On machine learning force fields for metallic nanoparticles

Claudio Zeni, Kevin Rossi, Aldo Glielmo & Francesca Baletto

Published online: 12 Sep 2019.

Submit your article to this journal ↗

View related articles ↗

View Crossmark data ↗

Taylor & Francis
Taylor & Francis Group

REVIEW ARTICLE

On machine learning force fields for metallic nanoparticles

Claudio Zeni[a], Kevin Rossi[a,b], Aldo Glielmo[a] and Francesca Baletto[a]

[a]Department of Physics, King's College London, Strand, London, UK; [b]Laboratory of Computational Science and Modelling, Institut de Materiaux, École polytechnique fédérale de Lausanne, Lausanne, Switzerland1654919

ABSTRACT

Machine learning algorithms have recently emerged as a tool to generate force fields which display accuracies approaching the ones of the ab-initio calculations they are trained on, but are much faster to compute. The enhanced computational speed of machine learning force fields results key for modelling metallic nanoparticles, as their fluxionality and multi-funneled energy landscape needs to be sampled over long time scales. In this review, we first formally introduce the most commonly used machine learning algorithms for force field generation, briefly outlining their structure and properties. We then address the core issue of training database selection, reporting methodologies both already used and yet unused in literature. We finally report and discuss the recent literature regarding machine learning force fields to sample the energy landscape and study the catalytic activity of metallic nanoparticles.
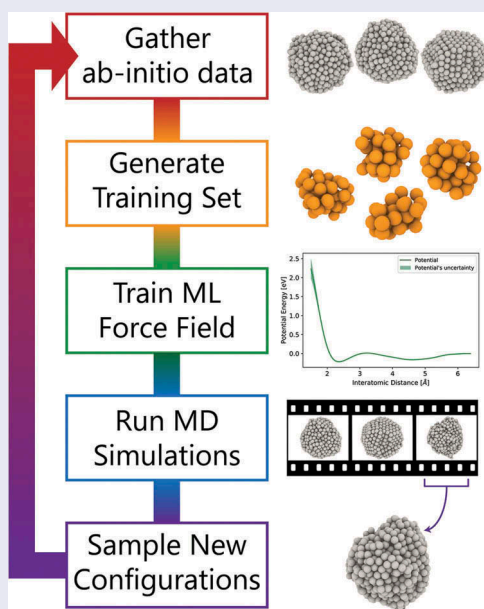
CONTACT Claudio Zeni ✉ claudio.zeni@kcl.ac.uk 🖥 Department of Physics, King's College London, Strand, London WC2R 2LS, UK

I. Introduction

Metallic nanoparticles (MNPs) are finite objects, containing up to several thousands of atoms. Being 0D, MNPs can display a very diverse and complex configurational space with many local minima, often very close in energy. [1–3] MNPs' chemical and physical properties – different from their single atom or bulk counterparts – depend strongly on the MNPs' architecture (i.e., their size, shape, chemical composition and chemical ordering). Therefore, harnessing the full potential of MNPs for practical applications in catalysis, plasmonics, optics, memory-storage, and biomedicine, hinges on the understanding and control of the dependence between the variety of accessible/available isomers over time and hence of MNPs' properties [4–8].

In the past two decades, many efforts have been carried out to sample the multi-funnelled energy landscape of MNPs, mainly to identify MNPs' putative global minima. [9] To this end, various sampling methods have been employed, such as Monte Carlo schemes [10], basin hopping [11–15], hyperspatial optimization [16], and evolutionary algorithms [17–21] to name a few.

While prediction from global minimum or archetypical structures may be helpful in deciphering structure–property relationship, only the accurate estimate of the finite-temperature population of relevant isomer enables to obtain accurate predictions of the ensemble properties. From a numerical point of view, molecular dynamics, path sampling [22,23], metadynamics [24–28], and finite-temperature partition function evaluation approaches [29–33] offer a dramatically effective way to account for structural changes and fluxional behaviours, and thus to estimate faithful population distributions. The evaluation of converged results is however expensive, i.e., it necessitates of thousands if not millions of energy and force calculations [27,31].

In the recent years, the advent of machine learning (ML) techniques paved the way towards establishing a novel paradigm in the computational studies of nanoparticles' structural and chemophysical properties. Together with big-data techniques that easily and better characterize nanoparticles' geometries during simulations [34–36] and experiments [37–39] machine learning force fields (ML-FFs) represent the most ground-breaking application of artificial intelligence models to further the breath and scope of numerical prediction and design of MNPs.

Machine learning force fields' (FFs) accuracy is comparable to the one of the electronic structure methods they are trained on, but their speed can approach the one of semi-empirical methods. Their use in algorithms to sample minima and transition states in nanoparticles' energy landscapes thus promises to reduce the computational cost of sampling schemes and

in turn to establish them as routine, comprehensive and exhaustive procedures that enable to bridge simulations and real systems used in experimental investigations.

The aim of this review is to offer a representative and pedagogical overview of the state-of-the-art of ML-FFs for MNPs, while also presenting a discussion on the open questions related to the subject. The review is organized as follows: in Section II the local atomic environment representations most commonly employed in ML-FF generation will be discussed and presented to the reader. In Section III three supervised ML algorithms for ML-FF generation will be formally introduced and briefly explained: linear regression, artificial neural networks (ANN), and Gaussian process regression (GPR). Section IV will discuss the issue of training database selection, presenting some methods that have been already used in literature together with algorithms that have not already been used in the field. Subsequently, Section V will present examples from literature on the application of ML to the study of nanoparticles, together with figures extracted from selected works. Finally, the conclusion will summarize the earlier presented discussion.

II. Descriptors of local atomic environments

A force field establishes a mapping between atomic coordinates and the corresponding total energy of the system. The construction of a force field can therefore be framed as a supervised learning problem where the objective is to find a function that yields a total energy E (and/or forces $\mathbf{F}_i$) given the coordinates and species of atoms in a system $\mathcal{R}$. The total energy $E(\mathcal{R})$ of the system $\mathcal{R}$ can be learnt as a function of the global coordinates of the system, but this leads to computational costs which scale non-linearly with the size of the system [40–42]. Therefore, in order to improve the computational scaling, and to construct force fields which can be used on systems with arbitrary number of atoms, the total energy $E(\mathcal{R})$ of a system can be approximated as a sum of local energy contributions $\epsilon(\rho_i)$, each pertaining to an atom i surrounded by its local atomic environment ρ_i: [43]

$$E(\mathcal{R}) \approx \sum_{i \in \mathcal{R}} \epsilon(\rho_i), \tag{1}$$

where the local atomic environment ρ_i refers to a vector or higher-dimensional object that contains information regarding the positions and atomic species of atoms contained in a sphere of radius r_c centred on atom i. The approximation of Equation (1) is based on the *near-sightedness* principle, i.e. on the fact that forces acting on atoms are not strongly dependant on the positions and species of atoms very far away. [44] The

near-sightedness assumption does not hold for interactions which are inherently long-ranged, i.e. electrostatic interactions; and these have to be treated separately.

The force vectors $\mathbf{F}_i$ can be calculated using the definition:

$$\mathbf{F}_i = -\frac{\partial E}{\partial \mathbf{r}_i} \tag{2}$$

where $\mathbf{r}_i$ is the vector containing the Cartesian coordinates of atom i.

A data set $\mathcal{D}$ composed of N input-output pairs $\mathcal{D} = \{\mathcal{R}_n, E_n\}$, which $n \in \{1, \cdots, N\}$ is required for supervised learning problems. A fraction of such data set is used to train the model (training set $\mathcal{D}_{tr}$), and the remaining part to test its accuracy (validation set $\mathcal{D}_{val}$).

We will now assume the learning problem to be focused on predicting local energies $\epsilon(\rho_i)$, since total energies and forces can be derived from local energies (Equations (1,2)). This assumption is made mainly for educational purposes, as machine learning force fields can be trained using total energies, forces, or a combination of the two.

In general, a local energy $\epsilon(\rho_i)$ of a given atom i is not learned directly as a function of the atomic Cartesian coordinates. On the contrary, the coordinates of the atoms in the vicinity of atom i (within a chosen cutoff r_c) are usually first transformed to a vector $\mathbf{q}_i$, commonly called a *descriptor*, and then a regression model is used to learn the map $\epsilon(\mathbf{q}_i)$ from the descriptor to the sought local energy function. This procedure is highly beneficial to the transferability of the generated force field as well as to the learning speed of the ML algorithm used, provided that the chosen descriptor possesses some key properties.

A. Descriptor's properties

A suitable representation of the local environment around an atom is encoded in a descriptor, which must possess the following key properties. Firstly, in order to strictly constrain the learned force field to respect fundamental physical invariances, any descriptor should be invariant upon rigid translations, rotations and reflections of the local environment, as well as invariant upon any permutation of atoms of the same chemical species. Arguably, invariance properties could be learned automatically by any sufficiently flexible algorithm when provided with enough data. Even so, their strict imposition through an invariant representation has been found to be extremely beneficial both for the learning speed and for the transferability of the force field [45–48]. Secondly, it is fundamental for a descriptor to be differentiable with respect to the atomic coordinates, as this is required for an analytic computation of the atomic forces after the

interpolation of an energy function. Thirdly, a descriptor should contain enough information to well capture the relevant physics of the system. As an illustrative example, the total mass of a local atomic environment ρ_i is a perfectly invariant function, but can hardly provide enough information for constructing a useful interaction model. Moreover, it would be desirable for a descriptor not only to contain enough information but also to capture, to some extent, important features of the sought energy or force function as this will make learning process faster and resulting force field potentially more accurate. Finally, in addition to all other properties, a descriptor should also be computationally inexpensive w.r.t. the reference method (e.g. DFT). Indeed, as an extreme example, the density functional theory (DFT) force predicted on a given atom fulfils all the conditions given above while still being a useless descriptor in practice.

The requirements listed above leave a large freedom of choosing a single descriptor among countless options. For this reason, many local atomic environment descriptors have been developed and applied to build ML-FFs [49–51], and in some occurrences such descriptors have been left for an ML algorithm to learn and optimize [52] In the following two paragraphs, we focus on the two descriptors that have been used the most for building atomistic force fields for nanoclusters: atomic symmetry functions [43] and power spectrum [53]. Afterwards, we will point out the common features possessed by these successful descriptors. Their remarkable similarity will then naturally bring to the exploration of another class of descriptors, simply encompassing explicit n-body degrees of freedom, which have recently been used successfully to build force fields for nanoclusters [54].

B. Atomic symmetry functions

The first descriptor used in the context of ML-FF fitting consists in a set of functions of the local environment which are invariant by construction over the physical symmetries mentioned earlier. These functions are called atomic symmetry functions (ASF) and, since they were introduced by Behler and Parrinello in their pioneer work on neural networks force fields [43], they are also often referred to as Behler-Parrinello functions. ASFs form a basis set for the expansion of two-body (radial) and three-body (angular) distribution functions [43]. Over the years, variations on the structure of ASFs have been developed, e.g. using Chebyshev polynomials to improve their spatial resolution's efficiency [55]. A brief description of their initial formulation [43] follows.

Given a local atomic environment ρ_i defined by the positions $\{\mathbf{r}_{ij}\}_{j=1}^{M}$ of M atoms within a cutoff radius r_c of a central atom located at the origin of the reference frame, the 2 and 3-body ASFs take the form, respectively:

$$G_2^i = \sum_{j \in \rho} e^{-\eta(r_{ij} - r_s)^2} f_c(r_{ij}) \tag{3}$$

$$G_3^i = \sum_{j \neq k \in \rho} 2^{1-\zeta}(1 + \lambda \cos \theta_{jk})^{\zeta} e^{-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)} f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk}). \tag{4}$$

where $f_c(r)$ is a cutoff function smoothly approaching zero as r_{ij} approaches the cutoff radius r_c. Its presence in the symmetry functions ensures that the descriptor smoothly varies when atoms transit through the radial cutoff, thus avoiding any discontinuity in the learned energy or force function.

Both G_2^i and G_3^i depend on some parameters, namely $\theta_2 = (\eta, r_s)$ and $\theta_3 = (\eta, \zeta, \lambda)$ (with $\lambda \in \{-1, 1\}$), and the descriptor vector $\mathbf{q}_{ASF}^i$ is built by evaluating these functions (Equations (3,4)) for a large number of these parameters. The ASF descriptor, therefore, requires a selection of relevant parameters for the θ_2 and θ_3; this can be achieved via fingerprint optimizations as recently discussed by Imbalzano et al. [56].

The function G_2^i can be seen to provide information about all the distances from the central atom to its M neighbours. On the other hand, G_3^i encompasses information on every triplet of atoms which include the central atom, meaning that they capture respectively two- and three-body information about the local environment ρ_i. It is important to note that using a two- or three-body descriptor does not limit the generated ML-FF to be low order, since an arbitrary non-linear function of such a descriptor will be able to model higher-order interactions as well [57].

The relative simplicity of implementation, the low computation cost, and perhaps the large body of research demonstrating the effectiveness of this descriptor, have made it the preferred choice for many practical applications in nanoparticle science [58–69].

C. Spherical harmonics power spectrum

The power spectrum of a spherical harmonics (SH) expansion is another popular choice for atomic descriptors. It was first introduced to fit energies with GPR [53,70], but has later found applications also for ANN FFs [71,72]. The power spectrum descriptor has been proven to be equivalent to the widely used 'Smooth Overlap of Atomic Orbitals' (SOAP) representation when using a dot product kernel within GPR. [53]

To build a power spectrum descriptor, it is customary to first write the local environment as a sum of Gaussian functions, each centred on one of the M atoms within the local environment ρ_i defined by the cutoff radius r_c:

$$\rho_i(\mathbf{r}) = \sum_{j \in \rho_i} e^{-(\mathbf{r}-\mathbf{r}_{ij})^2/2\sigma^2}. \tag{5}$$

Then, the key idea is that of expanding the angular part the above function in an SH basis $\{Y_{lm}(\hat{\mathbf{r}})\}$, in order to easily build a rotationally invariant descriptor as the power spectrum of this expansion. [53] Since SH can only retain the angular information of the above function (Equation (5)), the radial part is expanded in another basis, $g_n(r)$. The specific choice of the radial basis is not crucial to the descriptor, common choices are polynomials or equispaced Gaussian functions. Altogether, the function $\rho_i(\mathbf{r})$ can hence be formally rewritten as:

$$\rho_i(\mathbf{r}) = \sum_{n=0}^{\infty} \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_{nlm}^i g_n(r) Y_{lm}(\hat{\mathbf{r}}). \tag{6}$$

The coefficients c_{nlm}^i of the above SH expansion are not rotationally invariant. However, the power spectrum of the expansion can easily be written down in terms of the SH coefficients as:

$$p_{nn'l}^i = \frac{1}{\sqrt{2l+1}} \sum_{m=-l}^{l} c_{nlm}^i c_{n'lm}^{i*}. \tag{7}$$

The power spectrum descriptor $\mathbf{q}_{PS}^i$ is defined as the vector containing the power spectrum coefficients described above. It is rotationally invariant and can hence be effectively used to represent atomic environments. The power spectrum is more expensive to compute than the ASF descriptor, but it requires the user to choose only the width σ of the Gaussian functions in Equation (5) and where to truncate the expansion in Equation (6), instead of the full range of ASF parameters. Recently, a more efficient variant of the power spectrum descriptor which reduces its computational cost has been proposed. The interested reader is referred to Thompson2016 for more detailed information on such variant.

D. Explicit n-body features

In spite of the apparent differences between the mathematical expressions of the two descriptors discussed above, the power spectrum (Equation (7)) and the ASF representation (Equations (3) and (4)) are very similar at a fundamental level. Indeed, both can be considered as symmetric representations of the two- or three-body information present in the atomic environment. This is particularly clear in the case of the ASFs (Equation (4)), which are seen to depend rather explicitly on distances and angles. The n-body nature of the power spectrum descriptor is instead less obvious

from its mathematical formulation (Equations (6) and (7)), but it follows from the analysis reported in Ref [57–73]. later formulated also in Ref [74].–that the power spectrum is a three-body descriptor, meaning that any linear regression built on such a descriptor will give rise to a three-body force field. This fact can be intuitively understood considering that the Gaussian expansion in Equation (5) is made of contributions coming from pairs of atoms, and can thus be seen as a two-body descriptor. The power spectrum is then constructed by mixing rotationally invariant information coming from pairs of such two-body descriptors (Equation (7)), making the final descriptor three-body.

It is not a surprising fact that both of the descriptors mentioned so far (which are indeed among the most used in practical applications) are constructed using two- and/or three-body features. Low order descriptors can in fact be expected to capture well the ionic and covalent nature of chemical bonds, thus representing a rather natural choice. Moreover, while the absence of any angular information in a two-body descriptor will always prevent a correct characterization of higher-order interactions, a three-body descriptor does not present the same problem, meaning that a nonlinear function of a three-body descriptor (e.g., a neural network or a Gaussian process model) can in principle capture well also higher-order interactions.

Building on these ideas, one can think of explicitly using the two- and three-body degrees of freedom present in a given environment *directly* as descriptors. For instance, a two-body descriptor can be simply given by the *unordered* set of distances from the central atom to all other atoms within the configurations,

$$\mathbf{q}_2^i = \{r_{ij}\}_{j \in \rho_i}, \tag{8}$$

while a three-body descriptor can be given by the unordered set of triplets of distances between a central atom and any two neighbours

$$\mathbf{q}_3^i = \{(r_{ij}, r_{ik}, r_{jk})\}_{j,k \in \rho_i}. \tag{9}$$

The above descriptors contain by construction the full two- and three-body information about the local environment. They are moreover computationally efficient, simple to interpret, and do not require any choice of parameters or truncation approximation. These advantages come at a cost. Indeed, the requirement that the above sets are unordered is strictly needed to preserve permutational invariance, and such a condition must be imposed to the ML algorithm processing the inputs. This can be done rather straightforwardly when using a Gaussian process regression model (see Section III C), and in fact, explicit n-body features as the one provided

above have been extensively used in this framework [57,75,76], also for building force fields for nanoclusters [54].

III. Machine learning algorithms to generate force fields

The three most commonly used methods to learn the local energy function $\epsilon(\mathbf{q}_i)$ are linear regression, artificial neural networks and Gaussian process regression. These methods are schematically compared in Figure 1. In the following, these three algorithms are briefly presented.

A. Linear regression

A straightforward way to model the local energy function ϵ_i is via linear regression:

$$\hat{\epsilon}(\mathbf{q}_i) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{q}_i), \tag{10}$$

where $\mathbf{w}$ are the weights, $\hat{\epsilon}(\mathbf{q}_i)$ indicates the modelled energy function and $\boldsymbol{\phi}$ is a function chosen *a priori* (e.g. polynomial function, sinusoidal function, etc.). The weights that minimize the squared error loss L_2:

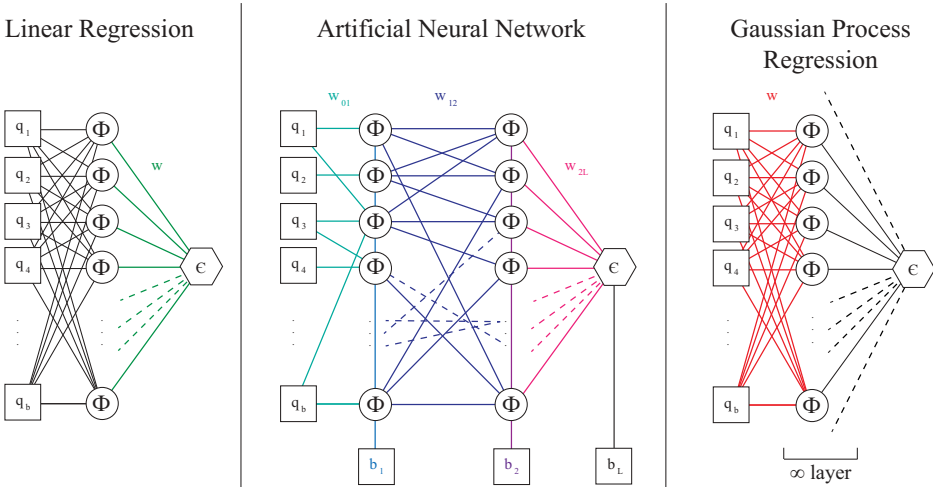$$L_2 = \sum_{n=1}^{N_{tr}} ||\hat{E}_n - E_n||^2 \tag{11}$$

Figure 1. Schematic comparison of linear regression, two-layer ANN, and GPR for the local energy prediction task. The symbols in the figure mirror the ones used in the main text. All three algorithms have been depicted in a similar manner so to ease the recognition of parallelisms and differences between the methods. Gaussian processes can be imagined to be equivalent to a fully connected ANN with a single infinite layer, this analogy has been proven rigorously in Refs. Neal [77] and Rasmussen [78].

can be found analytically. In Equation (11), the predicted and real total energies $\hat{E}_n$, E_n of each system are calculated per Equation (1), and N_{tr} indicates the number of training points used. In the linear regression case, the solution to the learning problem is fast to compute, and the predictions the model yields are computationally cheap. Despite the simplicity of the learning model, a variation of this method has been proven to yield interesting results, for example, in aiding the prediction of adsorption energies for RhAu nanoparticles. [79,80]

Nonetheless, in most applications, linear regression models lack the complexity needed to construct an accurate force field.

B. Artificial neural networks

a. Structure ANNs are composed of nodes organized in layers which connect the input $\mathbf{q}_i$ to an output $\epsilon(\mathbf{q}_i)$. Networks with few hidden layers are called 'shallow', whereas the term 'deep' is used to indicate networks with a high number of hidden layers. Nodes are connected by weights $\mathbf{w}$ which are optimized during training, and biases $\mathbf{b}$, also optimized, are added at every layer of the ANN.

ANNs are universal approximators [81,82], meaning that in the limit of an infinite hidden layer they can approximate any continuous function to arbitrary precision. ANNs have recently been used to predict total energies of MNP [63,71,72], study their thermodynamic stability [62], explore their phase diagram [63], aid the search for minima structures [60,67,69], and run molecular dynamics (MD) simulations [59,61,83]. The price to pay for such flexibility and expressive power is that ANNs are very data-hungry, and typically require orders of magnitude more training points than linear regression or GPR methods to reach the desired accuracy. In the specific case of an ANN with two hidden layers, the prediction for a local energy $\epsilon(\mathbf{q}_n)$ is given by:

$$\hat{\epsilon}(\mathbf{q}_n) = \phi\left\{ b_3 + \sum_{j_2} w_{23}^{j_2} \phi\left[b_2^{j_2} + \sum_{j_1} w_{12}^{j_1 j_2} \phi\left(b_1^{j_1} + \sum_i w_{01}^{i j_1} (\mathbf{q}_n)_i \right) \right] \right\},$$

(12)

where ϕ is an activation function (sigmoid, hyperbolic tangent, etc.), the j_l indices run over all nodes of layer l (here $l = 1, 2$), the $\mathbf{w}_{l,l'}$'s are the weights connecting the nodes of layer l to the nodes of layer l', $\mathbf{b}_l$ are the biases added to the nodes of layer l. The structure of the artificial neural network just described can be visualzsed in Figure 1 and it is easy to see how Equation (12) generalizes to networks containing more than two hidden layers.

b. Training The training of an ANN consists in the search of weights **w** and biases **b** which minimize a loss function L on a training set containing N_{tr} points. Typically, the squared error loss (Equation (11)) is chosen for regression problems such as energy and/or force fit. The parameter space $\{\mathbf{w}, \mathbf{b}\}$ contains from thousands to millions of parameters, and the optimization task is not trivial as the loss function is highly non-convex, presenting many local minima. For this reason, non-stochastic gradient descent on the loss landscape induced by the full training set would be non-optimal. Therefore, batch training is typically used to introduce stochasticity into the optimization: a subset of training points is selected and a number of gradient descent steps is taken; this process is then iterated. Typically, the training process is stopped once the error on the validation set starts increasing, indicating that the neural network has reached the point when it is starting to over-fit on the training data.

c. Beyond feed-forward artificial neural networks While only feed-forward ANNs are covered in this section, a plethora of variations on neural networks have been developed in recent years: recurrent neural networks, convolutional neural networks, variational autoencoders, and generative adversarial networks to name a few. [84] These methods share the same principles of weight and bias optimization via batch training, but contain differences (e.g. in layer structure) which substantially modify the tasks each method excels at. To the authors' knowledge, there is no existing literature where beyond-feed-forward neural networks are applied to the generation of force fields for MNPs specifically, but they have been employed for other systems [52,85].

C. Gaussian process regression

a. Formalism GPR is a Bayesian technique used for supervised learning tasks. Typically, GPR can require orders of magnitude less data points than ANNs to train, which makes it a suitable choice when data is expensive to generate and/or scarce. Another advantage of GPR techniques is that, being a fully Bayesian approach to regression, it is possible to estimate an uncertainty associated to every prediction.

As for the two previously described algorithms, here too the local energy function is learnt (instead of the total energy), and then summed according to Equation (1) to obtain the total energy. For educational purposes, in the following we assume that a database of local energies pertaining to local atomic environments is available. This is of course not the case in reality, and the interested reader is referred to Bartók and Csányi 70 and Glielmo *et al.* 86 for more details on how a local energy can be learnt using a database of total energies and forces.

GPR assumes that outputs are distributed like a Gaussian stochastic process. Given a training data set containing N_{tr} local atomic environment descriptors $\{\mathbf{q}_i\}$, $i = 1, \ldots, N_{tr}$, and a local energy *kernel* function $k(\mathbf{q}_i, \mathbf{q}_j)$, the prediction yielded by the trained GPR on a new local atomic environment descriptor $\mathbf{q}^*$ is:

$$\hat{\epsilon}(\mathbf{q}^*) = \sum_{i=1}^{N_{tr}} k(\mathbf{q}^*, \mathbf{q}_i) \alpha_i \tag{13}$$

where α_i are the weights computed during training via a straightforward matrix inversion. As already mentioned, we can also compute the uncertainty $\mathrm{VAR}(\mathbf{q}^*)$ associated with a prediction $\hat{\epsilon}(\mathbf{q}^*)$:

$$\mathrm{VAR}(\mathbf{q}^*) = k(\mathbf{q}^*, \mathbf{q}^*) + \sigma_n^2 - \mathbf{k}^\mathrm{T}(\mathbb{K} + \mathbb{I}\,\sigma_n^2)^{-1}\mathbf{k}, \tag{14}$$

where $\mathbf{k}$ indicates the vector containing the local energy kernel function evaluated between $\mathbf{q}^*$ and $\mathbf{q}_i$ for $i = 1, \cdots, N_{tr}$, $\mathbb{K}$ is the Gram matrix containing the kernel function evaluated between every pair of inputs $(\mathbf{q}_i, \mathbf{q}_j)$, with $i, j = 1, \cdots, N_{tr}$, and σ_n is a hyperparameter that governs the noise associated with the training data. This uncertainty (Equation (14)) is such that the error incurred by our GPR when predicting the total energy $\epsilon(\mathbf{q}^*)$ is expected to be normally distributed with mean zero and standard deviation $\sqrt{\mathrm{VAR}(\mathbf{q}^*)}$.

It is possible to learn from atomic forces instead of, or in conjunction with, total energies; this requires the construction of a kernel function for forces. In order to preserve energy conservation of the force field, such a force kernel function can be obtained by differentiation of an energy kernel function:

$$\mathbf{K}_{FF}(\mathbf{q}_i, \mathbf{q}_j) = \frac{\partial^2 k(\mathbf{q}_i, \mathbf{q}_j)}{\partial \mathbf{r}_i \partial \mathbf{r}_j^\mathrm{T}}, \tag{15}$$

where $\mathbf{r}_i, \mathbf{r}_j$ indicate the Cartesian coordinates of the central atom in the local atomic environments i and j respectively.

b. The kernel function The structure of the kernel function k is of great importance in GPR since its properties directly affect the properties of the force field it generates. The kernel function acts as a similarity function between pairs of descriptors of local atomic environments $(\mathbf{q}_i, \mathbf{q}_j)$. Kernel functions also allow for the use of descriptors which do not inherently posses all the required invariance properties, since these can be imposed on the kernel function itself. For instance, invariance over the rotations can be enforced by performing a Haar integration over the $O(3)$ group [47,57], invariance over translations can be enforced by integrating over $\mathbb{R}^3$, [74]

and invariance over permutations can be enforced by direct sum over the permutation group 57,76.

Some kernels commonly used in literature for ML-FF generation will now be described.

The SOAP energy-energy kernel k_{SOAP} can be straightforwardly built from the power spectrum descriptor derived in Equation (7): the dot product between normalized power spectrum descriptors of two local atomic environments $(\hat{\mathbf{q}}_{PS}^n, \hat{\mathbf{q}}_{PS}^m)$ is elevated to a power to obtain:

$$k_{SOAP}(\mathbf{q}_{PS}^n, \mathbf{q}_{PS}^m) = \left(\hat{\mathbf{q}}_{PS}^n \cdot \hat{\mathbf{q}}_{PS}^m\right)^\zeta \tag{16}$$

where ζ is a positive integer [53].

The n-body energy-energy kernel for any finite n can be built from the n-body explicit descriptors (Equation (8,9)) by, for example, taking the Gaussian difference of the descriptors summed over the relevant permutation groups and over each dimension of the descriptor. In the case of the two-body kernel, this results in:

$$k_2(\mathbf{q}_2^n, \mathbf{q}_2^m) = \sum_{i\in\rho_n}\sum_{j\in\rho_m} e^{-\frac{(r_{ni}-r_{mj})^2}{2\sigma^2}}, \tag{17}$$

where σ is the characteristic lengthscale of the kernel. The three-body kernel instead reads:

$$k_3(\mathbf{q}_3^n, \mathbf{q}_3^m) = \sum_{i>j\in\rho_n}\sum_{k>l\in\rho_m}\sum_{\mathbf{P}\in\mathcal{P}_c} e^{-\frac{\|(r_{ni},r_{nj},r_{ij})^{\mathrm{T}}-\mathbf{P}(r_{mk},r_{ml},r_{kl})^{\mathrm{T}}\|^2}{2\sigma^2}}, \tag{18}$$

where $\mathcal{P}_c$ $(|\mathcal{P}_c| = 6)$ is the set of permutations of three elements.

c. Mapping force fields One of the main drawbacks of GPR FFs is that, despite being computationally faster than *ab-initio* methods to evaluate, they scale linearly with the number of training points, N_{tr} (see Equation (13)). This burden can be avoided for a particular sub-class of kernel functions, and the computational cost of the so-derived mapped force fields (MFFs) is comparable to classical tabulated force fields. [54] The constraint that must be imposed on the kernel function in order for the FF to be mappable is that its order must be finite [57] or, in other words, that the kernel function is not many-body.

The mapping procedure starts with the identification of the maximum order of the kernel function used, for example two-body for kernels which only depend on interatomic distances from central atoms, three-body for kernels that use angles, four-body for kernels that use torsion angles, and so on. Subsequently, the local energy prediction made by the GP is calculated and stored on an array of values. For example, for a two-body kernel, the bond energy can be calculated and stored on an array of

distances between 1 Å and the cutoff radius r_c. This process yields a tabulated force field of the same order of the kernel. Predictions during MD runs can then be obtained by using an interpolator on the stored values for every pair, triplet, or quadruplet (when using, respectively, 2-, 3- or four-body kernels) of atoms in the system. This method has been shown to increase computational speed by a factor 10^4 with respect to traditional GPR without incurring in any additional accuracy loss. 54,57 The interested reader is directed to MFF, a Python package that implements the above method within the framework of GPR FFs: https://github.com/kcl-tscm/mff [87].

IV. Database selection

The accuracy of an ML-FF depends on whether the predictions are made in an extrapolation or interpolation region. Without giving a formal definition, it can be helpful to think of an interpolation region as that region of the descriptor space which is 'close' to the training database. Generally speaking, accuracy is higher, and predicted variance (in the case of GPR) is lower, when predictions are made in an interpolation regime. Therefore, it is preferable to always work in interpolation and, if possible, detect when and where the algorithm is extrapolating. Enforcing the force field to work in an interpolation regime becomes crucial in the case of nanoparticles and nanoalloys, where the number of competing isomers, even at small sizes, is very large. Due to this inherent complexity of the configuration landscape, it would be desirable to include in the initial training set a diverse array of isomers which yield accurate predictions also on never-seen-before geometries and/or chemical orderings. Two example cases extracted from the literature may help set this problem into practical scenarios.

In a study on Ni_{19}, it was shown that force prediction error was substantially lower if training and testing on the same morphology rather than training and testing on two different MNP morphologies [54]. Furthermore, training on structurally heterogeneous databases was shown to lead to a balanced trade-off between versatility and overall accuracy. By the same token, training on low-symmetry or defected structures, which present very different local atomic environments, resulted in a more accurate machine learning model. This is caused by the higher variety of local atomic environments present is low-symmetry morphologies, which allow the convex hull spanned by the training inputs to encompass a larger phase space.

In Figure 2 it can be seen how training on low-symmetry structures (4HCP) yields force fields that are more accurate on a target morphology w.r.t. training on high-symmetry structures (3HCP, DIH, BIP).
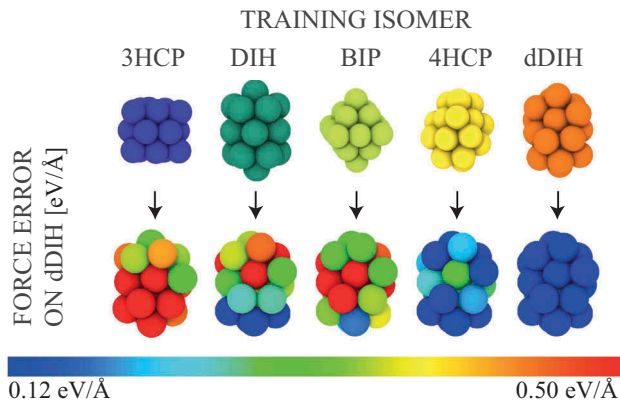
Figure 2. Mean absolute error on the DFT force vector incurred by a three-body GPR algorithm trained on five isomers of Ni_{19} and tested on the defected double icosahedron (dDIH) isomer. Figure adapted with permission from Zeni *et al*.[54].

In the case of a work focusing on AuRh nanoparticles, it was also reported that regression models predict binding energies of N, O, C molecules correctly or not depending on the size of the nanoparticles and surfaces in the training set. [79] Figure 3 shows how training sets which contained only single-crystal training points incur in high errors when predicting binding energies on nanoparticles, and vice-versa. On the other hand, training sets which contain both the nanoparticle and the
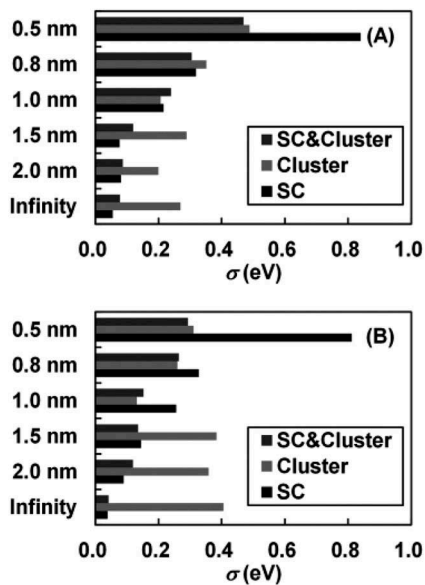

Figure 3. Mean absolute errors on the binding energies of O; the $Rh_{1-x}Au_x$ single crystals and particles are predicted using DFT data of single-crystal surfaces, small clusters, or both; using cutoff radii of (a) 6 and (b) 8 Å. Figure reprinted with permission from Jinnouchi and Asahi [79] . Copyright (2017) American Chemical Society.

single-crystal morphologies consistently incur in low binding energy mean absolute errors for MNPs ranging from diameter 0.5 nm to bulk.

A diverse set of strategies has therefore been developed in the machine learning community to build non-biased and automatically comprehensive training databases. These databases must be such that the machine learning force field generates predictions in an interpolation regime for the study of interest, and must assure that redundant, costly, information is not gathered for small regions of the phase space with respect to the investigation of interest.

Training database selection algorithms can be broadly classified into two categories: either the database needs to be generated ad-hoc (active learning), or it must be sub-sampled from a larger, already existent database (sparsification). In the following sections, we introduce the aforementioned concepts and extract some examples from literature to better inform the reader.

A. Active learning

If there is no sufficient pre-existent *ab-initio* training database for the task of interest, one convenient way to minimize the computational effort in building it is active learning. Active learning indicates a framework where the ML algorithm is iteratively trained on structures where the algorithm's predictions are deemed uncertain. If this is the case, forces and energies are computed with *ab-initio* methods on the structures where the ML predictions were not accurate enough, and finally inserted into the training database. While GPR can naturally return a predicted variance (Equation (14)), in the case of ANNs an uncertainty estimate can be obtained as the variance of the predictions made by multiple ANNs trained for the same task [58,88].

An example use of this approach can be found in conjunction with a genetic algorithm and a basin-hopping scheme in Jennings *et al.* [89]. Here, ML-FFs are used as a computationally inexpensive energy predictor so to fast-screen for energetically relevant isomers and thus facilitating the quick convergence of the optimization algorithms driving the energetic landscape exploration as well as the surrogate machine learning model training.

In a similar fashion, surrogate machine learning force fields can be also used to first quickly explore energetically relevant portions of the conformational space and rearrangement pathways via molecular dynamics and gather diverse ensembles of nanoparticle structures for which electronic structure calculations are executed. These are then fed as novel energy and force training points to the machine learning model in an iterative fashion until convergence of the model in the region of interest [64,67].

B. Database sparsification

If a database which is considered comprehensive for the system of interest is available, it can be useful to sub-sample a smaller training set from the full dataset, in order to reduce the computational cost of training and using an ML algorithm. This is especially beneficial in the case of GPR, where the training computational cost scales as $\mathcal{O}(N_{tr}^3)$ and the prediction cost scales with $\mathcal{O}(N_{tr})$, where N_{tr} is the number of training points. Many database sparsification methods exist, but up to now, only a few applications relative to the construction of ML-FFs can be found in the literature. Nonetheless, we will briefly present some database sparsification methods for ML-FFs, namely: farthest point sampling, measured error sampling, CUR decomposition, and descriptor-space sampling.

a. Farthest point sampling Farthest point sampling (FPS) is a database selection method where points are iteratively added to a training set based on their distance, according to a pre-selected metric, from the points already present in the training set. At each iteration, distances between the points inside and the points outside of the training set are computed, and the point furthest away from the training set, according to the chosen distance metric, is included. The computational cost of this method is $\mathcal{O}(N_{tr} \cdot N)$, where N is the total size of the database to sub-sample from. FPS sampling has been employed in literature to reduce training set size for Gaussian Process regression methods [90,91]. While using GPR, predicted variance can be used instead of distance from training set as a way to measure diversity. In this framework, points are iteratively added to the training set based on the value yielded by Equation (14).

b. Measured error sampling Another class of iterative methods, which can be used independently of the ML framework employed, is based on the measured error incurred by the ML algorithm. The starting training database contains a small number of randomly sampled data points, and is then progressively expanded by inclusion of data points on which the measured incurred error is maximum. This method should be effective in reducing the maximum error incurred on the global database at the cost of an increased computational complexity w.r.t. other similar methods, e.g. FPS.

c. CUR decomposition CUR decomposition is a matrix-approximation technique that decomposes a matrix $\mathbf{A}$ into a product of three matrices $\mathbf{C}$, $\mathbf{U}$, $\mathbf{R}$. [92] The method is similar to a low-rank single value decomposition approximation, but is built so that $\mathbf{C}$ is composed of columns of the original $\mathbf{A}$, and $\mathbf{R}$ from rows of the same $\mathbf{A}$. CUR decomposition can be applied to GPR by first building the Gram matrix $\mathbb{K}$ between all the points in the full dataset, and then reducing its rank by selecting the N columns (or rows) that compose $\mathbf{C}$ (or $\mathbf{R}$). The method has a computational scaling

of $\mathcal{O}(N^2)$; this can be reduced by subdividing the entire datasets into batches of size $N_{batch} \ll N$, thus introducing an approximation but reducing the scaling to $\mathcal{O}(N_{batch}^2 \cdot B)$ where B is the number of batches. Note that this method is not only helpful in identifying a sparse but information-rich database, but also in determining non-redundant sets of descriptors [56].

d. Descriptor-space sampling The following sampling method is inspired by the database sparsification algorithm used in Bernstein, Csányi, and Deringer [93], but more general in its formulation. First, a physical descriptor must be chosen, e.g. distance from, or angle w.r.t., the central atom of local atomic environments. Secondly, a set of discrete bins spanning the possible values the chosen descriptor can take must be built (e.g. a 1-D histogram containing distances from 1.5 Å to 5 Å binned every 0.05 Å). Local atomic environments in the total database are then shuffled and accepted into the training set if they contain at least one occurrence of a binned descriptor's value which is not already present in the training data set. This simple method assures that the final training set contains environments which are as diverse at possible in the descriptor's space. This method is very computationally efficient, as only a single pass over the database is required.

Figure 4 shows a comparison of the mean absolute error on force vectors of three of the aforementioned sampling methods when building an ML-FF using GPR on a database containing a total of 13,000 $(PtCu)_{13}$ structures, it is evident how descriptor-space sampling is the best performing algorithm for this system. For tests done with the measured error, predicted variance and CUR database selection methods, the total database was subdivided into batches and training sets were built by grouping the points selected in each of these batches. This was done in order to reduce the computational cost of such methods. It is interesting to note how, for
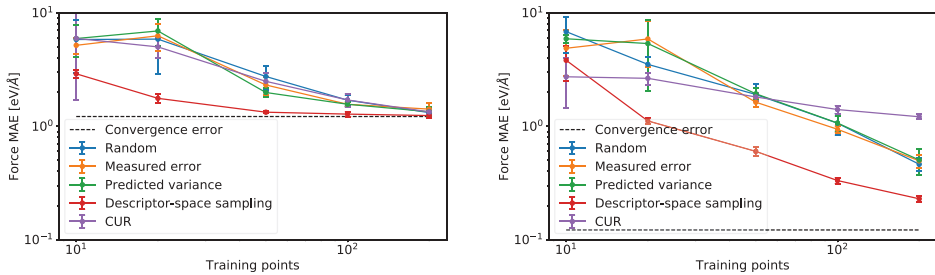

Figure 4. Log mean absolute error on force vectors incurred by a two-body (left) and a three-body (right) GPR algorithm as a function of the log number of training points N_{tr} when trained and tested on a total database of $(PtCu)_{13}$ local atomic environments, with $N = 13,000$. The training points have been selected according to the three algorithms described above, and also with random sampling. The black-dashed line indicates an estimate of the convergence error, obtained by training the algorithm on 2000 randomly selected training points. The error bars display the standard deviation of the error, obtained by repeating the process five times.

this particular system, GPR algorithms trained using databases selected by methods other than descriptor-space sampling incur in errors similar to the ones yielded by randomly chosen training datasets. Such poor performance could be caused by the batch approximation used. In any case, the descriptor-space sampling method has a better scaling than any other method presented, is overall faster to compute even on small datasets, and shows a good performance on the analysed dataset.

V. Applications of ML-FFs for metallic nanoparticles

After having introduced the formalisms behind the machine learning algorithms for generating fast and accurate energy predictions, we focus on their applications to understand structural properties of metallic nanoparticles. This section is divided into two parts. The first subsection reports applications where ML-FFs were used to supplement or substitute *ab initio* calculations for energy calculations, e.g. for global minima structure search or estimations and mapping of activation energies. The second subsection focuses on applications of ML-FFs to finite-temperature simulations over several tens of ns, carried out, e.g. to estimate phase diagrams and/or thermal properties of metallic nanoparticles.

A. ML-FFs for energy calculations

Machine learning force fields can be used as a surrogate for DFT when exploring the configurational space of nanoparticles. These ML-FFs must exhibit good accuracy especially when predicting energies, and can be trained either *a priori* or on-the-fly, depending on the size of the available database. In this section, we briefly resume the state-of-the-art on energy prediction yielded by ML-FFs to sample local minima, including chemical re-ordering, and to estimate adsorption energies.

   a. Monometallic nanoparticles The first application of machine learning to speed up the sampling of monometallic nanoparticles' energetic landscape consisted in the use of an ANN employing ASFs (Equations (3,4)) as descriptors to search for global minima of gold nanoclusters within the basin-hopping method by Ouyang, Xie and Jiang. A new putative global minimum which has a core-shell structure of $Au_{10}@Au_{48}$ and C4 symmetry was found, highlighting the benefit of comprehensive and fast exploration powered by neural network force fields [65].

   Subsequently, Zhai and Alexandrova employed a GPU accelerated ANN for sampling the potential energy surface of Pt_9 and Pt_{13}. The authors used a four-body descriptor as input to the ANN, and carried out global minima searches where the ML-FF was used to complement the DFT calculations and speed up the process. Finite-temperature effects were also included

a posteriori to probe the population distributions of such systems. This comprehensive study showed that the ensemble-averaged vertical ionization potential of the sytems under investigation changes when temperature increases, and that the catalytic property under *operando* conditions can be different from that evaluated at the global minimum structure. [94]

More recently, Sun and Sautet presented an application of a genetic algorithm coupled with a high-dimensional neural network potential using ASFs as descriptors to accelerate the comprehensive search for low-energy metastable structures of Pt_{13} under different H pressures. The presence and influence of these Pt_{13} structures during catalysis were discussed for hydrogen evolution reaction and methane activation. Although the ensemble of accessible metastable structures is relatively small under reaction condition, these structures can strongly influence the experimentally observed activity[68]. Figure 5 reports the lowest energy isomers found for Pt_{13} in the presence of H, together with the thermodynamic stability diagram of $Pt_{13}H_x$ clusters.

The same year, Kolsbjerg et al. looked at putative global minimum structures for Pt_{13} on a MgO support. The authors used an on-the-fly trained ANN force field to relax structures to local minima during evolutionary algorithm searches; here, ASFs were the descriptors of choice. The computational speed-up inherent to the framework most importantly enabled the screening of hundreds of kinetic rearrangement pathways connecting different low-energy conformers [67]. Figure 6 is shown to highlight the decrease in the number of *ab-initio* calculations performed thanks to the use of an ANN during the minima structure search.

b. Nanoalloys The computational speed-up offered by ML-FFs to global minima search is key in the study of nanoalloys, as the extra degree of freedom given by the number of possible homotops increases the dimensionality of the space that has to be explored.
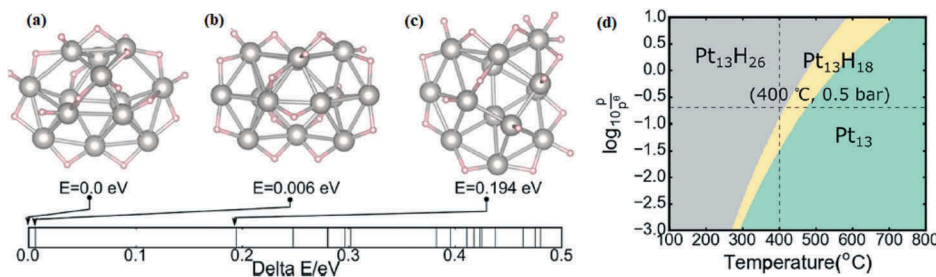

Figure 5. (a), (b), (c) Relative energies (eV) of the 20 $Pt_{13}H_{18}$ isomers in the low energy metastable ensemble (zero represents the energy of the global minimum) and structures of the three most stable ones. Grey spheres indicate Pt atoms while red ones indicate H atoms. (d) Thermodynamics stability of $Pt_{13}Hx$ clusters (x = 0, 18, 26) as a function of temperature and hydrogen pressure. Figure adapted with permission from Sun and Sautet [68] . Copyright (2018) American Chemical Society.
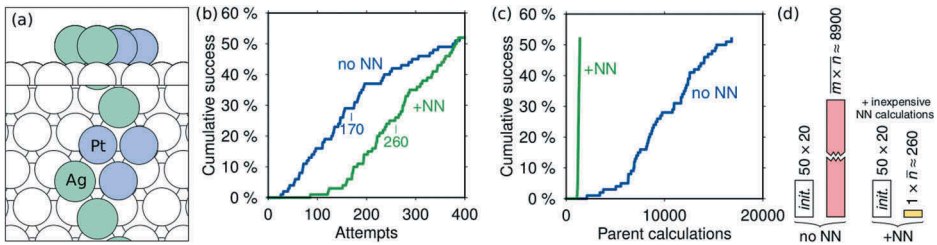
Figure 6. (a) The atomistic model of a Pt_3Ag_3 nanoparticle supported on a Pt(111). Ag and Pt atoms colored green and blue, respectively, the slab Pt atoms are white. (b) (c) The success rate of locating the global minimum as a function of the number of candidates evaluated and the number of needed parent calculations, respectively. (d) Graph highlighting the significant reduction in the average parent calculations needed. Figure reprinted with permission from Kolsbjerg, Peterson, and Hammer [67] . Copyright 2018 by the American Physical Society.

Jennings and co-workers employed GPR force fields using a built-for-purpose local atomic environment descriptor to search for stable, compositionally variant, geometrically similar PtAu nanoalloys. The machine learning approach yielded a 50-fold reduction in the number of required energy calculations compared to a traditional brute force genetic algorithm. [89] Figure 7 shows how the ML-based genetic algorithm was able to faithfully reproduce the convex hull for the excess energy of a $(PtAu)_{147}$ nanoparticle (left) while reducing by orders of magnitude the number of energy calculations required (right).
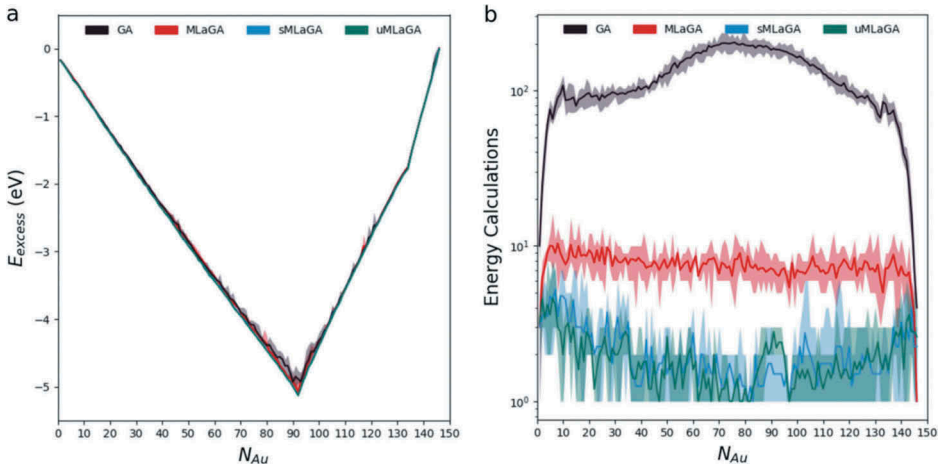

Figure 7. (a) Excess energy of a $(PtAu)_{147}$ nanoparticle as a function of its chemical composition, located with an ML-accelerated genetic algorithm employing effective-medium theory calculations. (b) Number of energy calculations as a function of the nanoparticle's composition. The four lines correspond to traditional genetic algorithms (GA), machine learning accelerated GA (MLaGA), serialized MLaGA and MLaGA utilizing uncertainty (uMLaGA); average and standard deviation over five searches are shown. Figure reprinted with permission from Jennings et al.[89].

One of the latest applications of machine learning force fields for nanoalloys involved the study of a trimetallic system, Cu–Pd–Ag. The fast and accurate ANN force field, powered by a stratified training scheme, was used to define a force field which, together with multi-tribe evolutionary searches, improved the efficiency in identifying stable elemental (30–80 atoms), binary (50, 55, and 80 atoms), and ternary (50, 55, and 80 atoms) Cu–Pd–Ag clusters. The best candidate structures identified with the neural network model, which used ASFs as local atomic environment descriptor, were showcased to have consistently lower energy at the density functional theory level compared with those found with searches employing an initial layer of inter-atomic potentials search [69].

c. Nanocatalysts Machine learning energy predictions not only aid the search for the ensemble of energetically relevant nanoparticles' isomers but also greatly enhance the scope for high throughput probing of active sites properties for catalytic reactions.

One of the first applications of machine learning algorithms to speed up the high throughput characterization of the adsoprtion properties of available sites in a nanoparticle was developed by Ulissi et al. Active sites for every stable low-index facet of a NiGa bimetallic crystal were enumerated and catalogued while the activity of these sites with respect to CO_2 adsorption was explored using a neural network-based surrogate model. This approach, which used ASFs as the local atomic environment descriptor of choice, reduced the number of explicit DFT calculations required for activity estimates by an order of magnitude. While most facets had similar activity to Ni surfaces, a few exposed Ni sites showed a very favourable on-top CO configuration [64]. Figure 8 reports the training scheme adopted in their work, along with a scatter plot displaying the increasing accuracy of the ANN model for the CO adsorption energies.

Jinnouchi et al. used linear regression in conjunction with the SOAP similarity function (Equation (17)) to interrogate catalytic activities for the direct NO decomposition on RhAu alloy nanoparticles. The employed method efficiently and accurately predicted the energetics of catalytic reactions on nanoparticles while providing information on structure–property relationships when combined with kinetic analysis [79].

The same authors later predicted the binding energies of N, O, and NO with RhAu surfaces and particles using the same approach. Kinetic analyses of the direct decomposition of NO on RhAu nanoparticles were carried on to demonstrate that catalytic activity increases with a decrease in the particle diameter to 2.0 nm. Below a diameter of 1.5 nm, a drop in the catalytic activity is registered and rationalized in terms of the disappearance of active-alloyed corner sites on the small nanoparticles [80].

AuCu nanoalloys' adsorption properties have also been investigated, as reported by Jäger et al. Potential energy scans of hydrogen on AuCu
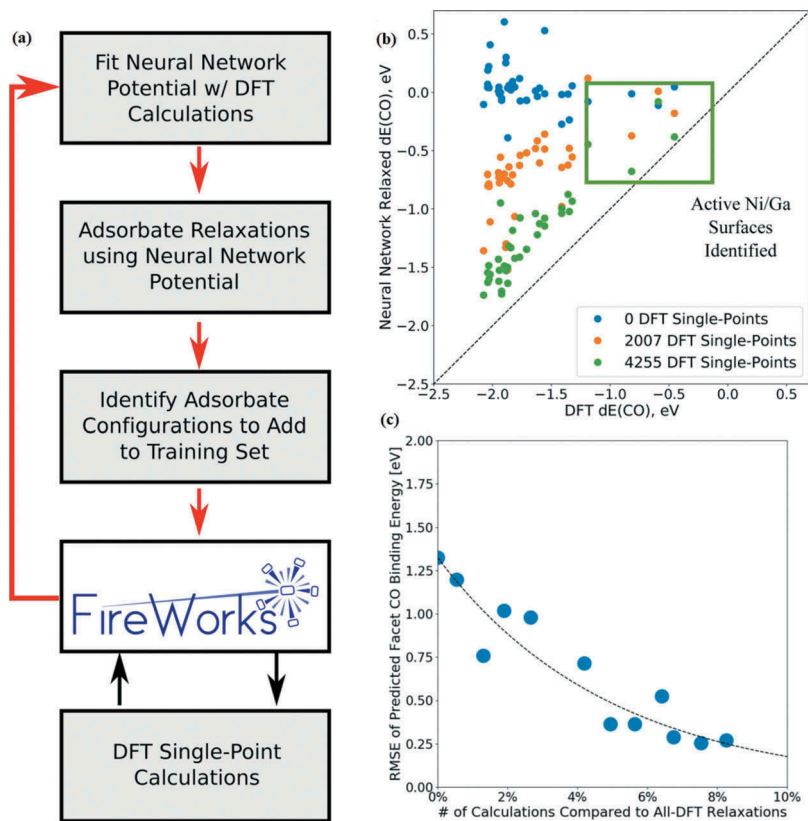
Figure 8. (a) Scheme used for training and use of the model. New training data is acquired via DFT single-point calculations. (b) Scatter plot for three iterations of the convergence system, starting from very poor predictions and converging to more accurate predictions of adsorption energy. (c) Convergence of the accuracy of the CO adsorption energies with respect to the training set size. Points possess some inherent noise due to the stochastic nature of the neural network training algorithms. Figure adapted with permission from Ulissi *et al*. [64]. Copyright (2017) American Chemical Society.

clusters and on MoS_2 surfaces were conducted to compare and assess the accuracy of the Smooth Overlap of Atomic Positions, Many-Body Tensor Representation [95], Couloumb matrix [96] and Atom-Centered Symmetry Functions in a kernel ridge regression framework. [97]

B. ML-FFs for finite-temperature simulations

Beyond sampling local minima in the energetic landscape of MNPs, MD-based studies of kinetic rearrangements and thermodynamic stability of MNPs are also of great interest. [1] With this in mind, employing ML-FFs to run fast MD simulations becomes more and more enticing. At the nanoscale, this allows to predict the dynamical and thermodynamical

properties of MNPs. In this section, we discuss the state-of-the-art of ML-FFs for MD simulations of MNPs and nanoalloys.

a. Monometallic nanoparticles Artrith et al. investigated the structural and energetic properties of copper clusters when supported on zinc oxide in the first-ever application of neural network force fields for metallic nanoparticles. In their seminal work, the authors assess the accuracy of ANN potentials employing the ASF descriptors. The manuscript builds on top of a previous paper by Artrith and Behler where a defected Cu surface was used as a benchmark for ANN force fields [58] providing an estimation of the accuracy and speed possible when using ML-FFs for metallic nanoparticles. The training of heterogeneous ensembles of structures was demonstrated to be transferable to complex large-scale simulations of several defected stepped surfaces and nanoparticles, such as the one displayed in Figure 9.[59]

ANN force fields developed in the group of Bulusu, which used the ASF descriptor, were employed to sample local and global minima in the potential energy surface of Na nanoparticles of size 16–40, where transitions were also thoroughly probed using a Monte Carlo scheme. The accuracy of the force field, and the timescale probed, allowed to establish
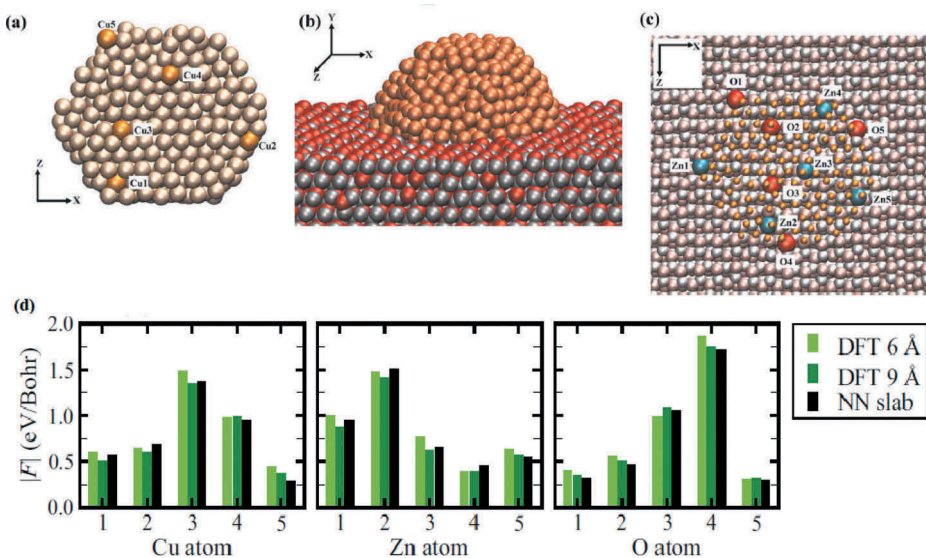

Figure 9. (a) Bottom-view of a snapshot of an MD simulation at 1000 K of a Cu_{612} cluster at the ZnO(1010) surface; five copper atoms have been selected to compare the ANN-predicted forces with the DFT forces (in d). (b) Side view of the cluster. (c) A top view of the ZnO(1010) surface is shown. Five oxygen and five zinc atoms have been chosen for a closer investigation of the forces (in (d)). (d) Comparison of the force modulus of two DFT force evaluations using atoms within 6 Å and 9 Å from the central one, and the neural network force field on the whole slab for the atoms highlighted in (a) and (c). Figure adapted with permission from Artrith, Hiller, and Behler [59] . Copyright 2013 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.

the presence of a characteristic premelting peak in the heat capacity curve, preceding a main melting peak, for clusters in the 20–40 atoms size range, [62], corroborating the observations of stepwise melting in small Na clusters first reported by Aguado and coworkers [98].

The same group later studied Au nanoclusters' energetic and thermodynamic properties. ANN force fields were employed to probe the potential energy landscape and thermodynamical behaviour of Au_{17}, Au_{34}, Au_{58}. Here too, ASFs were the local atomic descriptors of choice to generate inputs for the ANN. Canonical and microcanonical molecular dynamics sampling was performed for a total simulation time of around 3 ns for each nanoparticle. The study used such data to demonstrate the presence of a dynamical coexistence of solid-like and liquid-like phases near melting transition. The investigation further encompassed the estimation of the probability at finite temperatures for a set of isomers lying less than 0.5 eV from the global minimum structure. For Au_{34}, in particular, the global minimum structure resulted far from being the most dominant structure, even at low temperatures. [63]

Later, ANN force fields trained by first-principles density functional theory total energies were applied to search for global minima of gold nanoclusters within the basin-hopping method. In this case, the same authors decided to employ a descriptor based on the spherical harmonics expansion (Equation (6)), reporting its increased efficiency w.r.t. ASFs. A study on the fluxionality in Au_{147} was performed, and it was concluded that the system presents a dynamic surface. Such observation was concluded to be highly relevant in understanding reaction dynamics catalysed by Au nanoparticles. The putative global minimum of Au_{147} found by the authors using an ANN force field is reported in Figure 10, alongside the perfect icosahedron structure. [71]
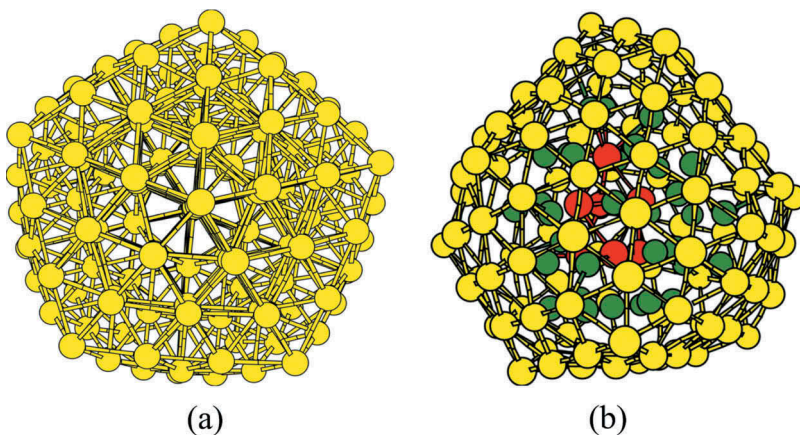

Figure 10. (a) Geometry of an Au_{147} icosahedron. **(b)** Geometry of the putative ground minimum for Au_{147} found using an ANN force field. Figure (b) is colour coded so to highlight the three shells of Au atoms. Figure reprinted from Jindal, Chiriki, and Bulusu [71], with the permission of AIP Publishing.

The use of Gaussian process regression with 2-body, 3-body, and many-body descriptors (Equations (8,9)) and kernel functions (Equations (18,19)) was instead reported for the first time by Zeni et al., which modelled interatomic forces in Ni_{19} nanoclusters. Thermodynamical properties as the melting point were probed thoroughly as the cost of the simulations carried out was 100,000 times lower than DFT calculations. Fluxionality at temperature below melting was observed along such timescales[54].

b. Nanoalloys Also for the case of nanoalloys, machine learning force fields represent a highly helpful technology enabling thorough assessment of energetic and thermodynamic properties.

The first application of neural networks to build force fields of nanoalloys was carried on by Arthrith and Behler. They employed shallow artificial neural networks with Behler-Parrinello symmetry function descriptors (Equation (4)) for the prediction of the composition and atomic ordering equilibrium architecture of AuCu alloy nanoparticles. Site-based Monte Carlo simulations were used to sample the composition space while molecular dynamics simulations simultaneously enabled to sample the structure space. An extensive set of equilibrium properties for many temperatures and chemical potentials were thus assessed. Consistent with previous studies, the most stable structures were found to exhibit Cu(core)Au(shell) configurations. Temperature-dependent favourable alloy arrangement was also observed, with enhanced Au concentration at the particle core for increasing temperatures. [61]

Other alloys of Au were investigated by the groups of Bulusu, by means of neural network force fields with ASF as descriptors to predict global minimum structures of $(AgAu)_{55}$ nanoalloys across different compositions. Pure Au and Au rich compositions minima resulted lower in energy compared to previous reports. Thermodynamical and energetic properties were also thoroughly assessed (c-T phase diagram, surface area, surface charge, probability of isomers, and Landau free energies) to rationalize the enhancement of the catalytic property of AgAu nanoalloys by incorporation of Ag up to 24 by composition in AgAu nanoparticles. This result was found to match previous experimental data. [66] The development of novel methods for fast and accurate force and energy calculations is of even greater importance for the study of nanoparticles in conditions closer to the *operando* one.

High-dimensional neural network force fields using the ASFs were also incorporated with Monte Carlo and MD simulations by Kang et al. to identify not only active but also electrochemically stable PtNiCu nanocatalysts for oxygen reduction reaction in acidic solution. The computationally efficient and precise approach proposed a promising oxygen reduction

reaction candidate: a 2.6 nm diameter icosahedron comprising a 60% of Pt and a remaining equal mixture of Ni and Cu. [83]

VI. Conclusions

In this review, we showcase how machine learning force fields offer the possibility to predict energies and forces with accuracy close to *ab-initio* methods, but at a much lower computational cost. The approaches to building machine learning force fields are multiple, and this review presents the major algorithms used so far in literature: linear regression, artificial neural networks and Gaussian process regression. Their advent and exploitation allow to tackle the complexity of the energy landscape of metallic nanoparticles, even in the case of *operando* conditions, for example, when an oxide substrate is included and explicitly modelled. Accurate but expensive algorithms to sample the free and potential energy landscapes of metallic nanoparticles (e.g. molecular dynamics, metadynamics, transition path sampling, basin hopping, harmonic approaches, nested sampling) will all greatly benefit from the deployment of machine learning-derived fast and accurate force fields. Particular care must be placed on training database selection to ensure that the relevant parts of phase space are included without redundancy. For this reason, a section of this review focuses entirely on describing the major algorithms for database selection.

The core of this review is, however, to provide the state-of-the-art of machine learning force fields to model metallic nanoparticles and nanoalloys. Inspired by the ongoing process of designing optimal metallic nanoparticles for target applications, and moving towards a numerical driven search, accurate estimates of how structural changes affect metallic nanoparticles' properties are in high need. To achieve this result, fast and accurate force fields, that allow the exploration of long time scales without the caveat of fitted parameters, are required.

The authors would like to dedicate this review to their mentor, colleague, and friend prof. Alessandro "Sandro" De Vita, passed away too early.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

References

[1] Baletto F, Ferrando R. Structural properties of nanoclusters: energetic, thermodynamic, and kinetic effects. Rev Mod Phys. 2005;77:371.

[2] Wang Z, Palmer RE. Experimental Evidence for Fluctuating, Chiral-Type Au55 Clusters by Direct Atomic Imaging. Nano letters. 2012;12:5510.

[3] Foster DM, Ferrando R, Palmer RE. Experimental determination of the energy difference between competing isomers of deposited, size-selected gold nanoclusters. Nat Commun. 2018;9:1323.

[4] Sun CQ. Size dependence of nanostructures: Impact of bond order deficiency. Prog Solid State Chem. 2007;35:1.

[5] Rossi K, Asara GG, Baletto F. A genomic characterisation of monometallic nanoparticles. Phys Chem Chem Phys. 2019;21:4888.

[6] Noguez C. Surface plasmons on metal nanoparticles: the influence of shape and physical environment. J Phys Chem C. 2007;111:3806.

[7] Cuenya BR. Synthesis and catalytic properties of metal nanoparticles: size, shape, support, composition, and oxidation state effects. Thin Solid Films. 2010;518:3127.

[8] Giljohann D, Seferos D, Daniel W, et al. Biocatalytic desymmetrization of an atropisomer with both an enantioselective oxidase and ketoreductases. Angew Chem. 2010;49:3280.

[9] Ferrando R, Jellinek J, Johnston RL. Nanoalloys: from theory to applications of alloy clusters and nanoparticles. Chem Rev. 2008;108:846.

[10] Rahm JM, Erhart P. Beyond magic numbers: atomic scale equilibrium nanoparticle shapes for any size. Nano Lett. 2017;17:5775.

[11] Li Z, Scheraga HA. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. Proc Nat Acad Sci. 1987;84:6611.

[12] Doye JPK, Wales DJ. Global minima for transition metal clusters described by sutton–chen potentials. New J Chem. 1998;22:733.

[13] Rossi G, Ferrando R. Searching for low-energy structures of nanoparticles: a comparison of different methods and algorithms. J Phys. 2009;21:084208.

[14] Barcaro G, Fortunelli A, Rossi G, et al. Electronic and structural shell closure in AgCu and AuCu nanoclusters. J Phys Chem B. 2006;110:23197.

[15] Calvo F, Schebarchov D, Wales DJ. Grand and semigrand canonical basin-hopping. J Chem Theory Comput. 2016;12:902.

[16] Pickard CJ. Hyperspatial optimization of structures. Phys Rev B. 2019;054102:1.

[17] Johnston RL. Evolving better nanoparticles: genetic algorithms for optimising cluster geometries. Dalton Trans. 2003;4193.

[18] Heiles S, Johnston RL. Global optimization of clusters using electronic structure methods. Int J Quantum Chem. 2013;113:2091.

[19] Lazauskas T, Sokol AA, Woodley SM. An efficient genetic algorithm for structure prediction at the nanoscale. Nanoscale. 2017;9:3850.

[20] Lv J, Wang Y, Zhu L, et al. Particle-swarm structure prediction on clusters. J Chem Phys. 2012;137:084104.

[21] Lepeshkin SV, Baturin VS, Uspenskii YA, et al. Method for simultaneous prediction of atomic structure and stability of nanoclusters in a wide area of compositions. J Phys Chem Lett. 2018;10:102.

[22] Trygubenko SA, Wales DJ. Kinetic analysis of discrete path sampling stationary point databases. Mol Phys ISSN. 2006;104:1497.

[23] Koslover EF, Wales JD. Comparison of double-ended transition state search methods. J Chem Phys. 2017;12.

[24] Laio A, Parrinello M. Escaping free-energy minima. Proc Nat Acad Sci. 2002;99:12562.

[25] Laio A, Gervasio FL. Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. Rep Prog Phys. 2008;71:126601.

[26] Pavan L, Rossi K, Baletto F. Metallic nanoparticles meet metadynamics. J Chem Phys. 2015;143:184304.

[27] Gould AL, Rossi K, Catlow CRA, et al. Controlling structural transitions in AuAg nanoparticles through precise compositional design. J Phys Chem Lett. 2016;7:4414.

[28] Rossi K, Baletto F. The effect of chemical ordering and lattice mismatch on structural transitions in phase segregating nanoalloys. Phys Chem Chem Phys. 2017;19:11057.

[29] Skilling J. Nested sampling for general Bayesian computation. Bayesian Anal. 2006;1:833.

[30] Partay LB, Bartok AP, Csanyi G. Efficient sampling of atomic configurational spaces. J Phys Chem A. 2010;114:10502.

[31] Rossi K, Pártay LB, Csányi G, et al. Thermodynamics of CuPt nanoalloys. Sci Rep. 2018;8:9150.

[32] Dorrell J, Pártay LB. Thermodynamics and the potential energy landscape: case study of small water clusters. Phys Chem Chem Phys. 2019;21:7305–7312.

[33] Doye JPK, Wales DJ. On potential energy surfaces and relaxation to the global minimum. J Chem Phys. 1996;8428.

[34] Fernandez M, Barnard AS. Identification of Nanoparticle Prototypes and Archetypes. ACS Nano. 2015;9:11980.

[35] Yan T, Sun B, Barnard AS. Predicting archetypal nanoparticle shapes using a combination of thermodynamic theory and machine learning. Nanoscale. 2018;10:21818.

[36] Gasparotto P, Meissner RH, Ceriotti M. Recognizing local and global structural motifs at the atomic scale. J Chem Theory Comput. 2018;14:486.

[37] Timoshenko J, Lu D, Lin Y, et al. Supervised Machine-Learning-Based Determination of Three-Dimensional Structure of Metallic Nanoparticles. J Phys Chem Lett. 2017;8:5091.

[38] Timoshenko J, Wrasman CJ, Luneau M, et al. Probing atomic distributions in mono- and bimetallic nanoparticles by supervised machine learning. Nano Lett. 2019;19:520.

[39] Madsen J, Liu P, Kling J, et al. A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images. Adv Theory Simul. 2018;1:1800037.

[40] Sumpter BG, Noid DW. Potential energy surfaces for macromolecules. A neural network technique. Chem Phys Lett. 1992;192:455.

[41] Blank TB, Brown SD, Calhoun AW, et al. Neural network models of potential energy surfaces. J Chem Phys. 1995;103:4129.

[42] Gassner H, Probst M, Lauenstein A, et al. Representation of intermolecular potential functions by neural networks. J Phys Chem A. 1998;102:4596.

[43] Behler J, Parrinello M. Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys Rev Lett. 2007;98:146401.

[44] Kohn W. Density functional and density matrix method scaling linearly with the number of atoms. Phys Rev Lett. 1996;76.

[45] Botu V, Ramprasad R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. Int J Quantum Chem. 2015;115(16):1074-1083.

[46] Botu V, Ramprasad R. Learning scheme to predict atomic forces and accelerate materials simulations. Phys Rev B. 2015;92:94305.

[47] Glielmo A, Sollich P, De Vita A. Accurate interatomic force fields via machine learning with covariant kernels. Phys Rev B. 2017;95:1.

[48] Li Z, Kermode JR, De Vita A. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. Phys Rev Lett. 2015;114:96405.

[49] Shapeev AV. Moment tensor potentials: a class of systematically improvable interatomic potentials. Multiscale Model Simul. 2016;14:1153.

[50] Xie T, Grossman JC. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Phys Rev Lett. 2018;120:145301.

[51] Isayev O, Oses C, Toher C, et al. Universal fragment descriptors for predicting properties of inorganic crystals. Nat Commun. 2017;8:15679 EP.

[52] Schütt KT, Sauceda HE, Kindermans P-J, et al. SchNet–A deep learning architecture for molecules and materials. J Chem Phys. 2018; 148:241722.

[53] Csányi G., On representing chemical environments. Phys Rev B. 2013;184115 (87):219902.

[54] Zeni C, Rossi K, Glielmo A, et al. Building machine learning force fields for nanoclusters. J Chem Phys. 2018;148:1802.01417.

[55] Artrith N, Urban A, Ceder G. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. Phys Rev B. 2017;96:014112.

[56] Imbalzano G, Anelli A, Giofré D, et al. Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials. J Chem Phys. 2018; 148(24):241730.

[57] Glielmo A, Zeni C, De Vita A. Efficient nonparametric-body force fields from machine learning. Phys Rev B. 2018;97:1.

[58] Artrith N, Behler J. High-dimensional neural network potentials for metal surfaces: A prototype study for copper. Phys Rev B. 2012;85:045439.

[59] Artrith N, Hiller B, Behler J. Neural network potentials for metals and oxides - first applications to copper clusters at zinc oxide. Phys Status Solidi(b). 2013;1203:1191.

[60] Artrith N, Kolpak AM. Understanding the Composition and Activity of Electrocatalytic Nanoalloys in Aqueous Solvents: A Combination of DFT and Accurate Neural Network Potentials. Nano Lett. 2014;14:2670.

[61] Artrith N, Kolpak AM. Grand canonical molecular dynamics simulations of Cu–Au nanoalloys in thermal equilibrium using reactive ANN potentials. Comput Mater Sci. 2015;110:20.

[62] Chiriki S, Bulusu SS. Modeling of DFT quality neural network potential for sodium clusters: Application to melting of sodium clusters (Na20 to Na40). Chem Phys Lett. 2016;652:130.

[63] Jindal S, Bulusu SS. Neural network potentials for dynamics and thermodynamics of gold nanoparticles. 2017;28:146(8):084314.

[64] Ulissi ZW, Tang MT, Xiao J, et al. Machine-Learning Methods Enable Exhaustive Searches for Active Bimetallic Facets and Reveal Active Site Motifs for CO_2 Reduction. ACS Catal. 2017;7:6600.

[65] Ouyang R, Xie Y, Jiang D-E. Global minimization of gold clusters by combining neural network potentials and the basin-hopping method. Nanoscale. 2015;7:14817.

[66] Chiriki S, Jindal S, Bulusu SS. c-T phase diagram and Landau free energies of (AgAu) 55 nanoalloy via neural-network molecular dynamic simulations. J Chem Phys. 2017;147:154303.

[67] Kolsbjerg EL, Peterson AA, Hammer B. Neural-network-enhanced evolutionary algorithm applied to supported metal nanoparticles. Phys Rev B. 2018;97:195424.

[68] Sun G, Sautet P. Metastable structures in cluster catalysis from first-principles: structural ensemble in reaction conditions and metastability triggered reactivity. J Am Chem Soc. 2018;140:2812.

[69] Hajinazar S, Sandoval ED, Cullo AJ, et al. Multitribe evolutionary search for stable Cu–Pd–Ag nanoparticles using neural network models. Phys Chem Chem Phys. 2019;21:8729.

[70] Bartók AP, Csányi G. Gaussian approximation potentials: A brief tutorial introduction. Int J Quantum Chem. 2015;115:1051.

[71] Chiriki SBulusu SS. Neural network potentials for dynamics and thermodynamics of gold nanoparticles. J Chem Phys. 2017; 146(8): 084314.

[72] Jindal S, Bulusu SS. A transferable artificial neural network model for atomic forces in nanoparticles. J Chem Phys. 2018;149:194101.

[73] Thompson A, Swiler L, Trott C, et al. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. J Comput Phys. 2015;285:316-330.

[74] Willatt MJ, Musil F, Ceriotti M. Atom-density representations for machine learning. J Chem Phys. 2019;150:154110.

[75] Deringer VL, Csányi G. Machine learning based interatomic potential for amorphous carbon. Phys Rev B. 2017;95:094203.

[76] Vandermause J, Torrisi SB, Batzner S, et al. On-the-Fly Bayesian Active Learning of Interpretable Force-Fields for Atomistic Rare Events. arXiv:190402042v1 [phys Comput Phys]. (2019).

[77] Neal RM. Bayesian learning for neural networks. Vol. 118. USA: Springer Science & Business Media; 2012.

[78] Rasmussen CKI, Edward C, Williams. Gaussian processes for machine learning. USA: MIT Press; 2006.

[79] Jinnouchi R, Asahi R. Predicting Catalytic Activity of Nanoparticles by a DFT-Aided Machine-Learning Algorithm. J Phys Chem Lett. 2017;8:4279.

[80] Jinnouchi R, Hirata H, Asahi R. Extrapolating energetics on clusters and single-crystal surfaces to nanoparticles by machine-learning scheme. J Phys Chem C. 2017;121.

[81] Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Networks. 1991;4:251.

[82] Csáji BC. [MSc Thesis]. Budapest, Hungary: Eötvös Loránd University (ELTE); 2001.

[83] Kang J, Noh SH, Hwang J, et al. First-principles database driven computational neural network approach to the discovery of active ternary nanocatalysts for oxygen reduction reaction. Phys Chem Chem Phys. 2018;20:24539.

[84] Goodfellow I, Bengio Y, Courville A. Deep learning. USA: MIT Press; 2016. Available from: http://www.deeplearningbook.org

[85] Ryczko K, Mills K, Luchak I, et al. Convolutional neural networks for atomistic systems. Comput Mater Sci. 2018;149:134.

[86] Glielmo A, Zeni C, Fekete Á, et al. Building non parametric n-body force fields using Gaussian process regression. arXiv:190507626 [phys Comput Phys]. (2019).

[87] Zeni C, Ádám F, Glielmo A. "MFF a python package for building nonparametric force fields from machine learning". 2018.

[88] Gastegger M, Marquetand P. Molecular Dynamics with Neural-Network Potentials. arXiv:1812.07676. (2018).

[89] Jennings PC, Lysgaard S, Hummelshøj JS, et al. Genetic algorithms for computational materials discovery accelerated by machine learning. NPJ Comput Mater. 2019;5:46.

[90] Bartók AP, De S, Poelking C, et al. Machine learning unifies the modeling of materials and molecules. Sci Adv. 2017;3:1.

[91] Musil F, De S, Yang J, et al. Machine learning for the structure–energy–property landscapes of molecular crystals. Chem Sci. 2018;9:1289.

[92] Mahoney MW, Drineas P. CUR matrix decompositions for improved data analysis. Proc Nat Acad Sci. 2009;106:697.

[93] Bernstein N, Csányi G, Deringer VL. De novo exploration and self-guided learning of potential-energy surfaces. arXiv:1905.10407. (2019).

[94] Zhai H, Alexandrova AN. Ensemble-average representation of pt clusters in conditions of catalysis accessed through GPU accelerated deep neural network fitting global optimization. J Chem Theory Comput. 2016;12:6213.

[95] Huo H, Rupp M. Unified Representation of Molecules and Crystals for Machine Learning. arXiv:1704.06439. (2017).

[96] Rupp M, Tkatchenko A, Müller K-R, et al. Fast and accurate modeling of molecular atomization energies with machine learning. Phys Rev Lett. 2012;108:058301.

[97] Jäger MOJ, Morooka EV, Canova FF, et al. Machine learning hydrogen adsorption on nanoclusters through structural descriptors. NPJ Comput Mater. (2018).

[98] Aguado A, Oleg K. First-principles determination of the structure of NaN and Na–N clusters with up to 80 atoms. J Chem Phys. 2011;164304.