parallels and inspirations from neuro-sciences [348] and how humans learn and teach each other to develop models which better generalizes to novel situations and objects well. We expect a body of material science knowledge and ideas to become generalized and accessible to other fields, conveyed by advanced models in the future, where we can generalize or verbalize properties of imagined materials or predict performance of material in novel situations with high accuracy with its formal deduction process generated by models. We can also observe the interaction, cooperation or contradiction between bodies of materials science knowledge for novel materials and circumstances, and perform research on the intersection of bodies of knowledge with more depth and rigor. ML models can also learn to identify potential directions for exploration, come up with a comprehensive experimentation plan and collaborate with human researchers as a navigation co-pilot. The novel direction identified will be novel and comprehensive because models can learn from passive observations of a large material science literature, its publication trend [349] and insight analysis of researchers.

9.1.3 Huge computational models

With the development of reaction environment models, one might also reasonably expect reinforcement learning game-play learning to learn an agent policy for a material, i.e., to first learn a material behaviour that is desirable for a particular purpose, followed by an automatic search/generation of material which suits the specification. In general, the compounding of learning methods to get a solution for an even more vaguely defined objective but more analyzable process for that solution results in a human-verifiable solution for large or vague problems. Moreover, the increasing synthesizability or explainability of the solution to vague problems will help material scientists navigate methods for solving huge overarching/generic problems with more finesse, evolution of subject through large models [350].

Model of computation might become the common language of material scientists and researchers from other fields. Task definition might become the lingua franca or the leading cause of concern for ML practitioners in material science. This broader definition of material science might then, in turn, propel the advancement of machine learning. In general, the barrier to entry to both advanced material science and advanced ML will be lowered, allowing more experts from other fields or individuals to contribute their efforts and ideas to the development of both fields.

Mechanisms in quantum ML will become readily-available to be integrated with quantum physics, chemistry and subsequently material science. As classical-quantum hybrid infrastructure and architectures [351] become more available, quantum learning for material science might incorporate mechanisms of both quantum computing and quantum analysis of materials as primitives. This trend is expected to speed up the inter-disciplinary mixing of these fields from both engineering and theoretical grounds.

The resulting phenomenon is the emergence of an ever more integrated huge ML model, a Super Deep-Learning Model, which will tackle most if not all of the most fundamental underlying problems in material science; it will integrate fundamental engineering ideas from computer science with domain-invariants of material science, which is designed to perform well for various tasks on both super-computing facilities, quantum or otherwise, and on limited resources devices, [352] scalable yet robust. Moreover, by integrating the best training and privacy practices from ML software and hardware development experience, future material scientists can quickly expect robust material science downstream models running smoothly and reliably as an application on widely available and portable devices like a cellphone.

9.2 Perspectives from material science viewpoint

Currently, one of the biggest challenges is the availability of high-quality data. The increasing number of research groups adapting the open data approach and the growing availability of internet of things (IoT) devices will solve this problem, albeit gradually. We have also discussed several possible methods to overcome the issues, which is the advancement of small training sample ML algorithms, such as transfer learning and few-shot learning algorithms will also be one of the possible solutions to this issue.

9.2.1 Theoretical and computational materials science

The various computational techniques in materials science, such as DFT, molecular dynamics in its various forms of molecular dynamics (MD), monte-carlo methods, and density functional tight binding method, has started to benefit from the application of ML and will continue to do so in a dramatic manner.

As of now, the Kohn–Sham DFT remains a reliable and popular method for determining various material properties. However, the accuracy of DFT calculations heavily relies on the quality of the approximations employed, such as the exchange-correlation (XC) functional. The search for improved approximations, including exact functionals, using ML has only recently commenced. Another area for improvement in DFT is reducing computational costs. Recently, ML-refined numerical techniques have emerged that offer faster speeds compared to their traditional counterparts [353–355]. It is hoped that these advancements can eventually be applied to accelerate DFT computations.

The integration of ML into MD, exemplified by methods like DeePMD, has demonstrated the potential to achieve DFT-level accuracy while maintaining the computational efficiency of classical MD. This breakthrough opens up new possibilities for conducting accurate calculations in ab initio molecular dynamics (AIMD) on extremely large systems (with over 100 million atoms) or over very long timescales (beyond 1 nanosecond) [333, 355]. By enabling adequate sampling of phase space, these advancements enable more comprehensive investigations across various applications, including (electro-)catalysis, sensors, fabrication, drug interactions, and more.

9.2.2 Experimental materials science

The availability of a vast number of predicted materials with desired properties is highly advantageous for experimentalists. With a large number of possible candidates, the experimentalist can focus on the materials that can be synthesized and tested on available facilities and equipments. Additionally, the automated learning of the fabrication parameters and conditions are on the rise recently [356–359]. The advancement of MD will also enable comprehensive simulations of fabrication process and finds out the best experimental conditions for successful synthesizes of new materials [360, 361]. Furthermore, the analysis of the data, a highly time-consuming and laborious task is being increasingly supported by ML algorithms. The implementation of on-the-fly accurate inference mechanism of experimental data will increase the producitivity and efficiency of fabrication process, enabling experimentalists to determine if the samples have been fabricated successfully and move on to the next attempt quickly.

9.2.3 Coupling of data-driven discovery with traditional techniques

The art of tailoring and creating materials with desired properties — materials engineering, includes techniques such as defect engineering [361–363], doping [44, 362, 364–369], fluorinating [370] or alloy engineering [30–34, 371] or salt engineering [35] by varying composition, strain engineering [11, 16, 26, 36, 170, 372–374] by applying mechanical load (such as hydro-static pressure [36, 375–379] or directional stress), and interfacial engineering [40, 42–46] by choosing different materials for forming interface with novel or exotic properties. These methods have been demonstrated to be very useful for tuning materials properties or creating new materials with advantageous properties. In quantum materials [14, 28, 37, 380], including strongly correlated functional materials and superconducting materials, the charge-spin-orbital engineering plays a crucial role in controlling the quantum behaviour.

The availability of advanced X-ray scattering and electron scattering techniques [14, 28, 191, 381, 382] such as synchrotron radiation and electron microscropy, and advancing nanotehnologies and simulation methods, has led to an increasingly growing amount of high quality experimental or simulated data, available for data-driven discovery and data mining. The integration of data-driven discovery with traditional techniques is expected to play an increasingly important role in materials science research at various length and time scales, ranging from microscropic scale to macroscropic scale.

10 Conclusion

In conclusion, this review briefly introduced basic concepts and history of machine learning, and provided detailed information of coupling between machine learning and materials science in fundamental and technical perspectives. The nuances of machine learning, from the descriptors to the various algorithms, have been discussed in the context of materials science. Besides, this review also covered the tasks or issues in materials scicence that has been tackled with the use of machine learning. We also discussed our vision for the future of materials science as the field matures with the integration of machine learning, which will be drastically different from what we know today.

Declarations The authors declare that they have no competing interests and there are no conflicts.

Acknowledgements This research was supported by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme (No. FRGS/1/2021/STG05/XMU/01/1).

References and notes

1. E. Weinan, Machine learning and computational mathematics, *Commun. Comput. Phys.* 28, 1639 (2020)
2. A. Agrawal and A. Choudhary, Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science, *APL Mater.* 4(5), 053208 (2016)
3. Y. Xu, X. Liu, X. Cao, C. Huang, E. Liu, et al., Artificial intelligence: A powerful paradigm for scientific research, *The Innovation* 2(4), 100179 (2021)
4. G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* 91, 045002 (2019)
5. G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, From DFT to machine learning: Recent approaches to materials science – A review, *J. Phys.: Mater.* 2(3), 032001 (2019)
6. R. Potyrailo, K. Rajan, K. Stoewe, I. Takeuchi, B. Chisholm, and H. Lam, Combinatorial and high-throughput screening of materials libraries: Review of state of the art, *ACS Combin. Sci.* 13(6), 579 (2011)
7. K. Alberi, M. B. Nardelli, A. Zakutayev, L. Mitas, S. Curtarolo, et al., The 2019 materials by design roadmap, *J. Phys. D* 52(1), 013001 (2018)
8. S. Torquato, Optimal design of heterogeneous materials, *Ann. Rev. Mater. Res.* 40, 101 (2010)
9. A. A. White, Big data are shaping the future of materials science, *MRS Bull.* 38(8), 594 (2013)
10. Z. Fan, H. Q. Wang, and J. C. Zheng, Searching for the best thermoelectrics through the optimization of transport distribution function, *J. Appl. Phys.* 109(7), 073713 (2011)
11. J. C. Zheng and Y. Zhu, Searching for a higher superconducting transition temperature in strained $\mathrm{MgB_2}$, *Phys. Rev. B* 73, 024509 (2006)
12. J. C. Zheng, Asymmetrical transport distribution function: Skewness as a key to enhance thermoelectric performance, *Research* 2022, 9867639 (2022)
13. J. C. Zheng, Recent advances on thermoelectric materials, *Front. Phys. China* 3(3), 269 (2008)
14. J. C. Zheng, A. I. Frenkel, L. Wu, J. Hanson, W. Ku, E. S. Bozin, S. J. L. Billinge, and Y. Zhu, Nanoscale disorder and local electronic properties of $\mathrm{CaCu_3Ti_4O_{12}}$:

An integrated study of electron, neutron, and X-ray diffraction, X-ray absorption fine structure, and first-principles calculations, *Phys. Rev. B* 81(14), 144203 (2010)

15. N. Sa, S. S. Chong, H. Q. Wang, and J. C. Zheng, Anisotropy engineering of ZnO nanoporous frameworks: A lattice dynamics simulation, *Nanomaterials (Basel)* 12(18), 3239 (2022)
16. H. Cheng and J. C. Zheng, *Ab initio* study of anisotropic mechanical and electronic properties of strained carbon-nitride nanosheet with interlayer bonding, *Front. Phys.* 16(4), 43505 (2021)
17. Y. Huang, C. Y. Haw, Z. Zheng, J. Kang, J. C. Zheng, and H. Q. Wang, Biosynthesis of zinc oxide nanomaterials from plant extracts and future green prospects: A topical review, *Adv. Sustain. Syst.* 5(6), 2000266 (2021)
18. Z. Q. Wang, H. Cheng, T. Y. Lü, H. Q. Wang, Y. P. Feng, and J. C. Zheng, A super-stretchable boron nanoribbon network, *Phys. Chem. Chem. Phys.* 20(24), 16510 (2018)
19. Y. Li, H. Q. Wang, T. J. Chu, Y. C. Li, X. Li, X. Liao, X. Wang, H. Zhou, J. Kang, K. C. Chang, T. C. Chang, T. M. Tsai, and J. C. Zheng, Tuning the nanostructures and optical properties of undoped and N-doped ZnO by supercritical fluid treatment, *AIP Adv.* 8(5), 055310 (2018)
20. Y. L. Li, Z. Fan, and J. C. Zheng, Enhanced thermoelectric performance in graphitic ZnO (0001) nanofilms, *J. Appl. Phys.* 113(8), 083705 (2013)
21. J. He, I. D. Blum, H. Q. Wang, S. N. Girard, J. Doak, L. D. Zhao, J. C. Zheng, G. Casillas, C. Wolverton, M. Jose-Yacaman, D. N. Seidman, M. G. Kanatzidis, and V. P. Dravid, Morphology control of nanostructures: Na-doped PbTe–PbS system, *Nano Lett.* 12(11), 5979 (2012)
22. Z. Fan, J. Zheng, H. Q. Wang, and J. C. Zheng, Enhanced thermoelectric performance in three-dimensional superlattice of topological insulator thin films, *Nanoscale Res. Lett.* 7(1), 570 (2012)
23. N. Wei, H. Q. Wang, and J. C. Zheng, Nanoparticle manipulation by thermal gradient, *Nanoscale Res. Lett.* 7(1), 154 (2012)
24. N. Wei, Z. Fan, L. Q. Xu, Y. P. Zheng, H. Q. Wang, and J. C. Zheng, Knitted graphene-nanoribbon sheet: A mechanically robust structure, *Nanoscale* 4(3), 785 (2012)
25. J. Q. He, J. R. Sootsman, L. Q. Xu, S. N. Girard, J. C. Zheng, M. G. Kanatzidis, and V. P. Dravid, Anomalous electronic transport in dual-nanostructured lead telluride, *J. Am. Chem. Soc.* 133(23), 8786 (2011)
26. N. Wei, L. Xu, H. Q. Wang, and J. C. Zheng, Strain engineering of thermal conductivity in graphene sheets and nanoribbons: A demonstration of magic flexibility, *Nanotechnology* 22(10), 105705 (2011)
27. J. He, J. R. Sootsman, S. N. Girard, J. C. Zheng, J. Wen, Y. Zhu, M. G. Kanatzidis, and V. P. Dravid, On the origin of increased phonon scattering in nanostructured PbTe-based thermoelectric materials, *J. Am. Chem. Soc.* 132(25), 8669 (2010)
28. Y. Zhu, J. C. Zheng, L. Wu, A. I. Frenkel, J. Hanson, P. Northrup, and W. Ku, Nanoscale disorder in $CaCu_3Ti_4O_{12}$: A new route to the enhanced dielectric response, *Phys. Rev. Lett.* 99(3), 037602 (2007)
29. J. C. Zheng, H. Q. Wang, A. T. S. Wee, and C. H. A. Huan, Structural and electronic properties of Al nanowires: An *ab initio* pseudopotential study, *Int. J. Nanosci.* 01(02), 159 (2002)
30. J. C. Zheng, H. Q. Wang, A. T. S. Wee, and C. H. A. Huan, Possible complete miscibility of $(BN)_x(C_2)_{1-x}$ alloys, *Phys. Rev. B* 66(9), 092104 (2002)
31. J. C. Zheng, H. Q. Wang, C. H. A. Huan, and A. T. S. Wee, The structural and electronic properties of $(AlN)_x(C_2)_{1-x}$ and $(AlN)_x(BN)_{1-x}$ alloys, *J. Phys.: Condens. Matter* 13(22), 5295 (2001)
32. H. Q. Wang, J. C. Zheng, R. Z. Wang, Y. M. Zheng, and S. H. Cai, Valence-band offsets of III–V alloy heterojunctions, *Surf. Interface Anal.* 28(1), 177 (1999)
33. J. C. Zheng, R. Z. Wang, Y. M. Zheng, and S. H. Cai, Valence offsets of three series of alloy heterojunctions, *Chin. Phys. Lett.* 14(10), 775 (1997)
34. J. C. Zheng, Y. Zheng, and R. Wang, Valence offsets of ternary alloy heterojunctions $In_xGa_{1-x}As/In_xAl_{1-x}As$, *Chin. Sci. Bull.* 41(24), 2050 (1996)
35. L. Liu, T. Wang, L. Sun, T. Song, H. Yan, C. Li, D. Mu, J. Zheng, and Y. Dai, Stable cycling of all-solid-state lithium metal batteries enabled by salt engineering of PEO-based polymer electrolytes, *Energy Environ. Mater.* (Feb.), e12580 (2023)
36. W. Zhang, F. Y. Du, Y. Dai, and J. C. Zheng, Strain engineering of Li^+ ion migration in olivine phosphate cathode materials $LiMPO_4$ (M = Mn, Fe, Co) and $(LiFePO_4)_n(LiMnPO_4)_m$ superlattices, *Phys. Chem. Chem. Phys.* 25(8), 6142 (2023)
37. B. Zhang, L. Wu, J. Zheng, P. Yang, X. Yu, J. Ding, S. M. Heald, R. A. Rosenberg, T. V. Venkatesan, J. Chen, C. J. Sun, Y. Zhu, and G. M. Chow, Control of magnetic anisotropy by orbital hybridization with charge transfer in $(La_{0.67}Sr_{0.33}MnO_3)_n/(SrTiO_3)_n$ superlattice, *NPG Asia Mater.* 10(9), 931 (2018)
38. L. Zhang, T. Y. Lü, H. Q. Wang, W. X. Zhang, S. W. Yang, and J. C. Zheng, First principles studies on the thermoelectric properties of $(SrO)_m(SrTiO_3)_n$ superlattice, *RSC Adv.* 6(104), 102172 (2016)
39. J. C. Zheng, C. H. A. Huan, A. T. S. Wee, M. A. V. Hove, C. S. Fadley, F. J. Shi, E. Rotenberg, S. R. Barman, J. J. Paggel, K. Horn, P. Ebert, and K. Urban, Atomic scale structure of the 5-fold surface of a AlPdMn quasicrystal: A quantitative X-ray photoelectron diffraction analysis, *Phys. Rev. B* 69(13), 134107 (2004)
40. H. Q. Wang, J. Xu, X. Lin, Y. Li, J. Kang, and J. C. Zheng, Determination of the embedded electronic states at nanoscale interface via surface-sensitive photoemission spectroscopy, *Light Sci. Appl.* 10(1), 153 (2021)
41. M. A. Van Hove, K. Hermann, and P. R. Watson, The NIST surface structure database – SSD version 4, *Acta Crystallogr. B* 58(3), 338 (2002)
42. H. Q. Wang, E. Altman, C. Broadbridge, Y. Zhu, and V. Henrich, Determination of electronic structure of oxide-oxide interfaces by photoemission spectroscopy, *Adv. Mater.* 22, 2950 (2010)

43. H. Zhou, L. Wu, H. Q. Wang, J. C. Zheng, L. Zhang, K. Kisslinger, Y. Li, Z. Wang, H. Cheng, S. Ke, Y. Li, J. Kang, and Y. Zhu, Interfaces between hexagonal and cubic oxides and their structure alternatives, *Nat. Commun.* 8(1), 1474 (2017)
44. J. D. Steiner, H. Cheng, J. Walsh, Y. Zhang, B. Zydlewski, L. Mu, Z. Xu, M. M. Rahman, H. Sun, F. M. Michel, C. J. Sun, D. Nordlund, W. Luo, J. C. Zheng, H. L. Xin, and F. Lin, Targeted surface doping with reversible local environment improves oxygen stability at the electrochemical interfaces of nickel-rich cathode materials, *ACS Appl. Mater. Interfaces* 11(41), 37885 (2019)
45. J. C. Zheng, H. Q. Wang, A. T. S. Wee, and C. H. A. Huan, Trends in bonding configuration at SiC/III–V semiconductor interfaces, *Appl. Phys. Lett.* 79(11), 1643 (2001)
46. H. Q. Wang, J. C. Zheng, A. T. S. Wee, and C. H. A. Huan, Study of electronic properties and bonding configuration at the BN/SiC interface, *J. Electron Spectrosc. Relat. Phenom.* 114–116, 483 (2001)
47. S. Lin, B. Zhang, T. Y. Lü, J. C. Zheng, H. Pan, H. Chen, C. Lin, X. Li, and J. Zhou, Inorganic lead-free B-γ-CsSnI 3 perovskite solar cells using diverse electron-transporting materials: A simulation study, *ACS Omega* 6(40), 26689 (2021)
48. F. Y. Du, W. Zhang, H. Q. Wang, and J. C. Zheng, Enhancement of thermal rectification by asymmetry engineering of thermal conductivity and geometric structure for the multi-segment thermal rectifier, *Chin. Phys. B* 32(6), 064402 (2023)
49. M. Kulichenko, J. S. Smith, B. Nebgen, Y. W. Li, N. Fedik, A. I. Boldyrev, N. Lubbers, K. Barros, and S. Tretiak, The rise of neural networks for materials and chemical dynamics, *J. Phys. Chem. Lett.* 12(26), 6227 (2021)
50. W. Sha, Y. Guo, Q. Yuan, S. Tang, X. Zhang, S. Lu, X. Guo, Y. C. Cao, and S. Cheng, Artificial intelligence to power the future of materials science and engineering, *Adv. Intell. Syst.* 2(4), 1900143 (2020)
51. S. Leonelli, Scientific research and big data, in: The Stanford Encyclopedia of Philosophy, Summer 2020 Ed., edited by E. N. Zalta, Metaphysics Research Lab, Stanford University, 2020
52. J. Westermayr, M. Gastegger, K. T. Schütt, and R. J. Maurer, Perspective on integrating machine learning into computational chemistry and materials science, *J. Chem. Phys.* 154(23), 230903 (2021)
53. D. Morgan and R. Jacobs, Opportunities and challenges for machine learning in materials science, *Annu. Rev. Mater. Res.* 50(1), 71 (2020)
54. C. Chen, Y. Zuo, W. Ye, X. Li, Z. Deng, and S. P. Ong, A critical review of machine learning of energy materials, *Adv. Energy Mater.* 10(8), 1903242 (2020)
55. J. Wei, X. Chu, X. Y. Sun, K. Xu, H. X. Deng, J. Chen, Z. Wei, and M. Lei, Machine learning in materials science, *InfoMat* 1(3), 338 (2019)
56. G. Pilania, Machine learning in materials science: From explainable predictions to autonomous design, *Comput. Mater. Sci.* 193, 110360 (2021)
57. K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Machine learning for molecular and materials science, *Nature* 559(7715), 547 (2018)
58. F. Oviedo, J. L. Ferres, T. Buonassisi, and K. T. Butler, Interpretable and explainable machine learning for materials science and chemistry, *Acc. Mater. Res.* 3(6), 597 (2022)
59. J. F. Rodrigues Jr, M. C. F. Florea, D. de Oliveira, D. Diamond, and O. N. Oliveira Jr, Big data and machine learning for materials science, *Discover Materials* 1(1), 12 (2021)
60. K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, and C. Wolverton, Recent advances and applications of deep learning methods in materials science, *npj Comput. Mater.* 8, 59 (2022)
61. L. Samuel, Some studies in machine learning using the game of checkers, *IBM J. Res. Develop.* 3(3), 210 (1959)
62. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees, 1983
63. L. G. Valiant, A theory of the learnable, in: STOC ’84 Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, pp 436–445, 1984
64. T. Mitchell, Machine Learning, New York, USA: McGrawHill, 1997
65. S. Roweis and Z. Ghahramani, A unifying review of linear gaussian models, *Neural Comput.* 11(2), 305 (1999)
66. J. C. Zheng, J. Y. Chen, J. W. Shuai, S. H. Cai, and R. Z. Wang, Storage capacity of the Hopfield neural network, *Physica A* 246(3), 313 (1997)
67. J. W. Shuai, J. C. Zheng, Z. X. Chen, R. T. Liu, and B. X. Wu, The three-dimensional rotation neural network, *Physica A* 238), 23 (1997)
68. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, 2nd Ed. , Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2018
69. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, arXiv: 1706.03762 (2017)
70. A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, SuperGLUE: A stickier benchmark for general-purpose language understanding systems, arXiv: 1905.00537 (2019)
71. D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent, and S. Bengio, Why does unsupervised pre-training help deep learning, *J. Mach. Learn. Res.* 11, 625 (2010)
72. Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou, CodeBERT: A pre-trained model for programming and natural languages, arXiv: 2002.08155 (2020)
73. H. Bao, L. Dong, and F. Wei, BEIT: BERT pre-training of image transformers, arXiv: 2106.08254 (2021)
74. K. Hakhamaneshi, M. Nassar, M. Phielipp, P. Abbeel, and V. Stojanović, Pretraining graph neural networks for few-shot analog circuit modeling and design, arXiv: 2203.15913 (2022)

75. J. Li, D. Li, C. Xiong, and S. Hoi, BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation, arXiv: 2201.12086 (2022)
76. K. Lu, A. Grover, P. Abbeel, and I. Mordatch, Pretrained transformers as universal computation engines, arXiv: 2103.05247 (2021)
77. M. Reid, Y. Yamada, and S. S. Gu, Can Wikipedia help offline reinforcement learning? arXiv: 2201.12122 (2022)
78. C. Sun, X. Qiu, Y. Xu, and X. Huang, How to fine-tune BERT for text classification? arXiv: 1905.05583 (2019)
79. H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, *Advances in Neural Information Processing Systems* 35, 1950 (2022)
80. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv: 1810.04805 (2018)
81. Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, arXiv: 1909.11942 (2019)
82. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ROBERTA: A robustly optimized BERT pretraining approach, arXiv: 1907.11692 (2019)
83. J. Vig and Y. Belinkov, Analyzing the structure of attention in a transformer language model, arXiv: 1906.04284 (2019)
84. S. Zhang and L. Xie, Improving attention mechanism in graph neural networks via cardinality preservation, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2020, page 1395
85. Y. Tay, V. Q. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. Gupta, T. Schuster, W. W. Cohen, and D. Metzler, Transformer memory as a differentiable search index, *Advances in Neural Information Processing Systems* 35, 21831 (2022)
86. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *J. Mach. Learn. Res.* 21(1), 5485 (2020)
87. T. Shin, and Y. Razeghi, R. L. L. IV, E. Wallace, and S. Singh, AutoPrompt: Eliciting knowledge from language models with automatically generated prompts, arXiv: 2010.15980 (2020)
88. N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H. -T. Zheng, and M. Sun, Openprompt: An open-source framework for prompt-learning, arXiv: 2111.01998 (2021)
89. S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, OPT: Open pre-trained transformer language models, arXiv: 2205.01068 (2022)
90. O. Lieber, O. Sharir, B. Lenz, and Y. Shoham, Jurassic-1: Technical Details and Evaluation, AI21 Labs, Tech. Rep., 2021
91. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 33 Curran Associates, Inc., 2020, pp 1877–1901, arXiv: 2005.14165
92. A. Bapna, I. Caswell, J. Kreutzer, O. Firat, D. van Esch, et al., Building machine translation systems for the next thousand languages, arXiv: 2205.03983 (2022)
93. T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space, arXiv: 1301.3781 (2013)
94. J. Pennington, R. Socher, and C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp 1532–1543
95. O. Melamud, J. Goldberger, and I. Dagan, Context2vec: Learning generic context embedding with bidirectional LSTM, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp 51–61
96. H. Dai, B. Dai, and L. Song, Discriminative embeddings of latent variable models for structured data, arXiv: 1603.05629 (2016)
97. J. Yang, R. Zhao, M. Zhu, D. Hallac, J. Sodnik, and J. Leskovec, Driver2vec: Driver identification from automotive data, arXiv: 2102.05234 (2021)
98. S. Schneider, A. Baevski, R. Collobert, and M. Auli, Wav2vec: Unsupervised pre-training for speech recognition, arXiv: 1904.05862 (2019)
99. H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence 35(12), 11106 (2021), arXiv: 2012.07436
100. I. Beltagy, M. E. Peters, and A. Cohan, Longformer: The long-document transformer, arXiv: 2004.05150 (2020)
101. K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, A survey on vision transformer, *IEEE Trans. Pattern Anal. Mach. Intell.* 45(1), 87 (2023)
102. J. B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, et al., Flamingo: A visual language model for few-shot learning, *Advances in Neural Information Processing Systems* 35, 23716 (2022), arXiv: 2204.14198
103. J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, COCA: Contrastive captioners are image-text foundation models, arXiv: 2205.01917 (2022)
104. X. Liu, C. Gong, L. Wu, S. Zhang, H. Su, and Q. Liu, Fusedream: Training-free text-to-image generation with improved CLIP+GAN space optimization, arXiv: 2112.01573 (2021)

105. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, Learning transferable visual models from natural language supervision, arXiv: 2103.00020 (2021)
106. L. He, Q. Zhou, X. Li, L. Niu, G. Cheng, X. Li, W. Liu, Y. Tong, L. Ma, and L. Zhang, End-to-end video object detection with spatial-temporal transformers, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp 1507–1516, arXiv: 2105.10920
107. X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, LIT: Zero-shot transfer with locked-image text tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp 18123–18133, arXiv: 2111.07991
108. A. Trockman and J. Z. Kolter, Patches are all you need? arXiv: 2201.09792 (2022)
109. A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, Zeroshot text-to-image generation, in: International Conference on Machine Learning, 2021, pp 8821–8831, arXiv: 2102.12092
110. A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, Y. Wang, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Niessner, J. T. Barron, G. Wetzstein, M. Zollhoefer, and V. Golyanik, Advances in neural rendering, *Computer Graphics Forum* 41(2), 703 (2022), arXiv: 2111.05849
111. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, NERF: Representing scenes as neural radiance fields for view synthesis, *Communications of the ACM* 65(1), 99 (2021), arXiv: 2003.08934
112. S. Zheng, J. Pan, C. Lu, and G. Gupta, Pointnorm: Normalization is all you need for point cloud analysis, arXiv: 2207.06324 (2022)
113. H. Ran, J. Liu, and C. Wang, Surface representation for point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp 18942–18952, arXiv: 2205.05740
114. X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, Rethinking network design and local geometry in point cloud: A simple residual MLP framework, arXiv: 2202.07123 (2022)
115. Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, Dynamic graph CNN for learning on point clouds, arXiv: 1801.07829 (2018)
116. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, Mastering the game of Go without human knowledge, *Nature* 550(7676), 354 (2017)
117. E. Zhao, R. Yan, J. Li, K. Li, and J. Xing, Alphaholdem: High-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning, in: Proceedings of the AAAI Conference on Artificial Intelligence 36(4), 4689 (2022)
118. S. Zou, T. Xu, and Y. Liang, Finite-sample analysis for SARSA with linear function approximation, arXiv: 1902.02234 (2019)
119. C. J. C. H. Watkins and P. Dayan, Q-learning, *Machine Learning* 8(3), 279 (1992)
120. P. Abbeel and A. Y. Ng, Apprenticeship learning via inverse reinforcement learning, in Proceedings of the Twenty-First International Conference on Machine Learning, Ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004
121. C. Finn, P. Abbeel, and S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, In International conference on machine learning, 2017, pp 1126–1135, arXiv: 1703.03400
122. C. Fifty, E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, Efficiently identifying task groupings for multi-task learning, *Advances in Neural Information Processing Systems* 34, 27503 (2021), arXiv: 2109.04617
123. N. Anand and D. Precup, Preferential temporal difference learning, arXiv: 2106.06508 (2021)
124. K. Chen, R. Cao, S. James, Y. Li, Y. H. Liu, P. Abbeel, and Q. Dou, Sim-to-real 6d object pose estimation via iterative self-training for robotic bin-picking, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX (pp 533–550). Cham: Springer Nature Switzerland, arXiv: 2204.07049
125. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, Playing atari with deep reinforcement learning, arXiv: 1312.5602 (2013)
126. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, Continuous control with deep reinforcement learning, arXiv: 1509.02971 (2015)
127. D. Yarats, D. Brandfonbrener, H. Liu, M. Laskin, P. Abbeel, A. Lazaric, and L. Pinto, Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning, arXiv: 2201.13425 (2022)
128. M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, et al., Do as I can, not as I say: Grounding language in robotic affordances, in: Conference on Robot Learning, 2023, pp 287–318, arXiv: 2204.01691
129. S. James and P. Abbeel, Coarse-to-fine Q-attention with learned path ranking, arXiv: 2204.01571 (2022)
130. C. Qi, P. Abbeel, and A. Grover, Imitating, fast and slow: Robust learning from demonstrations via decision-time planning, arXiv: 2204.03597 (2022)
131. L. Wang, X. Zhang, K. Yang, L. Yu, C. Li, L. Hong, S. Zhang, Z. Li, Y. Zhong, and J. Zhu, Memory replay with data compression for continual learning, arXiv: 2202.06592 (2022)
132. L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, Decision transformer: Reinforcement learning via sequence modeling, *Advances in Neural Information Processing Systems* 34, 15084 (2021), arXiv: 2106.01345
133. J. Parker-Holder, M. Jiang, M. Dennis, M. Samvelyan, J. Foerster, E. Grefenstette, and T. Rocktäschel, Evolving curricula with regret-based environment

design, in: International Conference on Machine Learning, 2022, pp 17473–17498, arXiv: 2203.01302

134. R. Wang, J. Lehman, J. Clune, and K. O. Stanley, Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions, arXiv: 1901.01753 (2019)

135. Z. Li, L. Li, Z. Ma, P. Zhang, J. Chen, and J. Zhu, Read: Large-scale neural scene rendering for autonomous driving, arXiv: 2205.05509 (2022)

136. W. Tang, C. J. Ho, and Y. Liu, Bandit learning with delayed impact of actions, in: Advances in Neural Information Processing Systems, edited by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, 2021, arXiv: 1904.01763

137. Z. Gao, Y. Han, Z. Ren, and Z. Zhou, Batched multi-armed bandits problem, in: Advances in Neural Information Processing Systems, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Curran Associates, Inc., 2019, arXiv: 1904.01763

138. Y. Yue, J. Broder, R. Kleinberg, and T. Joachims, The k-armed dueling bandits problem, *J. Comput. Syst. Sci.* 78(5), 1538 (2012)

139. A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, P. Auer, and A. Antos, Upperconfidence-bound algorithms for active learning in multi-armed bandits, in: Algorithmic Learning Theory: 22nd International Conference, ALT 2011, Espoo, Finland, October 5–7, 2011. Proceedings 22 (pp 189–203), Springer Berlin Heidelberg, arXiv: 1507.04523

140. W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao, Mastering Atari games with limited data, *Advances in Neural Information Processing Systems* 34, 25476 (2021), arXiv: 2111.00210

141. M. Samvelyan, T. Rashid, C. Schroeder de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C. M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson, The StarCraft multi-agent challenge, in: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019, arXiv: 1902.04043

142. T. Wang, T. Gupta, A. Mahajan, B. Peng, S. Whiteson, and C. Zhang, Rode: Learning roles to decompose multi-agent tasks, arXiv: 2010.01523 (2020)

143. O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, et al., Grandmaster level in StarCraft II using multi-agent reinforcement learning, *Nature* 575(7782), 350 (2019)

144. W. Du and S. Ding, A survey on multi-agent deep reinforcement learning: From the perspective of challenges and applications, *Artif. Intell. Rev.* 54(5), 3215 (2021)

145. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature* 549, 195 (2017)

146. Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, *Nat. Phys.* 17(9), 1013 (2021)

147. V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, *Nature* 567(7747), 209 (2019)

148. S. Moradi, C. Brandner, C. Spielvogel, D. Krajnc, S. Hillmich, R. Wille, W. Drexler, and L. Papp, Clinical data classification with noisy intermediate scale quantum computers, *Sci. Rep.* 12(1), 1851 (2022)

149. J. Zheng, K. He, J. Zhou, Y. Jin, and C. M. Li, Combining reinforcement learning with lin-kernighan-helsgaun algorithm for the traveling salesman problem, in: Proceedings of the AAAI Conference on Artificial Intelligence 35(14), 12445 (2021), arXiv: 2012.04461

150. Z. Li, Q. Chen, and V. Koltun, Combinatorial optimization with graph convolutional networks and guided tree search, *Advances in Neural Information Processing Systems* 31, 2018, arXiv: 1810.10659

151. M. Sundararajan, A. Taly, and Q. Yan, Axiomatic attribution for deep networks, in: International Conference on Machine Learning, 2017, pp 3319–3328, arXiv: 1703.01365

152. M. T. Ribeiro, S. Singh, and C. Guestrin, Why Should I Trust You? Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp 1135–1144, arXiv: 1602.04938

153. S. Lundberg and S. I. Lee, A unified approach to interpreting model predictions, arXiv: 1705.07874 (2017)

154. J. Crabbe, Z. Qian, F. Imrie, and M. van der Schaar, Explaining latent representations with a corpus of examples, in: Advances in Neural Information Processing Systems, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Curran Associates, Inc., 2021, pp 12154–12166, arXiv: 2110.15355

155. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, Striving for simplicity: The all convolutional net, arXiv: 1412.6806 (2014)

156. R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, Gnnexplainer: Generating explanations for graph neural networks, arXiv: 1903.03894 (2019)

157. H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, On explainability of graph neural networks via subgraph explorations, in: International Conference on Machine Learning, 2021, pp 12241–12252, arXiv: 2102.05152

158. Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, GraphLIME: Local interpretable model explanations for graph neural networks, *IEEE Transactions on Knowledge and Data Engineering*, 35(7), 6968 (2023), arXiv: 2001.06216

159. H. Yuan, H. Yu, S. Gui, and S. Ji, Explainability in graph neural networks: A taxonomic survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(5), 5782 (2023), arXiv: 2012.15445

160. G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer, ReLUPlex: An efficient smt solver for verifying deep neural networks, in: Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24–28, 2017, Proceedings, Part I 30, pp 97–117. Springer International Publishing, arXiv: 1702.01135

161. S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. J. Hsieh, and J. Z. Kolter, Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verifica-

tion, *Advances in Neural Information Processing Systems* 34, 2021, arXiv: 2103.06624

162. M. P. Owen, A. Panken, R. Moss, L. Alvarez, and C. Leeper, ACAS Xu: Integrated collision avoidance and detect and avoid capability for UAS, in: IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), 2019
163. S. Mittal and S. Vaishay, A survey of techniques for optimizing deep learning on GPUS, *J. Systems Archit.* 99, 101635 (2019)
164. F. Wang, W. Zhang, S. Lai, M. Hao, and Z. Wang, Dynamic GPU energy optimization for machine learning training workloads, *IEEE Transactions on Parallel and Distributed Systems* 33(11), 2943 (2022)
165. R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, P. Warden, and R. Rhodes, Tensorflow lite micro: Embedded machine learning for tinyML systems, in: Proceedings of Machine Learning and Systems, edited by A. Smola, A. Dimakis, and I. Stoica, 2021, pp 800–811, arXiv: 2010.08678
166. C. Tanasescu, V. Kesarwani, and D. Inkpen, Metaphor detection by deep learning and the place of poetic metaphor in digital humanities, in: The Thirty-First International Flairs Conference, 2018
167. H. Surden, Machine learning and law, *Wash. L. Rev.* 89, 87 (2014)
168. J. De Spiegeleer, D. B. Madan, S. Reyners, and W. Schoutens, Machine learning for quantitative finance: Fast derivative pricing, hedging and fitting, *Quantitative Finance* 18(10), 1635–1643, 2018
169. W. Solano-Alvarez, M. Peet, E. Pickering, J. Jaiswal, A. Bevan, and H. Bhadeshia, Synchrotron and neural network analysis of the influence of composition and heat treatment on the rolling contact fatigue of hypereutectoid pearlitic steels, *Materials Science and Engineering A* 707, 259 (2017)
170. J. J. Li, Y. Dai, and J. C. Zheng, Strain engineering of ion migration in $LiCoO_2$, *Front. Phys.* 17(1), 13503 (2022)
171. H. K. D. H. Bhadeshia, Neural networks and information in materials science, *Statistical Analysis and Data Mining* 1, 296 (2009)
172. Y. Liu, O. C. Esan, Z. Pan, and L. An, Machine learning for advanced energy materials, *Energy and AI* 3, 100049 (2021)
173. S. R. Kalidindi, Feature engineering of material structure for AI-based materials knowledge systems, *J. Appl. Phys.* 128(4), 041103 (2020)
174. Z. Xiang, M. Fan, G. Vázquez Tovar, W. Trehengekrn, B. J. Yoon, X. Qian, R. Arroyave, and X. Qian, Physics-constrained automatic feature engineering for predictive modeling in materials science, in: Proceedings of the AAAI Conference on Artificial Intelligence 35(12), pp 10414–10421 (2021)
175. Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.* 35(8), 1798 (2013)
176. P. K. Routh, Y. Liu, N. Marcella, B. Kozinsky, and A. I. Frenkel, Latent representation learning for structural characterization of catalyst, *J. Phys. Chem. Lett.* 12(8), 2086 (2021)
177. A. Franceschetti and A. Zunger, The inverse bandstructure problem of finding an atomic configuration with given electronic properties, *Nature* 402(6757), 6757 (1999)
178. Z. Liu, D. Zhu, L. Raju, and W. Cai, Tackling photonic inverse design with machine learning, *Adv. Sci.* 8, 2002923 (2021)
179. J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, Materials design and discovery with high-throughput density functional theory: The open quantum materials database (OQMD), *JOM* 65(11), 1501 (2013)
180. S. 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(1), 15010 (2015)
181. A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. Persson, The materials project: A materials genome approach to accelerating materials innovation, *APL Mater.* 1(1), 011002 (2013)
182. K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, et al., The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design, *npj Comput. Mater.* 6(1), 173 (2020)
183. AFLOW, URL: aflowlib.org
184. MatCloud, URL: matcloud.com.cn
185. MPDS, Pauling File, URL: mpds.io
186. NOMAD, URL: nomad-lab.eu
187. C2DB, URL: cmr.fysik.dtu.dk/c2db/c2db.html
188. J. Zhou, L. Shen, M. D. Costa, K. A. Persson, S. P. Ong, P. Huck, Y. Lu, X. Ma, Y. Chen, H. Tang, and Y. P. Feng, 2dmatpedia, an open computational database of two-dimensional materials from top-down and bottom-up approaches, Scientific Data 6, 86, June 2019
189. M. Hellenbrandt, The inorganic crystal structure database (ICSD) — Present and future, *Crystallography Rev.* 10(1), 17 (2004)
190. S. Gražulis, A. Daškevič, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N. R. Serebryanaya, P. Moeck, R. T. Downs, and A. Le Bail, Crystallography Open Database (COD): An open-access collection of crystal structures and platform for world-wide collaboration, *Nucleic Acids Research* 40(D1), D420 (2011)
191. J. C. Zheng, L. Wu, and Y. Zhu, Aspherical electron scattering factors and their parameterizations for elements from H to Xe, *Journal of Applied Crystallography* 42, 1043 (2009)
192. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, et al., Highly accurate protein structure prediction with alphafold, *Nature* 596(7873), 583 (2021)
193. A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, Benchmarking materials property prediction methods: The matbench test set and automatminer reference algorithm, *npj Comput. Mater.* 6(1), 138 (2020)
194. R. Lin, R. Zhang, C. Wang, X. Q. Yang, and H. L. Xin, Temimagenet training library and atomsegnet

deep-learning models for high-precision atom segmentation, localization, denoising, and deblurring of atomic-resolution images, *Sci. Rep.* 11(1), 5386 (2021)

195. L. Han, H. Cheng, W. Liu, H. Li, P. Ou, R. Lin, H.-T. Wang, C.-W. Pao, A. R. Head, C.-H. Wang, X. Tong, C.-J. Sun, W.-F. Pong, J. Luo, J.-C. Zheng, and H. L. Xin, A single-atom library for guided monometallic and concentration-complex multimetallic designs, *Nat. Mater.* 21, 681 (2022)
196. D. Mrdjenovich, M. K. Horton, J. H. Montoya, C. M. Legaspi, S. Dwaraknath, V. Tshitoyan, A. Jain, and K. A. Persson, Propnet: A knowledge graph for materials science, *Matter* 2(2), 464 (2020)
197. T. S. Lin, C. W. Coley, H. Mochigase, H. K. Beech, W. Wang, Z. Wang, E. Woods, S. L. Craig, J. A. Johnson, J. A. Kalow, K. F. Jensen, and B. D. Olsen, Bigsmiles: A structurally-based line notation for describing macromolecules, *ACS Cent. Sci.* 5(9), 1523 (2019)
198. M. Krenn, Q. Ai, S. Barthel, N. Carson, A. Frei, et al., Selfies and the future of molecular string representations, *Patterns* 3(10), 100588 (2022)
199. K. Michel and B. Meredig, Beyond bulk single crystals: A data format for all materials structure–property–processing relationships, *MRS Bull.* 41(8), 617 (2016)
200. M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, Deep graph library: A graph-centric, highly-performant package for graph neural networks, arXiv: 1909.01315 (2019)
201. I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, et al., The DeepMind JAX Ecosystem, 2020
202. F. Chollet, et al., Keras, URL: github.com/fchollet/keras (2015)
203. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation in PYTORCH, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017
204. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, URL: www.tensorflow.org
205. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer), von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, arXiv: 1910.03771 (2019)
206. Openrefine: A free, open source, powerful tool for working with messy data, URL: openrefine.org, 2022
207. PyG-Team, PYG documentation, URL: pytorch-geometric.readthedocs.io/en/latest/, 2022
208. PytorchLightning, URL: www.pytorchlightning.ai, 2022
209. GitHub - Netflix/vectorflow, URL: github.com/Netflix/vectorflow, 2022
210. L. Biewald, Experiment tracking with weights and biases, URL: www.wandb.com, 2020
211. L. Himanen, M. O. Jäger, E. V. Morooka, F. F. Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, Dscribe: Library of descriptors for machine learning in materials science, *Comput. Phys. Commun.* 247, 106949 (2020)
212. W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, Open graph benchmark: Datasets for machine learning on graphs, *Advances in neural information processing systems* 33, 22118 (2020), arXiv: 2005.00687
213. O. Source, Rdkit: Open-source cheminformatics software, URL: www.rdkit.org, 2022
214. D. Grattarola, Spektral, URL: graphneural.network, 2022
215. S. Li, Y. Liu, D. Chen, Y. Jiang, Z. Nie, and F. Pan, Encoding the atomic structure for machine learning in materials science, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 12(1) (2022)
216. J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, Recent advances and applications of machine learning in solid-state materials science, *npj Comput. Mater.* 5(1), 83 (2019)
217. M. 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(5), 058301 (2012)
218. J. Schrier, Can one hear the shape of a molecule (from its Coulomb matrix eigenvalues), *J. Chem. Inf. Model.* 60(8), 3804 (2020)
219. M. McCarthy and K. L. K. Lee, Molecule identification with rotational spectroscopy and probabilistic deep learning, *J. Phys. Chem. A* 124(15), 3002 (2020)
220. F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, Crystal structure representations for machine learning models of formation energies, *Int. J. Quantum Chem.* 115(16), 1094 (2015)
221. K. 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(20), 205118 (2014)
222. J. Behler and M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, *Phys. Rev. Lett.* 98(14), 146401 (2007)
223. J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, *J. Chem. Phys.* 134(7), 074106 (2011)
224. A. Seko, A. Takahashi, and I. Tanaka, Sparse representation for a potential energy surface, *Phys. Rev. B* 90(2), 024101 (2014)
225. M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, WACSF — Weighted atom-centered symmetry functions as descriptors in machine learning potentials, *J. Chem. Phys.* 148(24), 241709 (2018)
226. A. P. Bartók, R. Kondor, and G. Csányi, On representing chemical environments, *Phys. Rev. B* 87(18), 184115 (2013)
227. C. W. Rosenbrock, E. R. Homer, G. Csányi, and G. L. W. Hart, Discovering the building blocks of atomic systems using machine learning: Application to grain boundaries, *npj Comput. Mater.* 3, 29 (2017)

228. F. M. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, and L. Emsley, Chemical shifts in molecular solids by machine learning, *Nat. Commun.* 9(1), 4501 (2018)
229. A. S. Rosen, S. M. Iyer, D. Ray, Z. Yao, A. Aspuru-Guzik, L. Gagliardi, J. M. Notestein, and R. Q. Snurr, Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery, *Matter* 4(5), 1578 (2021)
230. Z. Fan, Z. Zeng, C. Zhang, Y. Wang, K. Song, H. Dong, Y. Chen, and T. Ala-Nissila, Neuroevolution machine learning potentials: Combining high accuracy and low cost in atomistic simulations and application to heat transport, *Phys. Rev. B* 104(10), 104309 (2021)
231. Z. Mihalić and N. Trinajstić, A graph-theoretical approach to structure-property relationships, *J. Chem. Educ.* 69(9), 701 (1992)
232. O. Isayev, C. Oses, C. Toher, E. Gossett, S. Curtarolo, and A. Tropsha, Universal fragment descriptors for predicting properties of inorganic crystals, *Nat. Commun.* 8(1), 15679 (2017)
233. T. Xie and J. C. Grossman, Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, *Phys. Rev. Lett.* 120(14), 145301 (2018)
234. K. Xia and G. W. Wei, Persistent homology analysis of protein structure, flexibility and folding, *Int. J. Numer. Methods Biomed. Eng.* 30(8), 814 (2014)
235. Z. Cang, L. Mu, K. Wu, K. Opron, K. Xia, and G. W. Wei, A topological approach for protein classification, *Comput. Math. Biophys.* 3(1) (2015)
236. Y. Jiang, D. Chen, X. Chen, T. Li, G.-W. Wei, and F. Pan, Topological representations of crystalline compounds for the machine-learning prediction of materials properties, *npj Comput. Mater.* 7, 28 (2021)
237. E. Minamitani, T. Shiga, M. Kashiwagi, and I. Obayashi, Topological descriptor of thermal conductivity in amorphous Si, *J. Chem. Phys.* 156(24), 244502 (2022)
238. M. E. Aktas, E. Akbas, and A. E. Fatmaoui, Persistence homology of networks: Methods and applications, *Appl. Netw. Sci.* 4(1), 1 (2019)
239. A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, Insightful classification of crystal structures using deep learning, *Nat. Commun.* 9(1), 2775 (2018)
240. W. B. Park, J. Chung, J. Jung, K. Sohn, S. P. Singh, M. Pyo, N. Shin, and K. S. Sohn, Classification of crystal structure using a convolutional neural network, *IUCrJ* 4(4), 486 (2017)
241. Y. Zhang, X. He, Z. Chen, Q. Bai, A. M. Nolan, C. A. Roberts, D. Banerjee, T. Matsunaga, Y. Mo, and C. Ling, Unsupervised discovery of solid-state lithium ion conductors, *Nat. Commun.* 10(1), 5260 (2019)
242. S. C. Sieg, C. Suh, T. Schmidt, M. Stukowski, K. Rajan, and W. F. Maier, Principal component analysis of catalytic functions in the composition space of heterogeneous catalysts, *QSAR Comb. Sci.* 26(4), 528 (2007)
243. R. Tranås, O. M. Løvvik, O. Tomic, and K. Berland, Lattice thermal conductivity of half-Heuslers with density functional theory and machine learning: Enhancing predictivity by active sampling with principal component analysis, *Comput. Mater. Sci.* 202, 110938 (2022)
244. L. M. Ghiringhelli, J. Vybiral, E. Ahmetcik, R. Ouyang, S. V. Levchenko, C. Draxl, and M. Scheffler, Learning physical descriptors for materials science by compressed sensing, *New J. Phys.* 19(2), 023017 (2017)
245. R. Ouyang, S. Curtarolo, E. Ahmetcik, M. Scheffler, and L. M. Ghiringhelli, SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates, *Phys. Rev. Mater.* 2, 083802 (2018)
246. W. C. Lu, X. B. Ji, M. J. Li, L. Liu, B. H. Yue, and L. M. Zhang, Using support vector machine for materials design, *Adv. Manuf.* 1(2), 151 (2013)
247. Y. Wu, N. Prezhdo, and W. Chu, Increasing efficiency of nonadiabatic molecular dynamics by Hamiltonian interpolation with kernel ridge regression, *J. Phys. Chem. A* 125(41), 9191 (2021)
248. T. Hastie, R. Tibshirani, and J. H. Friedman, The elements of statistical learning: Data mining, inference, and prediction, 2nd Ed., in: Springer series in statistics, NY: Springer, 2009
249. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp 770–778
250. O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects, *Nat. Commun.* 12, 7273 (2021)
251. C. Zheng, C. Chen, Y. Chen, and S. P. Ong, Random forest models for accurate identification of coordination environments from X-ray absorption near-edge structure, *Patterns* 1(2), 100013 (2020)
252. J. J. Kranz, M. Kubillus, R. Ramakrishnan, O. A. von Lilienfeld, and M. Elstner, Generalized density-functional tight-binding repulsive potentials from unsupervised machine learning, *J. Chem. Theory Comput.* 14(5), 2341 (2018)
253. S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, and Y. Jung, Generative adversarial networks for crystal structure prediction, *ACS Cent. Sci.* 6(8), 1412 (2020)
254. J. Noh, J. Kim, H. S. Stein, B. Sanchez-Lengeling, J. M. Gregoire, A. Aspuru-Guzik, and Y. Jung, Inverse design of solid-state materials via a continuous representation, *Matter* 1(5), 1370 (2019)
255. M. L. Hutchinson, E. Antono, B. M. Gibbons, S. Paradiso, J. Ling, and B. Meredig, Overcoming data scarcity with transfer learning, arXiv: 1711.05099 (2017)
256. R. Chang, Y.-X. Wang, and E. Ertekin, Towards overcoming data scarcity in materials science: Unifying models and datasets with a mixture of experts framework, *npj Comput. Mater.* 8, 242 (2022)
257. M. A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015
258. A. Akbari, L. Ng, and B. Solnik, Drivers of economic and financial integration: A machine learning approach, *J. Empir. Finance* 61, 82 (2021)
259. L. Weng, Flow-based deep generative models, URL: lilianweng.github.io, 2018

260. P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist, Machine-learning-assisted materials discovery using failed experiments, *Nature* 533(7601), 73 (2016)
261. A. O. Oliynyk, L. A. Adutwum, J. J. Harynuk, and A. Mar, Classifying crystal structures of binary compounds AB through cluster resolution feature selection and support vector machine analysis, *Chem. Mater.* 28(18), 6672 (2016)
262. J. Tang, Q. Cai, and Y. Liu, Prediction of material mechanical properties with support vector machine, in: 2010 International Conference on Machine Vision and Human-machine Interface, Aprl 2010, pp 592–595
263. D. C. Elton, Z. Boukouvalas, M. S. Butrico, M. D. Fuge, and P. W. Chung, Applying machine learning techniques to predict the properties of energetic materials, *Sci. Rep.* 8(1), 9059 (2018)
264. D. Hu, Y. Xie, X. Li, L. Li, and Z. Lan, Inclusion of machine learning kernel ridge regression potential energy surfaces in on-the-fly nonadiabatic molecular dynamics simulation, *J. Phys. Chem. Lett.* 9(11), 2725 (2018)
265. K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks, *Nat. Commun.* 8(1), 13890 (2017)
266. D. Jha, L. Ward, A. Paul, W.-K. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, Elemnet: Deep learning the chemistry of materials from only elemental composition, *Sci. Rep.* 8, 17593 (2018)
267. D. Jha, L. Ward, Z. Yang, C. Wolverton, I. Foster, W. K. Liao, A. Choudhary, and A. Agrawal, IRNet: A general purpose deep residual regression framework for materials discovery, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 2385–2393, 2019
268. O. T. Unke and M. Meuwly, Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges, *J. Chem. Theo. Comput.* 15(6), 3678 (2019)
269. Z. Liu, L. Lin, Q. Jia, Z. Cheng, Y. Jiang, Y. Guo, and J. Ma, Transferable multilevel attention neural network for accurate prediction of quantum chemistry properties via multitask learning, *J. Chem. Inform. Model.* 61(3), 1066 (2021)
270. A. M. Krajewski, J. W. Siegel, J. Xu, and Z. K. Liu, Extensible structure-informed prediction of formation energy with improved accuracy and usability employing neural networks, *Comput. Mater. Sci.* 208, 111254 (2022)
271. K. T. Schütt, P. J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K. R. Müller, SchNet: A continuous-filter convolutional neural network for modeling quantum interactions, in Proceedings of the 31st International Conference on Neural Information Processing Systems, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp 992–1002
272. J. Jung, et al., Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis, *npj Comput. Mater.* 7, 96 (2021)
273. A. A. K. Farizhandi, O. Betancourt, and M. Mamivand, Deep learning approach for chemistry and processing history prediction from materials microstructure, *Sci. Rep.* 12(1), 4552 (2022)
274. T. Xie and J. C. Grossman, Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties, *Phys. Rev. Lett.* 120(14), 145301 (2018)
275. C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, Graph networks as a universal machine learning framework for molecules and crystals, *Chem. Mater.* 31(9), 3564 (2019)
276. S. Y. Louis, Y. Zhao, A. Nasiri, X. Wang, Y. Song, F. Liu, and J. Hu, Graph convolutional neural networks with global attention for improved materials property prediction, *Phys. Chem. Chem. Phys.* 22(32), 18141 (2020)
277. Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller, OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features, *J. Chem. Phys.* 153(12), 124111 (2020)
278. J. Gasteiger, J. Groß, and S. Günnemann, Directional message passing for molecular graphs, arXiv: 2003.03123 (2020)
279. K. Choudhary and B. DeCost, Atomistic line graph neural network for improved materials property predictions, *npj Comput. Mater.* 7(1), 185 (2021)
280. S. Zhang, Y. Liu, and L. Xie, Molecular mechanics-driven graph neural network with multiplex graph for molecular structures, arXiv: 2011.07457 (2020)
281. M. Ghorbani, S. Prasad, J. B. Klauda, and B. R. Brooks, GraphVAMPNet, using graph neural networks and variational approach to Markov processes for dynamical modeling of biomolecules, *J. Chem. Phys.* 156(18), 184103 (2022)
282. T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, and J. C. Grossman, Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials, *Nat. Commun.* 10(1), 2667 (2019)
283. S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, $E(3)$-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nat. Commun.* 13, 2453 (2022)
284. K. T. Schütt, O. T. Unke, and M. Gastegger, Equivariant message passing for the prediction of tensorial properties and molecular spectra, in: International Conference on Machine Learning, pp 9377–9388, 2021
285. Y. Jiang, Z. Yang, J. Guo, H. Li, Y. Liu, Y. Guo, M. Li, and X. Pu, Coupling complementary strategy to flexible graph neural network for quick discovery of coformer in diverse co-crystal materials, *Nat. Commun.* 12(1), 5950 (2021)
286. C. W. Park and C. Wolverton, Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery, *Phys. Rev. Mater.* 4(6), 063801 (2020)
287. G. P. Ren, Y. J. Yin, K. J. Wu, and Y. He, Force field-inspired molecular representation learning for property prediction, *J. Cheminform.* 15(1), 17 (2023)

288. C. Chen and S. P. Ong, AtomSets as a hierarchical transfer learning framework for small and large materials datasets, *npj Comput. Mater.* 7, 173 (2021)
289. H. Yamada, C. Liu, S. Wu, Y. Koyama, S. Ju, J. Shiomi, J. Morikawa, and R. Yoshida, Predicting materials properties with little data using shotgun transfer learning, *ACS Cent. Sci.* 5(10), 1717 (2019)
290. S. Feng, H. Fu, H. Zhou, Y. Wu, Z. Lu, and H. Dong, A general and transferable deep learning framework for predicting phase formation in materials, *npj Comput. Mater.* 7(1), 10 (2021)
291. V. Gupta, K. Choudhary, F. Tavazza, C. Campbell, W. K. Liao, A. Choudhary, and A. Agrawal, Cross-property deep transfer learning framework for enhanced predictive analytics on small materials data, *Nat. Commun.* 12, 6595 (2021)
292. V. Stanev, C. Oses, A. G. Kusne, E. Rodriguez, J. Paglione, S. Curtarolo, and I. Takeuchi, Machine learning modeling of superconducting critical temperature, *npj Comput. Mater.* 4(1), 29 (2018)
293. D. S. Palmer, N. M. O'Boyle, R. C. Glen, and J. B. O. Mitchell, Random forest models to predict aqueous solubility, *J. Chem. Inform. Model.* 47(1), 150 (2007)
294. P. Banerjee and R. Preissner, Bittersweetforest: A random forest based binary classifier to predict bitterness and sweetness of chemical compounds, *Front. Chem.* 6, 93 (2018)
295. P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist, Machine-learning-assisted materials discovery using failed experiments, *Nature* 533(7601), 7601 (2016)
296. L. Chen, B. Xu, J. Chen, K. Bi, C. Li, S. Lu, G. Hu, and Y. Lin, Ensemble-machine-learning-based correlation analysis of internal and band characteristics of thermoelectric materials, *J. Mater. Chem. C* 8(37), 13079 (2020)
297. J. Venderley, K. Mallayya, M. Matty, M. Krogstad, J. Ruff, G. Pleiss, V. Kishore, D. Mandrus, D. Phelan, L. Poudel, A. G. Wilson, K. Weinberger, P. Upreti, M. Norman, S. Rosenkranz, R. Osborn, and E. A. Kim, Harnessing interpretable and unsupervised machine learning to address big data from modern X-ray diffraction, *Proc. Natl. Acad. Sci. USA* 119(24), e2109665119 (2022)
298. R. Cohn and E. Holm, Unsupervised machine learning via transfer learning and k-means clustering to classify materials image data, *Integr. Mater. Manuf. Innov.* 10(2), 231 (2021)
299. R. E. A. Goodall and A. A. Lee, Predicting materials properties without crystal structure: Deep representation learning from stoichiometry, *Nat. Commun.* 11(1), 6280 (2020)
300. K. Muraoka, Y. Sada, D. Miyazaki, W. Chaikittisilp, and T. Okubo, Linking synthesis and structure descriptors from a large collection of synthetic records of zeolite materials, *Nat. Commun.* 10(1), 4459 (2019)
301. D. Jha, K. Choudhary, F. Tavazza, W. Liao, A. Choudhary, C. Campbell, and A. Agrawal, Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning, *Nat. Commun.* 10(1), 5316 (2019)
302. X. Zhong, B. Gallagher, S. Liu, B. Kailkhura, A. Hiszpanski, and T. Y.-J. Han, Explainable machine learning in materials science, *npj Comput. Mater.* 8, 204 (2022)
303. P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, Explainable AI: A review of machine learning interpretability methods, *Entropy (Basel)* 23(1), 18 (2020)
304. W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, Definitions, methods, and applications in interpretable machine learning, *Proc. Natl. Acad. Sci. USA* 116(44), 22071 (2019)
305. R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, and R. Asahi, Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics, *Acta Mater.* 141, 29 (2017)
306. K. Das, B. Samanta, P. Goyal, S.-C. Lee, S. Bhattacharjee, and N. Ganguly, CrysXPP: An explainable property predictor for crystalline materials, *npj Comput. Mater.* 8, 43 (2022)
307. A. Y. T. Wang, S. K. Kauwe, R. J. Murdock, and T. D. Sparks, Compositionally restricted attention-based network for materials property predictions, *npj Comput. Mater.* 7(1), 77 (2021)
308. A. Y. T. Wang, M. S. Mahmoud, M. Czasny, and A. Gurlo, CrabNet for explainable deep learning in materials science: bridging the gap between academia and industry, *Integr. Mater. Manuf. Innov.* 11(1), 41 (2022)
309. A. Parnami and M. Lee, Learning from few examples: A summary of approaches to few-shot learning, arXiv: 2203.04291 (2023)
310. Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, *ACM Comput. Surv.* 53(3), 63 (2020)
311. Y. Wang, A. Abuduweili, Q. Yao, and D. Dou, Property-aware relation networks for few-shot molecular property prediction, arXiv: 2107.07994 (2021)
312. Z. Guo, et al., Few-shot graph learning for molecular property prediction, in: Proceedings of the Web Conference 2021, in: WWW '21. New York, USA: Association for Computing Machinery, June 2021, pp 2559–2567
313. K. Kaufmann, H. Lane, X. Liu, and K. S. Vecchio, Efficient few-shot machine learning for classification of EBSD patterns, *Sci. Rep.* 11(1), 8172 (2021)
314. S. Akers, et al., Rapid and flexible segmentation of electron microscopy data using few-shot machine learning, *npj Comput. Mater.* 7, 187 (2021)
315. J. P. Perdew and K. Schmidt, Jacob's ladder of density functional approximations for the exchange-correlation energy, *AIP Conf. Proc.* 577, 1 (2001)
316. S. Dick and M. Fernandez-Serra, Machine learning accurate exchange and correlation functionals of the electronic density, *Nat. Commun.* 11(1), 3509 (2020)
317. R. Nagai, R. Akashi, and O. Sugino, Completing density functional theory by machine learning hidden messages from molecules, *npj Comput. Mater.* 6(1), 43 (2020)
318. J. Kirkpatrick, B. McMorrow, D. H. P. Turban, A. L. Gaunt, J. S. Spencer, A. G. D. G. Matthews, A. Obika, L. Thiry, M. Fortunato, D. Pfau, L. R. Castellanos, S. Petersen, A. W. R. Nelson, P. Kohli, P. Mori-Sánchez,

D. Hassabis, and A. J. Cohen, Pushing the frontiers of density functionals by solving the fractional electron problem, *Science* 374(6573), 1385 (2021)

319. J. C. Snyder, M. Rupp, K. Hansen, K. R. Müller, and K. Burke, Finding density functionals with machine learning, *Phys. Rev. Lett.* 108(25), 253002 (2012)
320. X. Lei and A. J. Medford, Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors, *Phys. Rev. Mater.* 3(6), 063801 (2019)
321. Z. Fan, Y. Wang, P. Ying, K. Song, J. Wang, Y. Wang, Z. Zeng, K. Xu, E. Lindgren, J. M. Rahm, A. J. Gabourie, J. Liu, H. Dong, J. Wu, Y. Chen, Z. Zhong, J. Sun, P. Erhart, Y. Su, and T. Ala-Nissila, GPUMD: A package for constructing accurate machine-learned potentials and performing highly efficient atomistic simulations, *J. Chem. Phys.* 157(11), 114801 (2022)
322. H. Wang, L. Zhang, J. Han, and W. E, DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics, *Comput. Phys. Commun.* 228, 178 (2018)
323. Y. Zhang, H. Wang, W. Chen, J. Zeng, L. Zhang, H. Wang, and W. E, DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models, *Comput. Phys. Commun.* 253, 107206 (2020)
324. P. Pattnaik, S. Raghunathan, T. Kalluri, P. Bhimalapuram, C. V. Jawahar, and U. D. Priyakumar, Machine learning for accurate force calculations in molecular dynamics simulations, *J. Phys. Chem. A* 124(34), 6954 (2020)
325. J. Westermayr and P. Marquetand, Machine learning and excited-state molecular dynamics, *Mach. Learn.: Sci. Technol.* 1(4), 043001 (2020)
326. G. Fan, A. McSloy, B. Aradi, C. Y. Yam, and T. Frauenheim, Obtaining electronic properties of molecules through combining density functional tight binding with machine learning, *J. Phys. Chem. Lett.* 13(43), 10132 (2022)
327. Z. Ahmad, T. Xie, C. Maheshwari, J. C. Grossman, and V. Viswanathan, Machine learning enabled computational screening of inorganic solid electrolytes for suppression of dendrite formation in lithium metal anodes, *ACS Cent. Sci.* 4(8), 996 (2018)
328. S. Gong, S. Wang, T. Zhu, X. Chen, Z. Yang, M. J. Buehler, Y. Shao-Horn, and J. C. Grossman, Screening and understanding Li adsorption on two-dimensional metallic materials by learning physics and physics-simplified learning, *JACS Au* 1(11), 1904 (2021)
329. T. Xie, A. France-Lanord, Y. Wang, J. Lopez, M. A. Stolberg, M. Hill, G. M. Leverick, R. Gomez-Bombarelli, J. A. Johnson, Y. Shao-Horn, and J. C. Grossman, Accelerating amorphous polymer electrolyte screening by learning to reduce errors in molecular dynamics simulated properties, *Nat. Commun.* 13(1), 3415 (2022)
330. K. Gubaev, E. V. Podryabinkin, G. L. Hart, and A. V. Shapeev, Accelerating high-throughput searches for new alloys with active learning of interatomic potentials, *Comput. Mater. Sci.* 156, 148 (2019)
331. T. Xie, X. Fu, O. E. Ganea, R. Barzilay, and T. Jaakkola, Crystal diffusion variational autoencoder for periodic material generation, arXiv: 2110.06197 (2021)
332. Y. Dong, D. Li, C. Zhang, C. Wu, H. Wang, M. Xin, J. Cheng, and J. Lin, Inverse design of two-dimensional graphene/h-BN hybrids by a regressional and conditional GAN, *Carbon* 169, 9 (2020)
333. Y. Pathak, K. S. Juneja, G. Varma, M. Ehara, and U. D. Priyakumar, Deep learning enabled inorganic material generator, *Phys. Chem. Chem. Phys.* 22(46), 26935 (2020)
334. Y. Suzuki, H. Hino, T. Hawai, K. Saito, M. Kotsugi, and K. Ono, Symmetry prediction and knowledge discovery from X-ray diffraction patterns using an interpretable machine learning approach, *Sci. Rep.* 10(1), 21790 (2020)
335. A. A. Enders, N. M. North, C. M. Fensore, J. Velez-Alvarez, and H. C. Allen, Functional group identification for FTIR spectra using image-based machine learning models, *Anal. Chem.* 93(28), 9711 (2021)
336. B. Huang, Z. Li, and J. Li, An artificial intelligence atomic force microscope enabled by machine learning, *Nanoscale* 10(45), 21320 (2018)
337. A. Chandrashekar, P. Belardinelli, M. A. Bessa, U. Staufer, and F. Alijani, Quantifying nanoscale forces using machine learning in dynamic atomic force microscopy, *Nanoscale Adv.* 4(9), 2134 (2022)
338. S. V. Kalinin, C. Ophus, P. M. Voyles, R. Erni, D. Kepaptsoglou, V. Grillo, A. R. Lupini, M. P. Oxley, E. Schwenker, M. K. Y. Chan, J. Etheridge, X. Li, G. G. D. Han, M. Ziatdinov, N. Shibata, and S. J. Pennycook, Machine learning in scanning transmission electron microscopy, *Nat. Rev. Methods Primers* 2(1), 11 (2022)
339. J. Jung, et al., Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis, *npj Comput. Mater.* 7, 96 (2021)
340. L. Floridi and M. Chiriatti, GPT-3: Its nature, scope, limits, and consequences, *Minds Mach.* 30(4), 681 (2020)
341. OpenAI, GPT-4 Technical Report, arXiv: 2303.08774 (2023)
342. D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, GPT-4 passes the bar exam, Rochester, NY, Mar. 15, 2023
343. V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, Unsupervised word embeddings capture latent knowledge from materials science literature, *Nature* 571(7763), 95 (2019)
344. E. A. Olivetti, J. M. Cole, E. Kim, O. Kononova, G. Ceder, T. Y.-J. Han, and A. M. Hiszpanski, Data-driven materials research enabled by natural language processing and information extraction, *Appl. Phys. Rev.* 7(4), 041317 (2020)
345. P. Shetty and R. Ramprasad, Automated knowledge extraction from polymer literature using natural language processing, *iScience* 24(1), 101922 (2021)
346. A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, M. Lackenby, G. Williamson, D.

Hassabis, and P. Kohli, Advancing mathematics by guiding human intuition with AI, *Nature* 600(7887), 70 (2021)

347. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics informed machine learning, *Nat. Rev. Phys.* 3(6), 422 (2021)
348. A. Goyal and Y. Bengio, Inductive biases for deep learning of higher-level cognition, *Proc. R. Soc. A* 478(2266), 20210068 (2022)
349. B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune, Video pretraining (VPT): Learning to act by watching unlabeled online videos, *Advances in Neural Information Processing Systems* 35, 24639 (2022)
350. J. Lehman, J. Gordon, S. Jain, K. Ndousse, C. Yeh, and K. O. Stanley, Evolution through large models, arXiv: 2206.08896 (2022)
351. M. S. Anis, et al., Qiskit: An open-source framework for quantum computing, 2021
352. C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, Communication-efficient federated learning via knowledge distillation, *Nat. Commun.* 13, 2032 (2022)
353. H. G. Yu, Neural network iterative diagonalization method to solve eigenvalue problems in quantum mechanics, *Phys. Chem. Chem. Phys.* 17(21), 14071 (2015)
354. S. K. Ghosh and D. Ghosh, Machine learning matrix product state Ansatz for strongly correlated systems, *J. Chem. Phys.* 158(6), 064108 (2023)
355. P. C. H. Nguyen, J. B. Choi, H. S. Udaykumar, and S. Baek, Challenges and opportunities for machine learning in multiscale computational modeling, *J. Comput. Inf. Sci. Eng.* 23(6), 060808 (2023)
356. H. Wahab, V. Jain, A. S. Tyrrell, M. A. Seas, L. Kotthoff, and P. A. Johnson, Machine-learning-assisted fabrication: Bayesian optimization of laser-induced graphene patterning using *in-situ* Raman analysis, *Carbon* 167, 609 (2020)
357. A. Tayyebi, A. S. Alshami, X. Yu, and E. Kolodka, Can machine learning methods guide gas separation membranes fabrication, *J. Membrane Sci. Lett.* 2(2), 100033 (2022)
358. Y. T. Chen, M. Duquesnoy, D. H. S. Tan, J. M. Doux, H. Yang, G. Deysher, P. Ridley, A. A. Franco, Y. S. Meng, and Z. Chen, Fabrication of high-quality thin solid-state electrolyte films assisted by machine learning, *ACS Energy Lett.* 6(4), 1639 (2021)
359. W. Li, L. Liang, S. Zhao, S. Zhang, and J. Xue, Fabrication of nanopores in a graphene sheet with heavy ions: A molecular dynamics study, *J. Appl. Phys.* 114(23), 234304 (2013)
360. L. L. Safina and J. A. Baimova, Molecular dynamics simulation of fabrication of Ni-graphene composite: Temperature effect, *Micro & Nano Lett.* 15(3), 176 (2020)
361. B. Zhao, C. Shen, H. Yan, J. Xie, X. Liu, Y. Dai, J. Zhang, J. Zheng, L. Wu, Y. Zhu, and Y. Jiang, Constructing uniform oxygen defect engineering on primary particle level for high-stability lithium-rich cathode materials, *Chem. Eng. J.* 465, 142928 (2023)
362. X. X. Liao, H. Q. Wang, and J. C. Zheng, Tuning the structural, electronic, and magnetic properties of strontium titanate through atomic design: A comparison between oxygen vacancies and nitrogen doping, *J. Am. Ceram. Soc.* 96(2), 538 (2013)
363. H. Xing, H. Q. Wang, T. Song, C. Li, Y. Dai, G. Fu, J. Kang, and J. C. Zheng, Electronic and thermal properties of Ag-doped single crystal zinc oxide via laser-induced technique, *Chin. Phys. B* 32(6), 066107 (2023)
364. L. Wu, J. C. Zheng, J. Zhou, Q. Li, J. Yang, and Y. Zhu, Nanostructures and defects in thermoelectric $AgPb_{18}SbTe_{20}$ single crystal, *J. Appl. Phys.* 105(9), 094317 (2009)
365. H. Zeng, M. Wu, H. Q. Wang, J. C. Zheng, and J. Kang, Tuning the magnetic and electronic properties of strontium titanate by carbon doping, *Front. Phys.* 16(4), 43501 (2021)
366. D. Li, H. Q. Wang, H. Zhou, Y. P. Li, Z. Huang, J. C. Zheng, J. O. Wang, H. Qian, K. Ibrahim, X. Chen, H. Zhan, Y. Zhou, and J. Kang, Influence of nitrogen and magnesium doping on the properties of ZnO films, *Chin. Phys. B* 25(7), 076105 (2016)
367. R. Wang and J. C. Zheng, Promising transition metal decorated borophene catalyst for water splitting, *RSC Advances* 13(14), 9678 (2023)
368. J. He, L. D. Zhao, J. C. Zheng, J. W. Doak, H. Wu, H. Q. Wang, Y. Lee, C. Wolverton, M. G. Kanatzidis, and V. P. Dravid, Role of sodium doping in lead chalcogenide thermoelectrics, *J. Am. Chem. Soc.* 135(12), 4624 (2013)
369. L. D. Cooley, A. J. Zambano, A. R. Moodenbaugh, R. F. Klie, J. C. Zheng, and Y. Zhu, Inversion of two-band superconductivity at the critical electron doping of $(Mg, Al)B_2$, *Phys. Rev. Lett.* 95(26), 267002 (2005)
370. H. Yan, T. Wang, L. Liu, T. Song, C. Li, L. Sun, L. Wu, J. C. Zheng, and Y. Dai, High voltage stable cycling of all-solid-state lithium metal batteries enabled by top-down direct fluorinated poly (ethylene oxide)-based electrolytes, *J. Power Sources* 557, 232559 (2023)
371. J. C. Zheng, C. H. A. Huan, A. T. S. Wee, R. Z. Wang, and Y. M. Zheng, Ground-state properties of cubic C-BN solid solutions, *J. Phys.: Condens. Matter* 11(3), 927 (1999)
372. Z. Huang, T. Y. Lü, H. Q. Wang, S. W. Yang, and J. C. Zheng, Electronic and thermoelectric properties of the group-III nitrides (BN, AlN and GaN) atomic sheets under biaxial strains, *Comput. Mater. Sci.* 130, 232 (2017)
373. T. Y. Lü, X. X. Liao, H. Q. Wang, and J. C. Zheng, Tuning the indirect–direct band gap transition of SiC, GeC and SnC monolayer in a graphene-like honeycomb structure by strain engineering: A quasiparticle GW study, *J. Mater. Chem.* 22(19), 10062 (2012)
374. J. C. Zheng and J. W. Davenport, Ferromagnetism and stability of half-metallic MnSb and MnBi in the strained zinc-blende structure: Predictions from full potential and pseudopotential calculations, *Phys. Rev. B* 69(14), 144415 (2004)
375. L. Xu, H. Q. Wang, and J. C. Zheng, Thermoelectric properties of PbTe, SnTe, and GeTe at high pressure: An *ab initio* study, *J. Electron. Mater.* 40(5), 641

(2011)

376. L. Xu, Y. Zheng, and J. C. Zheng, Thermoelectric transport properties of PbTe under pressure, *Phys. Rev. B* 82(19), 195102 (2010)

377. J. C. Zheng, Superhard hexagonal transition metal and its carbide and nitride:Os, OsC, and OsN, *Phys. Rev. B* 72(5), 052105 (2005)

378. T. Sun, K. Umemoto, Z. Wu, J. C. Zheng, and R. M. Wentzcovitch, Lattice dynamics and thermal equation of state of platinum, *Phys. Rev. B* 78(2), 024304 (2008)

379. Z. Wu, R. M. Wentzcovitch, K. Umemoto, B. Li, K. Hirose, and J. C. Zheng, Pressure-volume-temperature relations in MgO: An ultrahigh pressure-temperature scale for planetary sciences applications, *J. Geophys. Res.* 113(B6), B06204 (2008)

380. S. Deng, L. Wu, H. Cheng, J. C. Zheng, S. Cheng, J. Li, W. Wang, J. Shen, J. Tao, J. Zhu, and Y. Zhu, Charge-lattice coupling in hole-doped $LuFe_2O_{4+\delta}$: The origin of second-order modulation, *Phys. Rev. Lett.* 122(12), 126401 (2019)

381. J. C. Zheng, L. Wu, Y. Zhu, and J. W. Davenport, On the sensitivity of electron and X-ray scattering factors to valence charge distribution, *J. Appl. Crystall.* 38, 648 (2005)

382. J. C. Zheng and H. Q. Wang, Principles and applications of a comprehensive characterization method combining synchrotron radiation technology, transmission electron microscopy, and density functional theory, *Sci. Sin. - Phys. Mech. & Astron.* 51(3), 030007 (2021)