Review

Advanced machine learning methods for learning from sparse data in high-dimensional spaces: a perspective on uses in the upstream of development of novel energy technologies

Sergei Manzhos [1]* and Manabu Ihara [1]

[1] School of Materials and Chemical Technology, Tokyo Institute of Technology, Ookayama 2-12-1, Meguro-ku, Tokyo 152-8552 Japan; E-mail: manzhos.s.aa@m.titech.ac.jp (S.M.), mihara@chemeng.titech.ac.jp (M.I.)

* Corresponding author, Tel.: +81-3-5734-3337

Abstract: Machine learning (ML) has found increasing use in research on energy conversion and storage technologies, in particular, so-called sustainable technologies. While often ML is used to directly optimize the parameters or phenomena of interest in the space of features, in this perspective, we focus on using ML to construct objects and methods that help in or enable the modeling of the underlying phenomena. We highlight the need for machine learning from very sparse and unevenly distributed numeric data in multidimensional spaces in these applications. After a brief introduction of some common regression-type machine learning techniques, we will focus on more advanced ML techniques which use these known methods as building blocks of more complex schemes and thereby allow working with extremely sparse data and also allow generating insight. Specifically, we will highlight the utility of using representations with sub-dimensional functions by combining the high-dimensional model representation ansatz with machine learning methods like neural networks or Gaussian process regressions in applications ranging from heterogeneous catalysis to nuclear energy.

Keywords: machine learning; neural network; Gaussian process regression; curse of dimensionality; high-dimensional model representation; energy conversion and storage; heterogeneous catalysis.

1. Introduction

Machine learning (ML) as well more complex techniques of artificial intelligence (AI) have been finding increasing use in research and development for novel technologies [1,1–7]. Whenever one can identify an input-output mapping whose construction is helpful in achieving a research or engineering goal, ML techniques are often of assistance. One can cast a research problem as a problem of constructing such a mapping and then searching in the space of the descriptors (features or inputs) [8,9] for a desirable optimal point. One area where ML is gaining more and more traction are novel energy conversion and storage technologies. These techniques are, in particular, intensely explored for application to the development of technologies typically associated with sustainable generation and use of energy such as advanced types (organic and inorganic materials based) of solar cells and LED (light-emitting diodes) [10–22], inorganic and organic metal ion batteries [23,24], fuel cells and generally heterogeneous catalysis including electro- and photocatalysis [25–34]. This is natural in the sense that the development of these technologies often passes through optimization and balancing of multiple factors acting simultaneously and to opposite ends; for example, in the case of organic solar cells, there is an optimum to be sought between the donor's bandgap, the band offset between the donor and the acceptor, the reorganization energies of both the donor and the acceptor, the charge transfer integral etc. [12,15,16]. These properties themselves are a function of the structure of the molecules which can be encoded through multiple descriptors [9]. In principle, most of these properties can be computed and or measured experimentally but doing so for each candidate

© 2022 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

molecule is expensive. It is enticing, based on a limited number of measurements and or calculations, to build a mapping – with the help of machine learning – between the chosen descriptors and the properties of interest, and then search the map for an optimal point (such as maximum power conversion efficiency in the case of a solar cell). Similarly, optima need to be sought in the design of battery materials (balancing thermodynamics and kinetics of cation insertion, stability etc.), catalysts (balancing adsorption energies of key species, kinetics and stability) and other types of functional materials. Some examples will be given in Section 2.

Functional materials and interfaces are key to the design of novel energy technologies. As compositional space is too vast to be explored by brute-force and trial-and-error approaches, rational guidance of design is necessary. Such rational guidance comes, in particular, from simulations at various scales (from atomistic and electronic structure levels to macroscopic). Specifically, for technologies such as fuel cells, batteries, or solar cells, electronic properties are key to functionality, and it is atomistic-scale, and in particular ab initio, simulations that are providing insight into the mechanism at the material level and thereby can guide rational design. Bandgaps and bandstructures, absorption spectra, molecular adsorption energies etc are today routinely computable at the DFT (density functional theory) [35,36] level. Further, structural information is obtainable with techniques such as molecular dynamics [37], Monte Carlo [38] and others which can operate at larger scales than DFT. Such calculations, however, remain costly, in particular, DFT with which routine calculations are still typically limited to about 10^3 atoms. It is therefore enticing to deploy ML methods to replace such calculations or at least to reduce the required number of such calculations. Of course, ML can boost the development not only of the so-called sustainable or "green" energy technologies; we will provide an example in this perspective where it can be used to advance research useful for the future of nuclear energy.

The purpose of this perspective is not a comprehensive or even brief review of ML uses in materials science and energy technologies nor is it to review staple ML methods – there is already ample literature on these topics [1–34]. Novel energy technologies are very knowledge-deep i.e. they require multiple steps in understanding and development; specifically, they require insight at the electronic structure level. Many aspects are being addressed with machine learning not just predicting material or device properties from descriptors - much is being done at the backend or upstream by using machine learning in the workflow of understanding modeling of materials and phenomena for these technologies, or to improve modeling capabilities [1,39–44]. *This in particular requires ML methods and tools capable of working with extremely sparse and unevenly distributed data in multidimensional spaces beyond the capabilities of standard methods. That is the focus of this perspective.*

We will first show illustrative examples of how machine learning can be used to help improve the way we produce and use energy, especially used for the benefit of so-called sustainable technologies. We will highlight the fact that in these applications, one usually has to deal with multiple variables i.e. to operate in multidimensional spaces, and we will see that there are properties of multidimensional spaces with which one typically does not have to deal with in one-, two-, or three-dimensional problems. We will show why ML techniques are effective when working with such data. We will briefly introduce neural networks (NN) and Gaussian process regressions (GPR) and their pros and cons. We will then focus on using the so-called high-dimensional model representations to structure the functional representation, in conjunction with NN or GPR used to build components of those representations. We will see that this is especially useful when trying to recover trends or functional forms from sparse data and to gain insight on the examples from catalysis and a perspective nuclear technology as well as provide an example how ML can revolutionize ab initio modeling capabilities which are critical for the modeling and mechanistic understanding of novel energy technologies.

2. High dimensionality of and extremely low data density in the space of descriptors

2.1. Examples of input-output mappings used in ML for energy technologies

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

The humanity is trying to develop more sustainable ways to produce and use energy. Currently the world still largely relies on fossil fuels which are believed to be the result of absorption of sun energy [45]. The use of fossil fuels generates many pollutants, including poisonous gases and heavy metals, and it also generates large amounts of CO_2. CO_2 can be recycled through vegetation but unfortunately the timeframe from dead vegetation to fossil fuels is too long to rely on such a natural carbon cycle with current and projected levels of energy consumption. There is a quest for what can be called an anthropogenic chemical carbon cycle [46]. In it, one does not rely on burning of fossil fuels but on a set of technologies which either generate electricity directly from the sun with solar cells [47] and from other sustainable sources such as wind [48], or synthesize, using sunlight and CO_2 and other inputs, fuels which can be utilized in a clean way, such as hydrogen or liquids which can be used with cleaner exhaust in fuel cells or even burnt directly [49–51].

Solar and wind technologies are of course intermittent, and to match their output schedule to the demand schedule, storage is necessary [52], in particular with batteries ([53] and references therein). Batteries are also needed for road transport electrification. At the same time, novel and safer types of safer nuclear reactors are being developed for reliable low-carbon footprint baseload [54–56]. All these technologies become an energy mix. Multiple technologies are used, and in each of these technologies there are multiple scientific, engineering, and technological problems that need to be addressed. Also bringing these technologies together to work in one system is a challenge. Machine learning can help resolve many of these problems, and we will now show some examples.

The reader will have heard about "hydrogen economy" and about fuel cells which allow "burning" in a clean way either hydrogen or other fuels. At the core of these capabilities is heterogeneous catalysis, i.e. reactions happening at and catalyzed by surfaces. Some of the key reactions relevant for energy production and use are driven by solid catalysts including the oxygen reduction reaction [57] which is a critical and still problematic step in fuel cells, the Fischer-Tropsch process [58] which allows producing synthetic fuels from a mixture of hydrogen and CO (so-called syngas); water-gas shift [59] and steam and dry reforming [60,61] are also very important reactions. These reactions permit transformation between hydrocarbons, water, hydrogen, and carbon oxides. Their efficiency depends on the quality of the catalyst. Many efficient catalysts are rare and or expensive such as platinum widely used in commercial fuel cells, e.g. those used in Toyota Mirai, in particular, due to the lack of non-expensive alternatives.

Design of better catalysts is therefore a major bottleneck on the way to wide deployment of more sustainable energy technologies. Recently, machine learning has found increasing use in rational design of catalysts [25–34]. Typically, one considers a set of descriptors of catalytic activity which include reaction thermodynamics and kinetics, adsorption energies of key reactants and intermediates, as well as structure and electronic structure descriptors of the catalyst [26,33]; some are shown in **Figure 1**. To sieve through candidate materials, one encodes them into a set of compositional and structural variables such as atomic numbers and positions, crystal structure and surface cut, and adsorption site. Structures and many of the properties such as adsorption and reaction energies and kinetic barriers as well as electronic structure descriptors are routinely computed, typically with density functional theory [36,62]. Computations play a key role in rational design of catalysts with or without machine learning, as many properties are either not directly accessible experimentally or cannot be measured with high throughput and modest cost. All together this makes a lot of variables on which catalytic activity depends and one needs to perform searches and uncover relations in multidimensional spaces, which can be done with the help of machine learning [25–29].

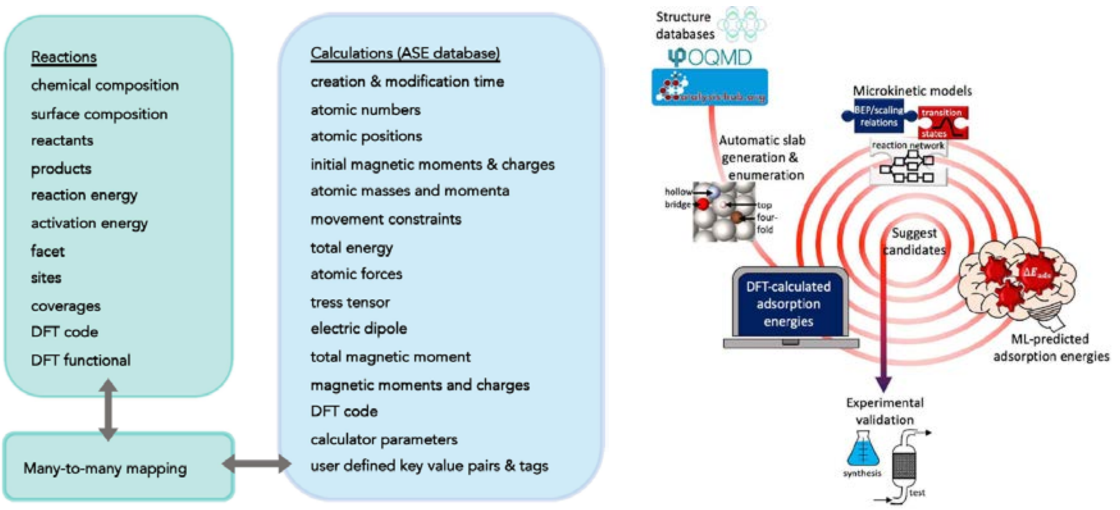

Figure 1. Left: Schematic of the SQL data structure of Catalysis-Hub.org, used to store reaction energies (Reactions table, green) and DFT calculations (ASE database, blue). Since each reaction energy involves several DFT calculations (and the same DFT calculations can potentially be used for several reactions), a many-to-many mapping schema is used to preserve connections between the table rows. Right: Machine learning-enhanced catalyst candidate prediction: Bulk and surface structures retrieved from structure databases like materialsproject.org, OQMD, catalysis-hub.org, etc. are used for automated slab generation and enumeration of possible adsorption sites. In an iterative process, limited numbers of DFT-calculated adsorption energies and machine-learning-predicted adsorption energies are used to inform microkinetic models to eventually suggest promising catalyst candidates that should be investigated by experiment. Adapted with permission from Ref. [26]. Copyright 2019 Wiley-VCH Verlag GmbH.

To model directly a catalyzed reaction from reactants to products, one would need an interatomic potential energy function, and those potentials also pose a bottleneck in the modeling of heterogeneous catalysis and are more and more often constructed with machine learning [1,40,63,64]. There is a key difference between the traditional fossil fuel based energy technologies and the newer technologies based on solar cells, fuel cells, batteries etc.: *the fossil fuel based technologies can be understood based on classical thermodynamics and mechanics, while these newer technologies critically depend on quantum phenomena, ultimately on electronic states energies, occupancies, and localizations. Quantum mechanics based modeling therefore takes center stage.* Such modeling is difficult and costly, and machine learning is also used to facilitate it; we will show a couple examples thereof below. This is most obvious in solar cells where details of the electronic structure such as the bandgap, the effective mass etc. determine directly solar cell performance. Machine learning is now more and more often used to help design better solar cells and specifically materials for solar cells and other optoelectronic applications [12–15,65,66]. It is used to predict better active materials, to optimize device performance or even optimize fabrication processes [15]. **Figure 2** shows main uses of ML for solar cell design as well as most widely used ML methods summarized in a recent review [15]. Artificial neural networks (ANN) remain the most widely used method, also widely used are genetic algorithms (GA), random forest (RF), particle swarm optimization (PSO), simulated annealing (SA), support vector machines (SVM), kernel ridge regression (KRR) and Gaussian process regression (GPR), externally randomized trees (ERT), clustering methods such as K nearest neighbors, principal component analysis - these are just some methods from by now very many machine learning methods. That is, all three major classes of ML – classification, regression, and clustering – are finding use in solar cell research as well as in catalysis research [26].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

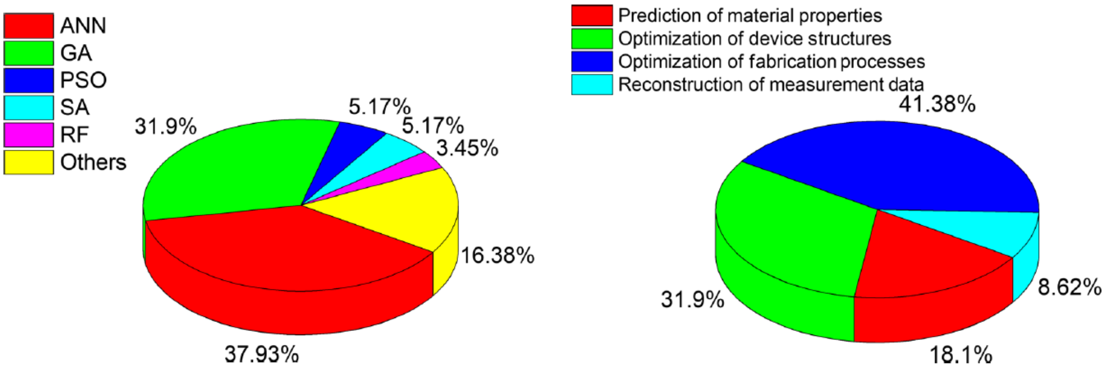

Figure 2. Left: Distribution of types of ML techniques applied in design and fabrication of solar cells. Right: Distribution of applications in design and fabrication of solar cells assisted by ML techniques. Adapted with permission from Ref. [15]. Copyright 2019 Wiley-VCH Verlag GmbH.

These examples from the fields catalysis and solar cells are barely illustrative. Machine learning (and the same methods as mentioned above) is also used to help design battery materials [23,24,67,68] and for other energy technologies as indicated above, but ML is in demand not just at the material or device level but also at the system level [69–72]. In an energy mix containing solar farms, wind farms, and other intermittent technologies alongside more established generation methods such as nuclear or natural gas powered stations, one needs to constantly balance supply and demand. Predicting demand and predicting possible supply from solar and wind and choosing how to palliate any excess or shortfall with either battery storage or call on nuclear or hydrocarbon based generation is an important problem where there is much potential for machine learning which is yet to be fully realized [73,74]. It also requires working with multidimensional datasets.

To illustrate specifically the challenge posed by the dimensionality of the feature space, in ML for materials for novel energy technologies, consider one recent example from the literature [10] of how ML is applied to design better materials for perovskite solar cells [75], in which the main active material and light absorber is a (typically organic-inorganic hybrid) perovskite material. The composition of the perovskite can be changed by different choices of atoms placed in different sites of the crystal lattice. The perovskites most widely used in labs today contain lead which is not desirable. Moreover, different structural choices also in principle allow one to modulate the bandgap and other properties which could better match with specific electron and hole transporting materials [76] which are in contact with the perovskite in a solar cell. In the example of Ref. [10], candidate compositions were encoded with 32 features which included properties of constituent atoms such as ionization potential, orbital radii etc. as well as properties of the crystal. We will see below that this dimensionality, which from the point of view of applications does not appear to be high (indeed, hundreds to thousands of features are sometimes available), is in fact very high from the point of view of data density.

We pointed out that in all of the above mentioned applications of machine learning, a multitude of methods are used: classification methods, regression methods, clustering methods; supervised and unsupervised method. In what follows, we will narrowly focus on specific regression type methods based on neural networks and Gaussian process regressions and on composite methods using them as building blocks to enable ML in high dimensional spaces from extremely sparse data.

2.2. New technologies and challengies require new simulation methods – a large scope for machine learning

Machine learning is also useful to improve simulation methods. The example of machine learning interatomic potentials was cited above; ML is also used to improve the widely used DFT method by learning better exchange correlation functionals, dispersion corrections, corrections to computed excitation energies etc. ([39] and references therein). We will draw attention here to only one but critically important example which allows to showcase the power of ML. We start with a seemingly purely mathematical problem: consider a function of space $\rho(\boldsymbol{x})$, where $\boldsymbol{x}$ is a vector of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

Cartesian coordinates x, y, z. The function is positively definite and integrates to N: $\int \rho(\boldsymbol{x})d\boldsymbol{x} = N$. We represent ρ as a sum of N positively definite pieces each integrating to 1:

$$\begin{gathered}\rho(\boldsymbol{x}) = \sum_{n=1}^{N} \rho_i(\boldsymbol{x}) = \sum_{n=1}^{N} |\phi_i(\boldsymbol{x})|^2 \\ \int \rho_i(\boldsymbol{x})d\boldsymbol{x} = 1\end{gathered} \tag{2.2.1}$$

Because they are positive definite, they are squares of some functions ϕ_i. It is known that this decomposition can be done so that ϕ_i are orthonormal: $\int \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})dx = \delta_{ij}$.

We are interested in this quantity T or the corresponding integrand $\tau(x)$ which can take two alternative forms:

$$\begin{gathered}T = -\frac{1}{2}\int \sum_{n=1}^{N} \phi_i \Delta \phi_i \, d\boldsymbol{x} = \int \tau_{(KS)}(\boldsymbol{x})d\boldsymbol{x} \\ T = \frac{1}{2}\int \sum_{n=1}^{N} |\nabla \phi_i|^2 \, d\boldsymbol{x} = \int \tau_{+}(\boldsymbol{x})d\boldsymbol{x}\end{gathered} \tag{2.2.2}$$

We want to express T without explicit reference to ϕ_i, as a function of ρ-dependent quantities only. It could be any derivative of ρ, any power of ρ, but only of ρ without explicit reference to ϕ_i.

The reason why this example is important is that a solution to this problem opens the way to routine, linear-scaling large-scale ab initio modeling of materials with the so-called orbital-free DFT (OF-DFT) [77]. We mentioned above that quantum mechanics based modeling is critical for understanding and design of novel energy technologies, yet the commonly used Kohn-Sham DFT in unwieldy for system with more than 10^1-10^2 atoms. Large scale modeling means more realistic modeling and is required to properly account for a range of phenomena which are intrinsically large-scale (e.g. microstructure-driven properties). In OF-DFT, ρ takes the meaning of the electron density, N is the number of electrons, ϕ_i are single-electron (Kohn-Sham) orbitals [36], T is the kinetic energy, and $\tau(\boldsymbol{x})$ is the kinetic energy density (KED) (we neglect spins and partial orbital occupancy without the loss of generality). The mapping $T = T[\rho(\boldsymbol{x})]$ is the kinetic energy functional (KEF). Approximate formulas for such an expression ($T[f(\boldsymbol{x})]$) exist but they are not accurate enough for use in most applications where ab initio modeling is needed, including (organic and inorganic) semiconductors and transition metal containing functional materials of novel energy technologies. In the last several years, substantial progress is being made on this problem with the help of machine learning using, in particular, techniques like neural networks and kernel methods [41,78–82]. We will return to this example later in the context of deep learning. Some uses of ML to improve quantum mechanics based modeling methods are reviewed in Ref. [39].

2.3. The curse of dimensionality and why ML techniques are effective

We mentioned in an example above 32 features used to describe perovskite materials [10]. Are 32 dimensions high or low? – it is (very) high. In multidimensional spaces one is hit by the so-called curse of dimensionality. Imagine we sample a simple univariate function with M points sufficient to recover the function to a desired accuracy with a common method (e.g. splines). If one wants to maintain the same density of sampling in D dimensions, the number of sampling points would grow exponentially: M^D. With M as small as 10 and D as small as 10, one would need 10^{10} data. This is the curse of dimensionality. Moreover, when one constructs a function with polynomials or Fourier expansions not only the number of required samples grows exponentially but the number of terms in the representation as well. The result of this is that it is impossible to achieve good density of sampling just by adding more data. For example, already in 20 dimensions, a million data points is equivalent to only about 2 data per degree of freedom (of an equivalent direct product grid). If we somehow managed to get 10 times more data, it would only increase the density of sampling to about 2.2 data per degree of freedom. Practically, one therefore always works with extremely sparse data.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

That is why 32 dimensions in the example above is in fact very high. In practice, one starts feeling this curse of dimensionality from about 6 dimensions and up.

However, this debilitating scaling strictly speaking only holds for direct product representations. One reason why various machine learning methods work well in multidimensional spaces is because they avoid direct product representations and thereby alleviate the problem of the curse of dimensionality. We will illustrate this point in section 3.1.

In multidimensional spaces, some intuitions break down. For example, what is local in low-*D* may not be in high-*D*. An example is the Gaussian function, $g(\boldsymbol{x}) = \prod_{i=1}^{D}(2\pi\sigma_i^2)^{-\frac{1}{2}} exp\left(\frac{(x_i-\overline{x_i})^2}{2\sigma_i^2}\right)$, which is localized around the mean $\overline{\boldsymbol{x}}$ in the sense that about 70% of the quadrature of this function is within one standard deviation σ from the mean in the one-dimensional case (*D* = 1). In six dimensions (*D* = 6), for example, only 10% of the quadrature is from within one standard deviation (in each respective dimension). That is, a multidimensional Gaussian function is no longer a localized function by this measure. Working in high dimensionality has also its advantages. For example, there is the concentration of measure whereby as dimension is increased, the width of the distribution of distances between data points collapses: $\lim_{D\to\infty} E\left(\frac{dist_{max}(D)-dist_{min}(D)}{dist_{min}(D)}\right) \to 0$, where $dist_{max}$ and $dist_{min}$ are maximum and minimum distances between the data points. We observed in our own experience that, for example, neural networks perform better in high-dimensional spaces than in 1, 2, 3 dimensions. This ultimately relates to growing advantages of non-direct product representations in high dimensions.

One important consequence of the low density of sampling is that the intrinsic dimensionality of the data set is less than the dimensionality of the space [83,84]. This is obvious when the data are confined to lower-dimensional hypersurfaces or other shapes, but even if the data are not confined to such a sub-dimensional shape, simply by the virtue of low density, the intrinsic dimensionality of the dataset is lowered. This ultimately justifies representations with lower-dimensional functions which will be described below.

3. Advanced techniques for working with sparse data

3.1. Brief introduction to neural networks and Gaussian process regression

3.1.1. Neural networks (NN)

Artificial neural networks [85] are an example of a representation of a multivariate function with univariate functions which are adapted to the problem by the choice of parameters. They are often presented with the help of an analogy with a biologic neural network, but for regression problems we find more useful the following interpretation. We expand a function $f(\boldsymbol{x}), \boldsymbol{x} \in R^D$ in a basis of univariate functions σ_n with coefficients c_n:

$$f_k(\boldsymbol{x}) = \sum_{n=0}^{N} c_{nk}\sigma_n(\boldsymbol{w_n x} + b_n) \qquad (3.1.1.1)$$

Here we added a subscript *k* to indicate that several outputs can be computed from the same basis. The basis functions depend on all components of $\boldsymbol{x}$ but their arguments are scalars dependent on $\boldsymbol{x}$ as well as on weights $\boldsymbol{w}$ and biases *b*. This parameterization by $\boldsymbol{w}$ and *b* makes a flexible, *non-direct product* basis. This representation goes back to the Kolmogorov theorem of 1957 [86], and since then, in a series of papers restrictions on σ have been relaxed [87–95]. This expression is a universal approximator even when σ is the same for all *n* and as long as σ is smooth and nonlinear [96]. In applications, typically σ is the same for all *n* and typically it is the sigmoid function, $\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x})$, but it does not have to be. The parameters w and *b* and the coefficients *c* are fitted to reproduce a set of known samples of *f*, $f^j = f(\boldsymbol{x}^j)$, *j* = 1, …, *M*. The fit is nonlinear because σ is nonlinear.

The equation (3.1.1.1) (which is often written as $f(\boldsymbol{x}) = \sigma(\sum_{n=0}^{N} c_n \sigma(\boldsymbol{w_n x} + b_n))$ i.e. with a so-called output neuron, which, however, can be subsumed in the definition of f, without loss of generality) expresses a so-called single hidden layer neural network, it is graphically represented in **Figure 3**, where the arrows to each σ, which are called neurons, reflect the formation a single scalar input. These neurons form a hidden later. The outputs are collected in a sum to one or more outputs in the output layer, i.e. one can fit a multi-sheet function or a function and its derivatives, by the same NN. These are called output neurons.

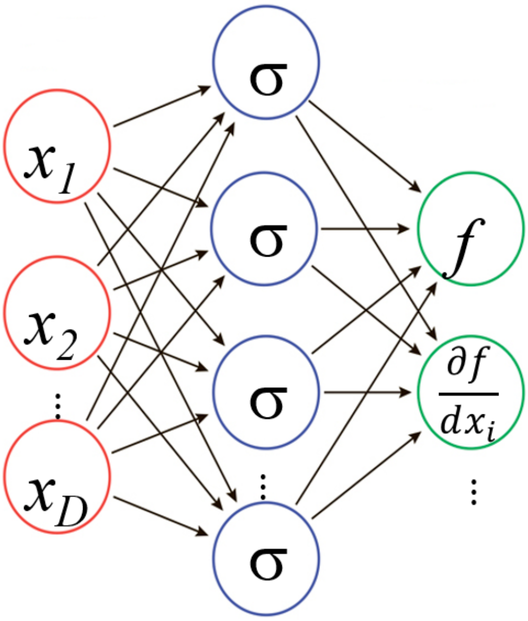

Figure 3. Schematic representation of a single-hidden layer neural network.

One can collect the outputs of this last layer of neurons and feed them to yet another layer of neurons and so on, obtaining a multilayer or deep neural network:

$$f_l = \sigma_{out}\left(\sum_{k_n=0}^{N_n} w_{l,k_n}^{(n)} \sigma_{n,\,k_n}\left(\sum_{k_{n-1}=0}^{N_{n-1}} w_{k_n,k_{n-1}}^{(n-1)} \sigma_{n-1,k_{n-1}}\left(\cdots \sum_{k_1=0}^{N_1} w_{k_2 k_1}^{(2)} \sigma_{1,k_1}\left(\sum_{i=0}^{d} w_{k_1 i}^{(1)} x_i\right)\right)\right)\right) \qquad (3.1.1.2)$$

We stress that a single hidden layer network is already a universal approximator. In many applications, one layer would be sufficient and one needs to be sure if one actually needs a deep NN as it comes at a price of a large number of non-linear parameters. We will show examples later when one actually needs a deep NN.

The fact that σ can be any smooth function opens new possibilities. We showed, for example, that with exponential neurons one easily obtains a sum of products representation with a relatively small number of terms [97]: $f(\boldsymbol{x}) = \sum_{i=0}^{M} w_i^{(2)} \prod_{k=0}^{d} e^{w_{ik}^{(1)} x_k}$. Sum of products representations are very useful because they greatly simplify integration of the function (multidimensional integrals over the function f can be computed as sums of products of one-dimensional integrals), which is a major advantage when the dimensionality is high. In fact, sum of product representations are required in certain computational methods, for example is some powerful quantum dynamics methods [98]. A more complicated way to get a sum of products are multiplicative neural networks which use error function type of neurons [99]: $f(\boldsymbol{x}) = \mu_1^{(2)} + \sum_{k_1=1}^{n_1} w_{1,k_1}^{(2)} \prod_{k_0=1}^{J} erf\left(\mu_{k_1 k_0}^{(1)} + w_{k_1 1 k_o}^{(1)} x_{k_0}\right)$. These are examples of a departure from the orthodox network designs which bring significant advantages. We will show below how with even more drastic changes in the architecture one can realize significant advantages.

3.1.2. Gaussian process regression (GPR)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

Another method we briefly highlight is Gaussian process regression (GPR) [100]. GPR answers the question "given the set of samples f^j of a function $f(\boldsymbol{x})$ at certain points in space $\boldsymbol{x}^j$, what are the expectation values $f(\boldsymbol{x})$ and their variance $\Delta f(\boldsymbol{x})$ for function values at other points in space $\boldsymbol{x}$?" One assumes that correlation between data can be described with a kernel, a chosen type of the covariance function $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$. The answer is given by

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{K}^{*}\boldsymbol{K}^{-1}\boldsymbol{f} \\ \Delta f(\boldsymbol{x}) &= K^{**} - \boldsymbol{K}^{*}\boldsymbol{K}^{-1}\boldsymbol{K}^{*\boldsymbol{T}} \end{aligned} \qquad (3.1.2.1)$$

where $\boldsymbol{f}$ is a vector of all f^j values, and the matrices $\boldsymbol{K}$ and $\boldsymbol{K^*}$ are computed from pairwise covariances among the data:

$$\boldsymbol{K} = \begin{pmatrix} k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) + \delta & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) & \cdots & k(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(M)}) \\ k(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}) & k(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(2)}) + \delta & & k(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(M)}) \\ \vdots & & \ddots & \vdots \\ k(\boldsymbol{x}^{(M)}, \boldsymbol{x}^{(1)}) & k(\boldsymbol{x}^{(M)}, \boldsymbol{x}^{(2)}) & \cdots & k(\boldsymbol{x}^{(M)}, \boldsymbol{x}^{(M)}) + \delta \end{pmatrix} \qquad (3.1.2.2)$$

$$\boldsymbol{K}^{*} = \left(k(\boldsymbol{x}, \boldsymbol{x}^{(\boldsymbol{1})}) \quad k(\boldsymbol{x}, \boldsymbol{x}^{(\boldsymbol{2})}) \quad \ldots \quad k(\boldsymbol{x}, \boldsymbol{x}^{(\boldsymbol{M})})\right), \quad K^{**} = k(\boldsymbol{x}, \boldsymbol{x})$$

The optional δ on the diagonal has the meaning of the magnitude of Gaussian noise and it helps generalization. Note that Eq. (3.1.2.1) as written (and as it commonly appears in the literature) holds for data (the set of f^j) normalized to unit variance, otherwise the left hand side should be multiplied by the variance of f.

The covariance function is usually chosen as one of the Matern family of functions [101] given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{|\boldsymbol{x} - \boldsymbol{x}'|}{l}\right)^{\nu} K_{\nu}\left(\sqrt{2\nu}\frac{|\boldsymbol{x} - \boldsymbol{x}'|}{l}\right) \qquad (3.1.2.3)$$

At different values of ν this function becomes a Gaussian ($\nu \to \infty$), a simple exponential ($\nu = 1/2$) and various other widely used functions (such as Matern3/2 and Matern5/2 for ν = 3/2 and 5/2, respectively). The parameters of the covariance function are the only parameters, they are hyper-parameters and they are few, as few as one (for an isotropic kernel at fixed ν). This is therefore a non-parametric method, which is an advantage. While hyper-parameters still need to be chosen, the performance is usually about equally good as long as hyper-parameters are in some reasonable range. In Eq. (3.1.2.3), the critical parameter is the length parameter l; the prefactor σ^2 is fully correlated with δ in Eq. (3.1.2.2). Note that Eq. (3.1.2.1) is a non-direct product representation.

3.1.3. Relative pros and cons of GPR vs NN

Because in GPR one has to wield matrices and inverses of the matrices of the size $M \times M$, where M is number of data, this method becomes increasingly costly as the dataset size grows. In fact, it quickly becomes prohibitively expensive for datasets as small as tens of thousands of data unless additional approximations are made [102]. But the method is very accurate. In fact, GPR is the only method which we have seen beat NN in controlled comparisons and on the quality of observables, i.e. not just various fitting error measures, but controlled tests of the quality of observables computed with functions constructed using NN or GPR [103]. GPR has been shown to be equivalent to an infinitely large neural network [104]. Contrary to NN, it provides estimates of uncertainty (we caution however that Δf in Eq. (3.1.2.1) should not be directly used to compute an error bar of the mode; it is the uncertainty of the expectation value, which can be higher or lower than the fitting error [105,106]). It is quite costly not just to build but also to evaluate a GPR model when the datasets are more than a few thousand data. The matrix inverse also means instability when data points are close to each other, although it can be handled e.g. by using pseudoinverses and appropriate values of δ. We also recently proposed a rectangular GPR which achieves numeric stability and good generalization without δ [107].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

Neural networks, on the contrary, work well when some points are close, they work well with millions of data even on a desktop computer [78]. Their disadvantage is a large number of parameters that need to be determined in a non-linear fit. There is therefore danger of overfitting and one needs to carefully control for it by a judicious choice of architecture and also by using test sets. Often as few as 5 to 20% of the available data are reserved for testing with the rest used for training. In our experience, this is grossly insufficient, and we recommend using test sets which are at least as large as training sets. This is a disadvantage as this requires more data. In general, with the same number of training points, an NN achieves a lower accuracy than GPR; alternatively, one needs more data to reach the same accuracy with NN as with GPR [103].

Given the advantages of GPR, it would be desirable to have a scheme in which GPR is used in a regime where it is most efficient i.e. with few data. The approach which we introduce next allows doing just that.

3.2. High-dimensional model representation (HDMR)

We now briefly introduce high-dimensional model representation (HDMR) [108–110]. HDMR is an expansion over orders of coupling. Its form is similar to the ANOVA decomposition [111,112]. The idea is to represent the function $f(\boldsymbol{x})$ as a sum of first an uncoupled approximation, then terms due to collective action of couples of variables, then triples etc, until the last term that describes the coupling term among all D variables:

$$f(\boldsymbol{x}) = f_0 + \sum_{i=1}^{D} f_i(x_i) + \\ + \sum_{\substack{i=1,\\ j=i+1}}^{D} f_{ij}(x_i, x_j) + \dots + \sum_{i_1,i_2,\dots,i_d}^{D} f_{i_1,i_2,\dots,i_d}(x_{i_1}, x_{i_2}, \dots, x_{i_d}) + \dots + f_{1,2,\dots,D}(\boldsymbol{x}) \tag{3.2.1}$$

If taken to this last term, the expansion is exact. If the expression is truncated at some lower-order d terms, it is approximate. In most physical phenomena, the relative importance of these coupling terms drops rapidly with the order of coupling d. Depending on the application, one can stop at e.g. 3rd order or 2nd order or even 1st order, i.e. uncoupled approximation, without making much error [108].

When one has stopped at some relatively small order of coupling d, one works only with low-dimensional component functions $f_{i_1,i_2,\dots,i_d}(x_{i_1}, x_{i_2}, \dots, x_{i_d})$. All these component functions can be obtained from one and the same set of training points distributed in full D-dimensional space (in which case one obtains the so-called RS (random-sampling)-HDMR [113,114]). This is advantageous on several counts: low-dimensional functions are generally easier to build, they can be built from fewer data e.g. with sparse data, and not suffer from overfitting [115]. We indicated above that sparse data are a key problem and that e.g. the Gaussian process regression does not work well with large dataset. One can use HDMR to build a multivariate function from sparse data and stay in the "comfort zone" of the method which is used to build the component functions. A representation with lower-dimensional functions is also easier to use in applications particularly when f needs to be integrated. HDMR in that case allows computing only low-dimensional quadratures.

In the original HDMR formulation, all component functions are mutually orthogonal, i.e.

$$\int_D f_{i_1 i_2 \dots i_d}(x_{i_1}, x_{i_2}, \dots, x_{i_d}) f_{j_1 j_2 \dots j_m}(x_{j_1}, x_{j_2}, \dots, x_{j_m}) d\boldsymbol{x} = 0 \\ \{i_1, i_2, \dots, i_d\} \neq \{j_1, j_2, \dots, j_m\} \tag{3.2.2}$$

and are equal to multidimensional integrals (specifically, $(D - d)$ dimensional integrals need to be computed for d-dimensional component functions) which are quite difficult to compute [113,114]; further, any lower-dimensional component functions need to be constructed first before constructing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

d-dimensional functions. We proposed an extension of HMDR whereby we do not build the entire expansion but directly represent the function *f* with *d*-dimensional functions [105,116–119].

$$f(\boldsymbol{x}) \approx \sum_{i=i}^{N} f_i(x_1\ , x_2\ , \dots , x_d\ ) \qquad (3.2.3)$$

This is achieved by dropping the orthogonality requirement. One need not build the entire HDMR expansion; it is possible to use only the terms of the desired dimensionality *d* that provides a desired accuracy. The lower-order terms are effectively subsumed into *d*-dimensional terms. Most importantly, when using machine learning to represent the component functions, no integrals need to be computed, and it is possible to alleviate the issue of combinatorial growth of the number of HDMR terms with *d* and *D*.

3.3. Combining HDMR with ML for learning from sparse data

3.3.1. Machine learning of HDMR terms

HDMR component functions can be built with machine learning methods like neural networks [116,118,120] or Gaussian process regressions [105,106,119]. Practically this can be done by fitting component functions one at a time to the difference of the value of the function f at the training points and the sum of all other component functions:

$$f_{k_1k_2\dots k_d}^{(NN\ or\ GPR)}\left(x_{k_1}, x_{k_2}, \dots , x_{k_d}\right) = f(\boldsymbol{x}) - \sum_{\substack{\{i_1i_2\dots i_d\}\in\{12\dots D\} \\ \{i_1i_2\dots i_d\}\neq\{k_1k_2\dots k_3\}}}^{D} f_{i_1i_2\dots i_d}^{(NN\ or\ GPR)}\left(x_{i_1}, x_{i_2}, \dots , x_{i_d}\right) \qquad (3.3.1.1)$$

One then cycles through all functions in a sort of a self-consistency loop until convergence. We called the resulting methods (RS-)HDMR-NN and (RS-)HDMR-GPR, respectively [105,120]. In Eq. (3.3.1.1) a single NN or GPR instance is used for each component function. In the case of NN, the entire HDMR representation can also be encored into the architecture of one NN. In practice, we found this to be rather cumbersome; the advantage of Eq. (3.3.1.1) is that existing NN engines with simple architecture can be used and work well. When using GPR, one can achieve an HDMR form of also by using an HDMR-type kernel [119,121]:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\{i_1i_2\dots i_d\}\in\{12\dots D\}} k_{i_1i_2\dots i_d}\left(\boldsymbol{x}_{i_1i_2\dots i_d}, \boldsymbol{x}'_{i_1i_2\dots i_d}\right) \qquad (3.3.1.2)$$

where $\boldsymbol{x}_{i_1i_2\dots i_d} \equiv \left(x_{i_1}, x_{i_2}, \dots , x_{i_d}\right)$. There is in that case the disadvantage over Eq. (3.3.1.1) in that a custom kernel has to be defined, but one uses a single GPR instance and the $f_{i_1i_2\dots i_d}^{GPR}(x_{i_1}, x_{i_2}, \dots , x_{i_d})$ are optimal in the least squares sense without the need for fitting cycles of Eq. (3.3.1.1). The individual component functions are then

$$f_{k_1k_2\dots k_d}\left(x_{k_1}, x_{k_2}, \dots , x_{k_d}\right) = \boldsymbol{K}^{*}_{i_1i_2\dots i_d}\boldsymbol{c} \qquad (3.3.1.3)$$

where $\boldsymbol{c} = \boldsymbol{K}^{-1}\boldsymbol{f}$ and $\boldsymbol{K}^{*}_{i_1i_2\dots i_d}$ is a row vector with elements $k_{i_1i_2\dots i_d}\left(\boldsymbol{x}_{i_1i_2\dots i_d}, \boldsymbol{x}^{(n)}_{i_1i_2\dots i_d}\right)$. The values of the component functions at the training set are then $\boldsymbol{f}_{i_1i_2\dots i_d} = \boldsymbol{K}_{i_1i_2\dots i_d}\boldsymbol{c}$. The relative importance of different component functions can be evaluated by computing the variance of $\boldsymbol{K}_{i_1i_2\dots i_d}\boldsymbol{c}$, where the (*m*,*n*) elements of the matrix $\boldsymbol{K}_{i_1i_2\dots i_d}$ are $k_{i_1i_2\dots i_d}\left(\boldsymbol{x}^{(m)}_{i_1i_2\dots i_d}, \boldsymbol{x}^{(n)}_{i_1i_2\dots i_d}\right)$. It can be used, in particular, to prune unnecessary component functions and thereby alleviate the issue of combinatorial scaling of the number of HDMR terms with *D* and *d* [106,119]. This scaling is a major bottleneck in high-dimensional problems. It arises because in HDMR the number and dimensionality of terms are hardwired to each other: the number of *d*-dimensional terms is the binomial coefficient $C_d(D)$ and can be quite high. An even further extension that we proposed is to represent *f*(*x*) as a sum of lower-dimensional functions in new coordinates $\boldsymbol{x}^{(i)}$ which are linear combination of the original coordinates:

$$\boldsymbol{x}^{(i)} = \boldsymbol{A}^{(i)}\boldsymbol{x} + \boldsymbol{b}^{(i)}$$
$$f(\boldsymbol{x}) \approx \sum_{i=i}^{N} f_i(x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}) \qquad (3.3.1.4)$$

The full set of is generally redundant, and the method, when using with NN component functions, was called Red-RS-HDMR-NN [117]. This has the advantage that the number of terms and their dimensionality are uncoupled, and one can achieve good accuracy with a small number N of low-dimensional terms by varying independently N and d. This is illustrated in **Figure 4** where the error in the interatomic potential of vinyl bromide is plotted as function of both the number and the dimensionality of terms. For comparison, in the standard HDMR, there would be about 3000 5-dimensional terms. The coordinate transformation itself can be conveniently done automatically (during the fit) by introduction of an additional NN layer with linear neurons. A non-linear layer can also be used, but we showed that the universal approximator property of the form of Eq. (3.3.4) holds even with a linear coordinate transformation [117].

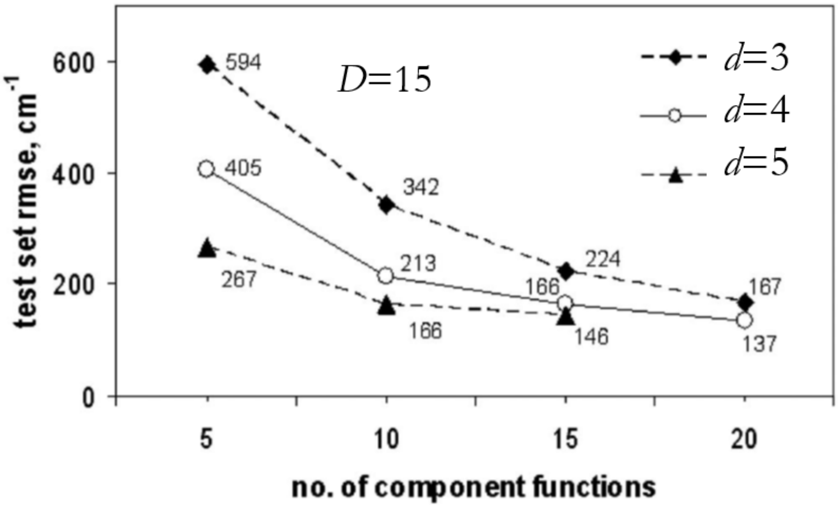

Figure 4. Test set root mean square error (rmse) of the interatomic potential of vinyl bromide as function of both the number and the dimensionality of terms of Red-RS-HDMR-NN. Reproduced with permission from Ref. [117]. Copyright 2010 American Institute of Physics.

3.3.2. RS-HDMR-NN (random sampling high-dimensional model representation neural network)

When the product of N and d in the representation of Eq. (3.3.1.1) is smaller than the dimensionality of the space D, one obtains dimensionality reduction. Specifically, if the coordinate transformation and the component functions are built with neural networks, one obtains effectively a simple version of an autoencoder. An autoencoder is a type of neural network that performs dimensionality reduction, typically by using layers with diminishing numbers of neurons [122].

Dimensionality reduction can be used to find intrinsic dimensionality of the data, which may be different from the dimensionality of the space. We highlighted above that heterogeneous catalysis is an important pillar of emerging energy technologies. Modeling of catalyzed reactions with either classical or quantum dynamics requires interatomic potentials. We used this type of HDMR-inspired dimensionality reduction to construct interatomic potentials. For example, in Ref. [123] the authors constructed with this approach a potential for catalytic decomposition of nitrous oxide on copper, in the frozen surface approximation, and computed dissociation probabilities with it. This was the first ab initio based interatomic potential for a polyatomic molecule-surface catalytic reaction with full account of all molecular degrees of freedom (intramolecular as well as orientational with respect to the surface). The interatomic potential was built in a 15-dimensional configuration space that allowed preserving symmetries and asymptotic behavior even though the intrinsic dimensionality of this

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

problem was nine. **Figure 5** plots the error (on a test set) in the interatomic potential as a function of dimensionality of a single used component function (i.e. N=1). The method allows correctly identifying intrinsic dimensionality as a value of d beyond which there is no improvement, and it allows building the function f sampled with only about 2 data per degree of freedom, with no overfitting [123].

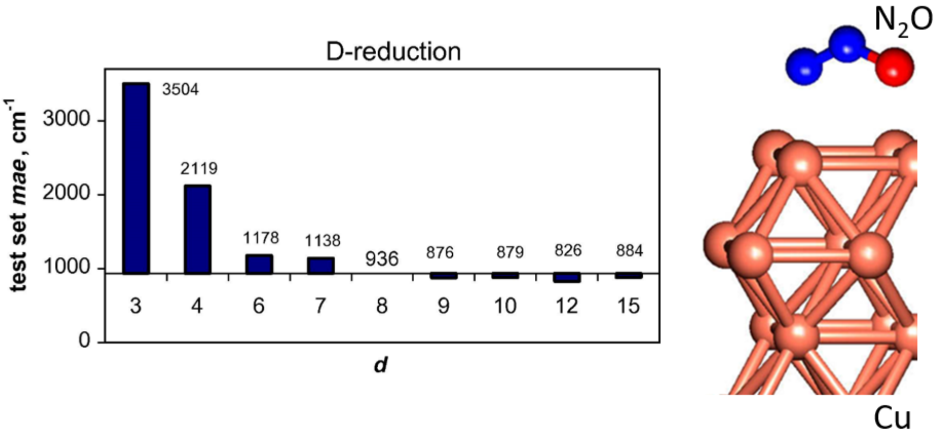

Figure 5. Test set mean absolute error (mae) of the interatomic potential of N_2O over Cu (the system shown on the right) as a function of the dimensionality of a single component function of Red-RS-HDMR-NN. Reproduced with permission from Ref. [123]. Copyright 2010 Elsevier.

3.3.3. RS-HDMR-GPR (random sampling high-dimensional model representation Gaussian process regression)

The previous example was about using a combination of HDMR with neural networks. Here we give examples of using a combination of HDMR with Gaussian process regression. The first example has to do with the future of nuclear energy. Another example of HMDR-NN will be cited in the next section. Nuclear energy remains a great source of stable baseload electricity which is needed in the context of intermittent sources such as solar and wind. A typical nuclear fuel cycle includes uranium mining (of uranium oxides), conversion to UF_6 gas, enrichment of the $^{235}UF_6$ fraction, conversion to solid uranium oxide and then production of nuclear fuel assemblies. A key step in this cycle is uranium enrichment to bring the fraction of uranium-235 from the naturally occurring 0.7% to 3-5% (about 00.3% of naturally occurring U is ^{238}U) [124]. Not all reactor types require enrichment but most reactors in the world do, while other applications (defense) require much higher enrichment degrees. The enrichment is typically done in the gaseous phase, by enriching UF_6 gas in centrifuges. This is a costly process, the enrichment cost can account for about 10% of the electricity cost [125]. For many years now, researchers and the industry have been studying laser-driven enrichment whereby one excites isotope-sensitive hyperfine transitions either of uranium or uranium hexafluoride [126,127]. This requires very high laser coherency (on the order of 10^5) and brings with it various issues. If one could use instead vibrational transitions in UF_6 some of which are isotopomer-selective (such as the mode at around 628 cm^{-1} which is different by about 0.6 cm^{-1} between $^{235}UF_6$ and $^{238}UF_6$ implying necessary coherency on the order of 10^3) [128], one could use cheaper and less coherent IR lasers.

One of us recently proposed the concept of IR laser-driven isotopomer selective desorption of UF_6 [129]. We computed that UF_6 can be adsorbed on different graphene derivatives with tunable adsorption energy depending on the derivative, and that in the adsorbed state there exists an isotopomer unique vibrational mode which can be used to heat and make desorb the molecules in an isotopomer-selective way, as illustrated in **Figure 9**.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

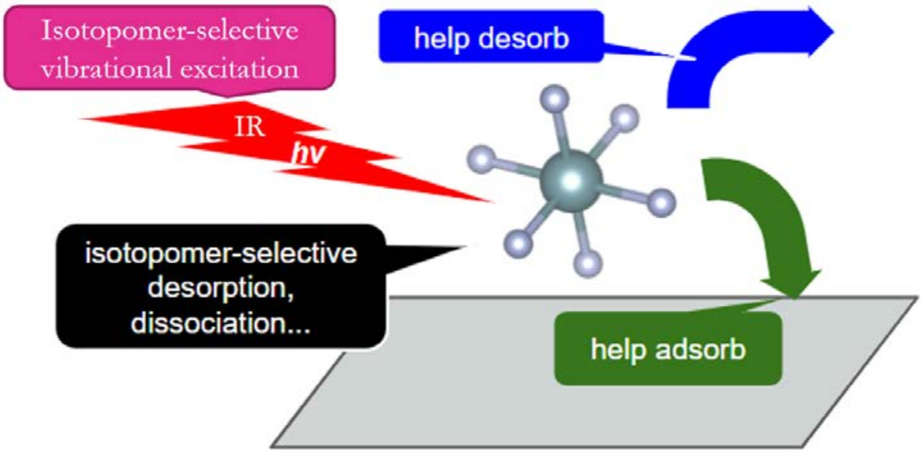
Isotopomer-selective vibrational excitation
IR
hν
help desorb
isotopomer-selective desorption, dissociation...
help adsorb

Figure 6. Principle of IR laser-driven isotopomer-selective desorption of UF_6 proposed in Ref. [129].

Vibrational dynamics of UF_6 would be critical for such a technique, and accurate modeling of vibrational properties and dynamics critical for ability to simulate this process [130]. Unfortunately, good, well-resolved vibrational spectra of UF_6 are not even found in the experimental literature [128,131], and to compute accurate vibrational spectra or vibrational dynamics, one requires a good interatomic potential function (potential energy surface, PES), which for a UF_6 molecule is a 15-dimensional function. Accurate PESs for UF_6 are still unavailable, in particular, due to difficulty of building a 15-dimensional function from sparse data.

In Ref. [105], we applied the HDMR based approach with Gaussian process regression component functions to construct the 15-dimensional interatomic potential of UF_6 as $E^{UF_6}(\boldsymbol{x}) \approx \sum_{i=i}^{N_{cf}} f_i^{GPR}(x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)})$ from samples computed with density functional theory [130]. We trained the model on 2, 3, and 5 thousand data and tested its quality on 50,000 data, i.e. we used a test set much larger that the training set. The results of test set errors obtained with different orders d of HDMR and different numbers of training data are summarized in **Table 1**. More details of the calculations are given in Ref. [105].

Table 1. Test set root mean square error (rmse) when fitting the potential energy surface of UF_6 with HDMR-GPR of different orders d for different numbers of training points N_{train}. For comparison, the results with a full 15-dimensional GPR are also shown. The numbers of component functions N_{cf} at each d are also shown.

rmse[1]	**N_{cf} \ N_{train}:**	**5,000**	**3,000**	**2,000**
Full-D ($d = D$)	1	42.2	75.4	106.7
$d = 1$	15	234.6	236.4	237.3
$d = 2$	105	168.1	178.6	190.3
$d = 3$	455	65.6	78.0	97.4

[1] On the test set of 50,000 points.

If we first look at the fit results with 5,000 data, we see something quite expected: the higher the considered order of coupling, d, the smaller the test error. The smallest error is achieved with a full-dimensional GPR (i.e. $d = D$, $N_{cf} = 1$). Consider now the results obtained with only 2,000 training data. This is a very sparse dataset, with sampling density of only about 1.7 data per dimension. What we observe here is that the fit with three-dimensional functions gives a better test error than the full-dimensional fit. This is because with low density of sampling, it is impossible to recover the full D-dimensional function [120]. The data points were sampled quasi-randomly [132] in the 15-dimensional space, they do not lie on sub-dimensional manifolds. But the information to recover the full D-dimensional terms is just not there. Low data density also increases the danger of overfitting. All this together argues for representations with lower-dimensional functions such as those based on

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

HDMR. Ultimately this has to do with the fact that a finite-size dataset in a *D*-dimensional space is not a D-dimensional object but has a dimension anywhere between 0 and *D* [83,84].

3.4. When are deep NNs useful?

Finally let us touch on the subject of so-called deep neural networks which have now been widely popularized ("deep learning"). We consider again the problem of kinetic energy functionals for orbital-free DFT, of which a good solution would significantly enhance researchers' capabilities to perform large-scale ab initio simulations materials. We are interested in the electronic kinetic energy or kinetic energy density which we know how to express through the orbitals ϕ_i, with the expression of Eq. (2.2.2), but want to express as a function of density only. We can use any derivatives or powers of ρ or any other functions of ρ but without reference to ϕ_i: $T = T\left[\rho(\boldsymbol{x})|\nabla\rho, \Delta\rho, \rho^n, \ldots\right]$ or $\tau = \tau\left[\rho(\boldsymbol{x})|\nabla\rho, \Delta\rho, \rho^n, \ldots\right]$. As an example, we show in **Figure 7** how the kinetic energy density $\tau(x)$ looks like for aluminum, magnesium, and silicon crystals:

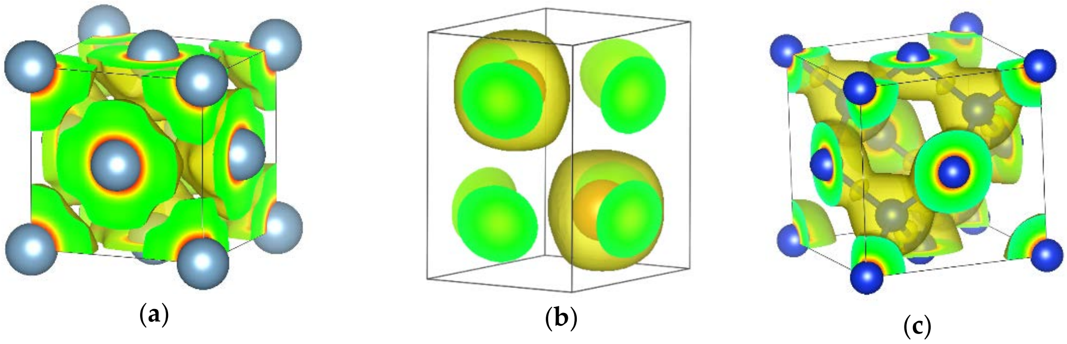

Figure 7. Kinetic energy densities within the unit cells of (a) fcc Al, (b) hcp Mg, and (c) cubic diamond Si crystals.

It is very difficult or impossible to derive analytically an expression for the KEF accurate enough for use in applied simulations of most materials [77]. There is now much progress in machine learning this dependence, in particular, with neural networks but also with other methods, including GPR and some of the methods mentioned earlier [41,78–82]. This problem is a very stringent test for machine learning methods because the accuracy required here is very high, on the order of a thousandth of a per cent unless there is significant error cancellation. We consider NN-based learning of $\tau[\rho(\boldsymbol{x})]$.

Figure 8 shows one-dimensional cuts of the kinetic energy densities of bcc Li, hcp Mg, fcc Al, and cubic diamond Si along selection directions in these crystals, computed in Ref. [78]. The black lines are for Kohn-Sham kinetic energy density that we want to machine-learn. Neural networks were trained in a five-dimensional space of density dependent variables which were terms of the fourth-order gradient expansion [133]:

$$\begin{gathered}
T_4 = T^{(0)} + T^{(2)} + T^{(4)} \\
T^{(0)} \equiv \int \tau_0(\boldsymbol{r})d\boldsymbol{r} = \frac{3}{10}(3\pi^2)^{\frac{2}{3}} \int \rho^{\frac{5}{3}}(\boldsymbol{r})d\boldsymbol{r} \\
T^{(2)} \equiv \int \tau_2(\boldsymbol{r})d\boldsymbol{r} = \frac{1}{72}\int \frac{|\nabla\rho(\boldsymbol{r})|^2}{\rho(\boldsymbol{r})}d\boldsymbol{r} \qquad (3.4.1) \\
T^{(4)} \equiv \int \tau_4(\boldsymbol{r})d\boldsymbol{r} = \frac{(3\pi^2)^{-\frac{2}{3}}}{540}\int \rho^{\frac{1}{3}}\left[\left(\frac{\Delta\rho}{\rho}\right)^2 - \frac{9}{8}\left(\frac{\Delta\rho}{\rho}\right)\left(\frac{\nabla\rho}{\rho}\right)^2 + \frac{1}{3}\left(\frac{\nabla\rho}{\rho}\right)^4\right]d\boldsymbol{r}
\end{gathered}$$

The five summands in the integrands served as density-dependent variables (hence "T4" in **Figure 8**).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

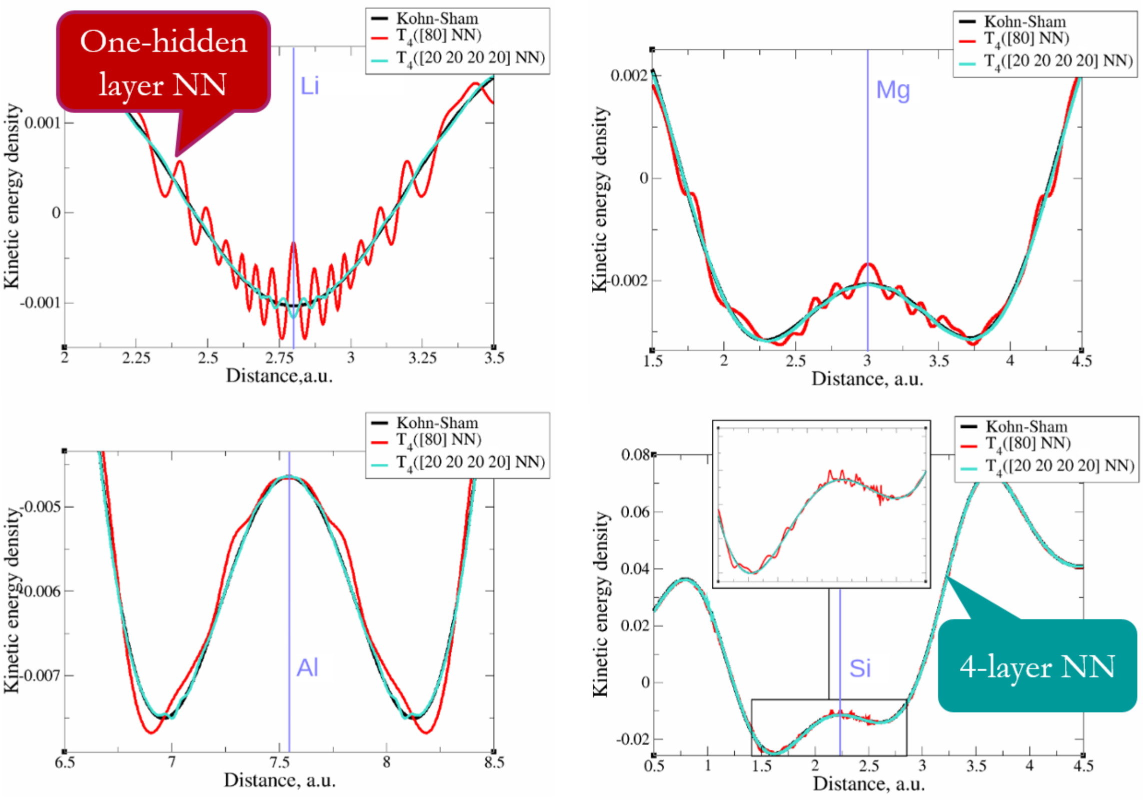

Figure 8. One-dimensional cuts of the kinetic energy densities of bcc Li (top left), hcp Mg (top right), fcc Al (bottom left), and cubic diamond Si (bottom right) along selected directions in the crystal lattice. The target Kohn-Sham kinetic energy density is shown as a black line, results of a single-hidden layer NN fit of the KEDs of all materials simultaneously with a red line ("[80] NN", where 80 is the number of neurons in the hidden layer), and the results of a four-hidden hidden layer NN fit of the KEDs of all materials simultaneously with a turquoise line ("[20 20 20 20] NN", where 20 is the number of neurons in each hidden layer). See Ref. [78] for details.

With single hidden layer neural networks, we could fit accurately the data for each of the materials separately, visually overlapping with the black curve [78]. The red lines are from attempts to fit the kinetic energy densities of all four materials simultaneously with a single hidden layer NN. It is important to be able to do so to make sure that the resulting expression of the kinetic energy as a function of density has certain portability across many materials. The result is obviously not good. With a multilayer neural network, however, we could get a good fit simultaneously for all materials, shown in **Figure 8** by the turquoise lines for a four-hidden layer NN with 20 neurons per layer [78]. This difference in the quality of the fit is not due to an inadequate number of neurons – no single-hidden layer NN was able to learn the KED of all the materials simultaneously.

A single hidden layer NN is a universal approximator, and indeed we saw no advantage of using multilayer networks when fitting individual materials, and we also saw no such advantage in our prior works on interatomic potentials [134,135]. One difference between those works and this case is the extremely uneven distribution of data. To illustrate this, we show in **Figure 9** distributions (histograms) of the target function values, i.e. kinetic energy density τ (and τ_+, see Eq. (2.2.2)) and of some of the density dependent variables we used [41]. In **Figure 9**, $p = \frac{|\nabla\rho|^2}{4(3\pi^2)^{2/3}\rho^{8/3}}$ is the scaled (to satisfy the so-called exact conditions [136]) squared gradient and $q = \frac{\Delta\rho}{4(3\pi^2)^{2/3}\rho^{5/3}}$ is the scaled Laplacian of the density, TF is for $\tau_{TF}(\boldsymbol{r}) = \frac{3}{10}(3\pi^2)^{2/3}\rho^{5/3}(\boldsymbol{r})$ - the Thomas-Fermi KED [137], and vW is for $\tau_{vW}(\mathrm{r}) = \frac{1}{8}\frac{|\nabla\rho(\boldsymbol{r})|^2}{\rho(\boldsymbol{r})}$ – the von Weiszacker KED [138]. The KED distribution is very uneven.

The distributions of the density-dependent variables are in some cases extremely uneven. What it means is that there are vast parts of the space which are extremely sparsely sampled, but which are still important for the quality of the model. And here we clearly see the advantage of a deep NN. Data distribution is an issue that still needs to be better addressed in machine learning. Just using weighted fitting is not sufficient, as we also saw in our research [41,78].

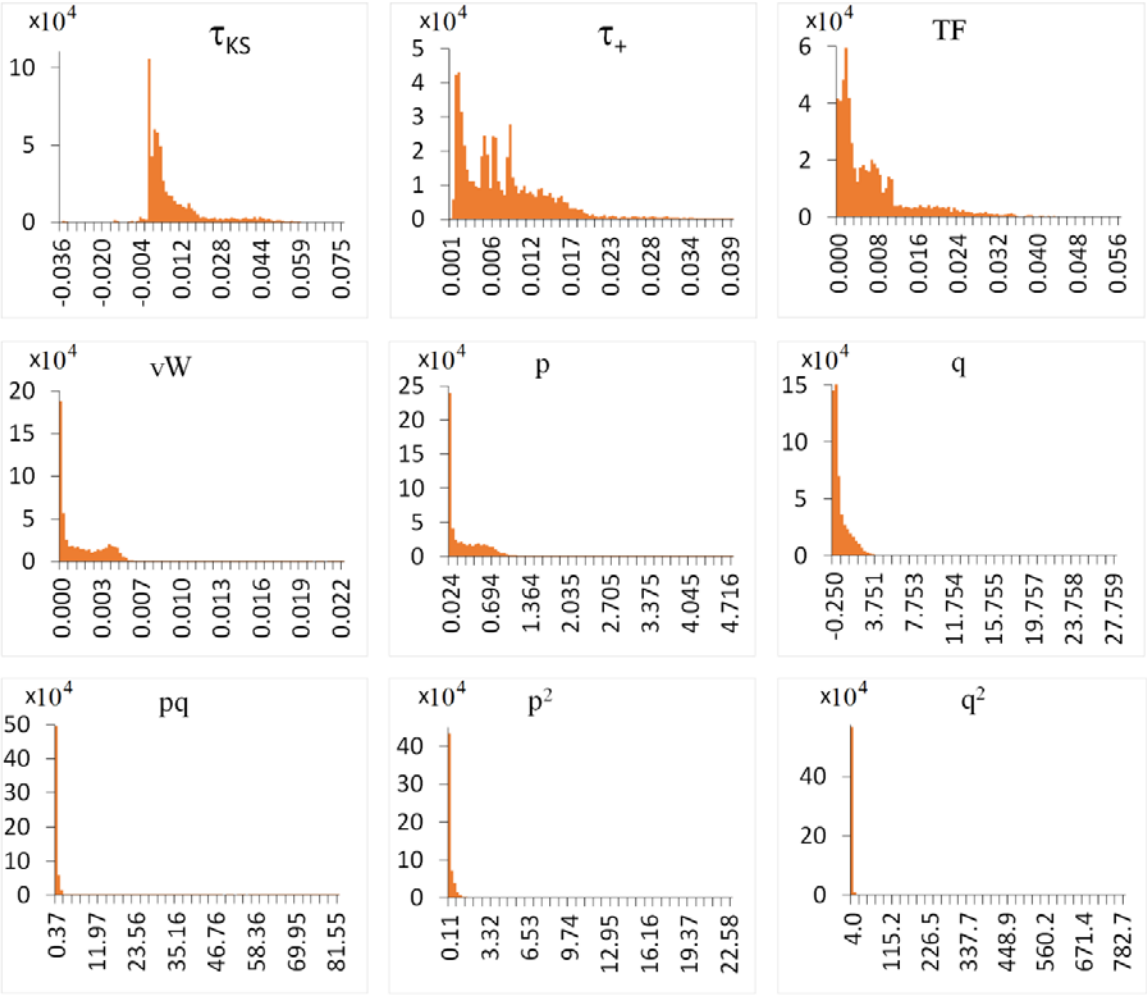

Figure 9. Distributions (histograms) of the kinetic energy densities and density dependent variables in a dataset combining data from Al, Mg, and Si at equilibrium geometry as well as under uniform compression and extension. Adapted with permission from Ref. [41]. Copyright 2020 American Institute of Physics.

When machine learning the KED from the data whose distribution is shown in **Figure 9**, we also confirmed that the HDMR-GPR combination is able to inform on relative importance of different combinations of variables (based on Eq. (3.3.1.3)) [106]. This is a capability which goes beyond the automated relevance determination (ARD) used with GPR, whereby the inverse of the optimal length scale parameter in GPR (l in Eq. (3.1.2.3)) with a squared exponential kernel can be used to determine the importance of different variables [100]. With HDMR-GPR, it is the relative importance of different variable *combinations* which can be estimated. As non-important variable combinations can be omitted, this provides a possibility to prune the number of HDMR terms, which is important to address a key disadvantage of HDMR (i.e. the combinatorial scaling of the number of terms). The knowledge of relative importance of different combinations of descriptors can be used for KEF development with other methods, not necessarily ML-based, including analytic methods.

4. Discussion and conclusions

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

Machine learning today is very widely used to assist in the development of different technologies, including energy technologies. Especially novel technologies based on fuel cells, advanced solar cells etc. are very "knowledge-deep" i.e. they require multiple steps in understanding and development. This includes modeling at the materials level, device level, and system level, and at all these levels machine learning is useful, usually for prediction of material or device properties and performance parameters from descriptors. What we attempted to showcase in this perspective is that as far as the uses of ML for the benefit of energy technologies are concerned, there is more than meets the eye in: many more aspects are being addressed with machine learning than just predicting material or device properties from descriptors. Much is being done at the back-end by using machine learning to construct objects – such as interatomic potentials – which are part of the workflow of understanding and modeling of materials and phenomena for these technologies, or to improve modeling capabilities. This is important in particular for emergent energy technologies, as they require quantum mechanics-based understanding and modeling which is difficult and costly but can be helped with machine learning.

We saw that when one learns dependences in multidimensional spaces, one always works with sparse data, and in this regime machine learning methods are effective because they avoid direct product representations. We used neural networks and Gaussian process regressions in our work. When one uses (regression type) NNs, there is no need to be restricted to the commonly used sigmoid neurons, one can get some useful properties like sum-of-products with other types of activation functions. When the sampling is sparse, the dimensionality of the data is typically lower than the dimensionality of the space, and it may be impossible to reconstruct the full-dimensional function. The danger of overfitting is also then increased. We reviewed combined methods which represent a multidimensional function with low-dimensional functions of any desired dimensionality, and those component functions are built with either NN or GPR and potentially could be built with other methods. This allows working with very sparse datasets without overfitting and it allows using machine learning methods within their "comfort zone".

Much is being talked about deep learning and deep networks. One should not forget that a single hidden layer NN is already a universal approximator. In practice though, when the distribution of the data is very uneven, we found that deep NNs *are* very useful. In other applications, where data were more uniformly distributed, we did not see an advantage with deep NN. In general, the issue of extremely uneven data distributions in some applications is a problem that is yet to be solved in substance. Going forward, we therefore expect methods that go beyond commonly used tools like NN and GPR and use them instead as building blocks of more involved and powerful methods to gain increased attention and use, as well as methods which are specifically tailored to address the issues of data distribution and sparsity, and the issue of missing data.

Author Contributions: Conceptualization and methodology, S.M.; writing—original draft preparation, S.M.; writing—review and editing, S.M. and M.I.; project administration, S.M. and M.I.

Funding: This research was funded by NAME OF FUNDER, grant number XXX.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tong, Q.; Gao, P.; Liu, H.; Xie, Y.; Lv, J.; Wang, Y.; Zhao, J. Combining Machine Learning Potential and Structure Prediction for Accelerated Materials Design and Discovery. *J. Phys. Chem. Lett.* **2020**, *11*, 8710–8720, doi:10.1021/acs.jpclett.0c02357.
2. Walters, W.P.; Barzilay, R. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Acc. Chem. Res.* **2021**, *54*, 263–270, doi:10.1021/acs.accounts.0c00699.
3. Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine Learning in Materials Informatics: Recent Applications and Prospects. *npj Comput Mater* **2017**, *3*, 1–13, doi:10.1038/s41524-017-0056-5.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

4. Wang, A.Y.-T.; Murdock, R.J.; Kauwe, S.K.; Oliynyk, A.O.; Gurlo, A.; Brgoch, J.; Persson, K.A.; Sparks, T.D. Machine Learning for Materials Scientists: An Introductory Guide toward Best Practices. *Chem. Mater.* **2020**, *32*, 4954–4965, doi:10.1021/acs.chemmater.0c01907.
5. Butler, K.T.; Davies, D.W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559*, 547–555, doi:10.1038/s41586-018-0337-2.
6. Moosavi, S.M.; Jablonka, K.M.; Smit, B. The Role of Machine Learning in the Understanding and Design of Materials. *J. Am. Chem. Soc.* **2020**, *142*, 20273–20287, doi:10.1021/jacs.0c09105.
7. Cueto, M. del; Troisi, A. Determining Usefulness of Machine Learning in Materials Discovery Using Simulated Research Landscapes. *Phys. Chem. Chem. Phys.* **2021**, *23*, 14156–14163, doi:10.1039/D1CP01761F.
8. Kalidindi, S.R. Feature Engineering of Material Structure for AI-Based Materials Knowledge Systems. *Journal of Applied Physics* **2020**, *128*, 041103, doi:10.1063/5.0011258.
9. Li, S.; Liu, Y.; Chen, D.; Jiang, Y.; Nie, Z.; Pan, F. Encoding the Atomic Structure for Machine Learning in Materials Science. *WIREs Computational Molecular Science*, doi:10.1002/wcms.1558.
10. Im, J.; Lee, S.; Ko, T.-W.; Kim, H.W.; Hyon, Y.; Chang, H. Identifying Pb-Free Perovskites for Solar Cells by Machine Learning. *npj Comput Mater* **2019**, *5*, 1–8, doi:10.1038/s41524-019-0177-0.
11. Meftahi, N.; Klymenko, M.; Christofferson, A.J.; Bach, U.; Winkler, D.A.; Russo, S.P. Machine Learning Property Prediction for Organic Photovoltaic Devices. *npj Comput Mater* **2020**, *6*, 1–8, doi:10.1038/s41524-020-00429-w.
12. Sahu, H.; Ma, H. Unraveling Correlations between Molecular Properties and Device Parameters of Organic Solar Cells Using Machine Learning. *J. Phys. Chem. Lett.* **2019**, *10*, 7277–7284, doi:10.1021/acs.jpclett.9b02772.
13. Zhuo, Y.; Brgoch, J. Opportunities for Next-Generation Luminescent Materials through Artificial Intelligence. *J. Phys. Chem. Lett.* **2021**, *12*, 764–772, doi:10.1021/acs.jpclett.0c03203.
14. Mahmood, A.; Wang, J.-L. Machine Learning for High Performance Organic Solar Cells: Current Scenario and Future Prospects. *Energy Environ. Sci.* **2021**, *14*, 90–105, doi:10.1039/D0EE02838J.
15. Li, F.; Peng, X.; Wang, Z.; Zhou, Y.; Wu, Y.; Jiang, M.; Xu, M. Machine Learning (ML)-Assisted Design and Fabrication for Solar Cells. *Energy & Environmental Materials* **2019**, *2*, 280–291, doi:10.1002/eem2.12049.
16. Wang, C.-I.; Joanito, I.; Lan, C.-F.; Hsu, C.-P. Artificial Neural Networks for Predicting Charge Transfer Coupling. *J. Chem. Phys.* **2020**, *153*, 214113, doi:10.1063/5.0023697.
17. An, N.G.; Kim, J.Y.; Vak, D. Machine Learning-Assisted Development of Organic Photovoltaics via High-Throughput in Situ Formulation. *Energy Environ. Sci.* **2021**, *14*, 3438–3446, doi:10.1039/D1EE00641J.
18. Rodríguez-Martínez, X.; Pascual-San-José, E.; Campoy-Quiles, M. Accelerating Organic Solar Cell Material's Discovery: High-Throughput Screening and Big Data. *Energy Environ. Sci.* **2021**, *14*, 3301–3322, doi:10.1039/D1EE00559F.
19. Priya, P.; Aluru, N.R. Accelerated Design and Discovery of Perovskites with High Conductivity for Energy Applications through Machine Learning. *npj Comput Mater* **2021**, *7*, 1–12, doi:10.1038/s41524-021-00551-3.
20. Srivastava, M.; Howard, J.M.; Gong, T.; Rebello Sousa Dias, M.; Leite, M.S. Machine Learning Roadmap for Perovskite Photovoltaics. *J. Phys. Chem. Lett.* **2021**, *12*, 7866–7877, doi:10.1021/acs.jpclett.1c01961.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

21. Teunissen, J.L.; Da Pieve, F. Molecular Bond Engineering and Feature Learning for the Design of Hybrid Organic–Inorganic Perovskite Solar Cells with Strong Noncovalent Halogen–Cation Interactions. *J. Phys. Chem. C* **2021**, *125*, 25316–25326, doi:10.1021/acs.jpcc.1c07295.
22. Miyake, Y.; Saeki, A. Machine Learning-Assisted Development of Organic Solar Cell Materials: Issues, Analyses, and Outlooks. *J. Phys. Chem. Lett.* **2021**, *12*, 12391–12401, doi:10.1021/acs.jpclett.1c03526.
23. Xu, S.; Liang, J.; Yu, Y.; Liu, R.; Xu, Y.; Zhu, X.; Zhao, Y. Machine Learning-Assisted Discovery of High-Voltage Organic Materials for Rechargeable Batteries. *J. Phys. Chem. C* **2021**, *125*, 21352–21358, doi:10.1021/acs.jpcc.1c06821.
24. Moses, I.A.; Joshi, R.P.; Ozdemir, B.; Kumar, N.; Eickholt, J.; Barone, V. Machine Learning Screening of Metal-Ion Battery Electrode Materials. *ACS Appl. Mater. Interfaces* **2021**, *13*, 53355–53362, doi:10.1021/acsami.1c04627.
25. Chen, A.; Zhang, X.; Chen, L.; Yao, S.; Zhou, Z. A Machine Learning Model on Simple Features for CO2 Reduction Electrocatalysts. *J. Phys. Chem. C* **2020**, *124*, 22471–22478, doi:10.1021/acs.jpcc.0c05964.
26. Schlexer Lamoureux, P.; Winther, K.T.; Garrido Torres, J.A.; Streibel, V.; Zhao, M.; Bajdich, M.; Abild-Pedersen, F.; Bligaard, T. Machine Learning for Computational Heterogeneous Catalysis. *ChemCatChem* **2019**, *11*, 3581–3601, doi:10.1002/cctc.201900595.
27. Back, S.; Yoon, J.; Tian, N.; Zhong, W.; Tran, K.; Ulissi, Z.W. Convolutional Neural Network of Atomic Surface Structures To Predict Binding Energies for High-Throughput Screening of Catalysts. *J. Phys. Chem. Lett.* **2019**, *10*, 4401–4408, doi:10.1021/acs.jpclett.9b01428.
28. Toyao, T.; Maeno, Z.; Takakusagi, S.; Kamachi, T.; Takigawa, I.; Shimizu, K. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catal.* **2020**, *10*, 2260–2297, doi:10.1021/acscatal.9b04186.
29. Li, X.; Paier, W.; Paier, J. Machine Learning in Computational Surface Science and Catalysis: Case Studies on Water and Metal–Oxide Interfaces. *Frontiers in Chemistry* **2020**, *8*.
30. Pablo-García, S.; García-Muelas, R.; Sabadell-Rendón, A.; López, N. Dimensionality Reduction of Complex Reaction Networks in Heterogeneous Catalysis: From Linear-Scaling Relationships to Statistical Learning Techniques. *WIREs Computational Molecular Science* **2021**, *11*, e1540, doi:10.1002/wcms.1540.
31. Li, X.; Chiong, R.; Page, A.J. Group and Period-Based Representations for Improved Machine Learning Prediction of Heterogeneous Alloy Catalysts. *J. Phys. Chem. Lett.* **2021**, *12*, 5156–5162, doi:10.1021/acs.jpclett.1c01319.
32. Wu, D.; Zhang, J.; Cheng, M.-J.; Lu, Q.; Zhang, H. Machine Learning Investigation of Supplementary Adsorbate Influence on Copper for Enhanced Electrochemical CO2 Reduction Performance. *J. Phys. Chem. C* **2021**, *125*, 15363–15372, doi:10.1021/acs.jpcc.1c05004.
33. Palkovits, S. A Primer about Machine Learning in Catalysis – A Tutorial with Code. *ChemCatChem* **2020**, *12*, 3995–4008, doi:10.1002/cctc.202000234.
34. Giordano, L.; Akkiraju, K.; Jacobs, R.; Vivona, D.; Morgan, D.; Shao-Horn, Y. Electronic Structure-Based Descriptors for Oxide Properties and Functions. *Acc. Chem. Res.* **2022**, *55*, 298–308, doi:10.1021/acs.accounts.1c00509.
35. Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136*, B864–B871, doi:10.1103/PhysRev.136.B864.
36. Kohn, W.; Sham, L.J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133–A1138, doi:10.1103/PhysRev.140.A1133.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

37. Rapaport, D.C. *The Art of Molecular Dynamics Simulation*; 2nd ed.; Cambridge University Press: Cambridge, 2004; ISBN 978-0-521-82568-9.
38. Jansen, A.P.J. Kinetic Monte Carlo Algorithms. In *An Introduction to Kinetic Monte Carlo Simulations of Surface Reactions*; Jansen, A.P.J., Ed.; Lecture Notes in Physics; Springer: Berlin, Heidelberg, 2012; pp. 37–71 ISBN 978-3-642-29488-4.
39. Manzhos, S. Machine Learning for the Solution of the Schrödinger Equation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 013002, doi:10.1088/2632-2153/ab7d30.
40. Behler, J. Perspective: Machine Learning Potentials for Atomistic Simulations. *J. Chem. Phys.* **2016**, *145*, 170901, doi:10.1063/1.4966192.
41. Manzhos, S.; Golub, P. Data-Driven Kinetic Energy Density Fitting for Orbital-Free DFT: Linear vs Gaussian Process Regression. *J. Chem. Phys.* **2020**, *153*, 074104, doi:10.1063/5.0015042.
42. Kulik, H.J.; et al. Machine Learning in Electronic Structure. *EST accepted*.
43. Duan, C.; Liu, F.; Nandy, A.; Kulik, H.J. Putting Density Functional Theory to the Test in Machine-Learning-Accelerated Materials Discovery. *J. Phys. Chem. Lett.* **2021**, *12*, 4628–4637, doi:10.1021/acs.jpclett.1c00631.
44. Friederich, P.; Häse, F.; Proppe, J.; Aspuru-Guzik, A. Machine-Learned Potentials for next-Generation Matter Simulations. *Nat. Mater.* **2021**, *20*, 750–761, doi:10.1038/s41563-020-0777-6.
45. Statistical Review of World Energy | Energy Economics | Home Available online: https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html (accessed on 7 February 2022).
46. Olah, G.A.; Prakash, G.K.S.; Goeppert, A. Anthropogenic Chemical Carbon Cycle for a Sustainable Future. *J. Am. Chem. Soc.* **2011**, *133*, 12881–12898, doi:10.1021/ja202642y.
47. Nayak, P.K.; Mahesh, S.; Snaith, H.J.; Cahen, D. Photovoltaic Solar Cell Technologies: Analysing the State of the Art. *Nat Rev Mater* **2019**, *4*, 269–285, doi:10.1038/s41578-019-0097-0.
48. Joselin Herbert, G.M.; Iniyan, S.; Sreevalsan, E.; Rajapandian, S. A Review of Wind Energy Technologies. *Renewable and Sustainable Energy Reviews* **2007**, *11*, 1117–1145, doi:10.1016/j.rser.2005.08.004.
49. What Are Batteries, Fuel Cells, and Supercapacitors? | Chemical Reviews Available online: https://pubs.acs.org/doi/10.1021/cr020730k (accessed on 7 February 2022).
50. Birdja, Y.Y.; Pérez-Gallent, E.; Figueiredo, M.C.; Göttle, A.J.; Calle-Vallejo, F.; Koper, M.T.M. Advances and Challenges in Understanding the Electrocatalytic Conversion of Carbon Dioxide to Fuels. *Nat Energy* **2019**, *4*, 732–745, doi:10.1038/s41560-019-0450-y.
51. Detz, R.J.; Reek, J.N.H.; Zwaan, B.C.C. van der The Future of Solar Fuels: When Could They Become Competitive? *Energy Environ. Sci.* **2018**, *11*, 1653–1669, doi:10.1039/C8EE00111A.
52. Barnhart, C.J.; Benson, S.M. On the Importance of Reducing the Energetic and Material Demands of Electrical Energy Storage. *Energy Environ. Sci.* **2013**, *6*, 1083–1092, doi:10.1039/C3EE24040A.
53. Winter, M.; Barnett, B.; Xu, K. Before Li Ion Batteries. *Chem. Rev.* **2018**, *118*, 11433–11456, doi:10.1021/acs.chemrev.8b00422.
54. Abram, T.; Ion, S. Generation-IV Nuclear Power: A Review of the State of the Science. *Energy Policy* **2008**, *36*, 4323–4330, doi:10.1016/j.enpol.2008.09.059.
55. Ho, M.; Obbard, E.; Burr, P.A.; Yeoh, G. A Review on the Development of Nuclear Power Reactors. *Energy Procedia* **2019**, *160*, 459–466, doi:10.1016/j.egypro.2019.02.193.
56. Suman, S. Hybrid Nuclear-Renewable Energy Systems: A Review. *Journal of Cleaner Production* **2018**, *181*, 166–177, doi:10.1016/j.jclepro.2018.01.262.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

57. Shao, M.; Chang, Q.; Dodelet, J.-P.; Chenitz, R. Recent Advances in Electrocatalysts for Oxygen Reduction Reaction. *Chem. Rev.* **2016**, *116*, 3594–3657, doi:10.1021/acs.chemrev.5b00462.
58. Jahangiri, H.; Bennett, J.; Mahjoubi, P.; Wilson, K.; Gu, S. A Review of Advanced Catalyst Development for Fischer–Tropsch Synthesis of Hydrocarbons from Biomass Derived Syn-Gas. *Catal. Sci. Technol.* **2014**, *4*, 2210–2229, doi:10.1039/C4CY00327F.
59. Chen, W.-H.; Chen, C.-Y. Water Gas Shift Reaction for Hydrogen Production and Carbon Dioxide Capture: A Review. *Applied Energy* **2020**, *258*, 114078, doi:10.1016/j.apenergy.2019.114078.
60. Chen, L.; Qi, Z.; Zhang, S.; Su, J.; Somorjai, G.A. Catalytic Hydrogen Production from Methane: A Review on Recent Progress and Prospect. *Catalysts* **2020**, *10*, 858, doi:10.3390/catal10080858.
61. Lavoie, J.-M. Review on Dry Reforming of Methane, a Potentially More Environmentally-Friendly Approach to the Increasing Natural Gas Exploitation. *Frontiers in Chemistry* **2014**, *2*.
62. Jain, A.; Ong, S.P.; Hautier, G.; Chen, W.; Richards, W.D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; et al. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Materials* **2013**, *1*, 011002, doi:10.1063/1.4812323.
63. Liu, M.; Kitchin, J.R. SingleNN: Modified Behler–Parrinello Neural Network with Shared Weights for Atomistic Simulations with Transferability. *J. Phys. Chem. C* **2020**, *124*, 17811–17818, doi:10.1021/acs.jpcc.0c04225.
64. Behler, J. Constructing High-Dimensional Neural Network Potentials: A Tutorial Review. *International Journal of Quantum Chemistry* **2015**, *115*, 1032–1050, doi:10.1002/qua.24890.
65. Na, G.S.; Jang, S.; Lee, Y.-L.; Chang, H. Tuplewise Material Representation Based Machine Learning for Accurate Band Gap Prediction. *J. Phys. Chem. A* **2020**, *124*, 10616–10623, doi:10.1021/acs.jpca.0c07802.
66. Xu, P.; Lu, T.; Ju, L.; Tian, L.; Li, M.; Lu, W. Machine Learning Aided Design of Polymer with Targeted Band Gap Based on DFT Computation. *J. Phys. Chem. B* **2021**, *125*, 601–611, doi:10.1021/acs.jpcb.0c08674.
67. Aykol, M.; Herring, P.; Anapolsky, A. Machine Learning for Continuous Innovation in Battery Technologies. *Nat Rev Mater* **2020**, *5*, 725–727, doi:10.1038/s41578-020-0216-y.
68. Deringer, V.L. Modelling and Understanding Battery Materials with Machine-Learning-Driven Atomistic Simulations. *J. Phys. Energy* **2020**, *2*, 041003, doi:10.1088/2515-7655/abb011.
69. Thomas, J.K.; Crasta, H.R.; Kausthubha, K.; Gowda, C.; Rao, A. Battery Monitoring System Using Machine Learning. *Journal of Energy Storage* **2021**, *40*, 102741, doi:10.1016/j.est.2021.102741.
70. Li, W.; Cui, H.; Nemeth, T.; Jansen, J.; Ünlübayir, C.; Wei, Z.; Zhang, L.; Wang, Z.; Ruan, J.; Dai, H.; et al. Deep Reinforcement Learning-Based Energy Management of Hybrid Battery Systems in Electric Vehicles. *Journal of Energy Storage* **2021**, *36*, 102355, doi:10.1016/j.est.2021.102355.
71. Elkamel, M.; Schleider, L.; Pasiliao, E.L.; Diabat, A.; Zheng, Q.P. Long-Term Electricity Demand Prediction via Socioeconomic Factors—A Machine Learning Approach with Florida as a Case Study. *Energies* **2020**, *13*, 3996, doi:10.3390/en13153996.
72. Krishnadas, G.; Kiprakis, A. A Machine Learning Pipeline for Demand Response Capacity Scheduling. *Energies* **2020**, *13*, 1848, doi:10.3390/en13071848.
73. Nti, I.K.; Teimeh, M.; Nyarko-Boateng, O.; Adekoya, A.F. Electricity Load Forecasting: A Systematic Review. *Journal of Electrical Systems and Information Technology* **2020**, *7*, 13, doi:10.1186/s43067-020-00021-8.
74. Antonopoulos, I.; Robu, V.; Couraud, B.; Kirli, D.; Norbu, S.; Kiprakis, A.; Flynn, D.; Elizondo-Gonzalez, S.; Wattam, S. Artificial Intelligence and Machine Learning Approaches to Energy Demand-Side

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

Response: A Systematic Review. *Renewable and Sustainable Energy Reviews* **2020**, *130*, 109899, doi:10.1016/j.rser.2020.109899.

75. Kim, J.Y.; Lee, J.-W.; Jung, H.S.; Shin, H.; Park, N.-G. High-Efficiency Perovskite Solar Cells. *Chem. Rev.* **2020**, *120*, 7867–7918, doi:10.1021/acs.chemrev.0c00107.
76. Pham, H.D.; Xianqiang, L.; Li, W.; Manzhos, S.; Kyaw, A.K.K.; Sonar, P. Organic Interfacial Materials for Perovskite-Based Optoelectronic Devices. *Energy Environ. Sci.* **2019**, *12*, 1177–1209, doi:10.1039/C8EE02744G.
77. Witt, W.C.; Rio, B.G. del; Dieterich, J.M.; Carter, E.A. Orbital-Free Density Functional Theory for Materials Research. *Journal of Materials Research* **2018**, *33*, 777–795, doi:10.1557/jmr.2017.462.
78. Golub, P.; Manzhos, S. Kinetic Energy Densities Based on the Fourth Order Gradient Expansion: Performance in Different Classes of Materials and Improvement via Machine Learning. *Phys. Chem. Chem. Phys.* **2018**, *21*, 378–395, doi:10.1039/C8CP06433D.
79. Fujinami, M.; Kageyama, R.; Seino, J.; Ikabata, Y.; Nakai, H. Orbital-Free Density Functional Theory Calculation Applying Semi-Local Machine-Learned Kinetic Energy Density Functional and Kinetic Potential. *Chemical Physics Letters* **2020**, *748*, 137358, doi:10.1016/j.cplett.2020.137358.
80. Seino, J.; Kageyama, R.; Fujinami, M.; Ikabata, Y.; Nakai, H. Semi-Local Machine-Learned Kinetic Energy Density Functional Demonstrating Smooth Potential Energy Curves. *Chemical Physics Letters* **2019**, *734*, 136732, doi:10.1016/j.cplett.2019.136732.
81. Snyder, J.C.; Rupp, M.; Hansen, K.; Blooston, L.; Müller, K.-R.; Burke, K. Orbital-Free Bond Breaking via Machine Learning. *J. Chem. Phys.* **2013**, *139*, 224104, doi:10.1063/1.4834075.
82. Yao, K.; Parkhill, J. Kinetic Energy of Hydrocarbons as a Function of Electron Density and Convolutional Neural Networks. *J. Chem. Theory Comput.* **2016**, *12*, 1139–1147, doi:10.1021/acs.jctc.5b01011.
83. Hausdorff, F. Dimension und äußeres Maß. *Math. Ann.* **1918**, *79*, 157–179, doi:10.1007/BF01457179.
84. Kak, S. Information Theory and Dimensionality of Space. *Sci Rep* **2020**, *10*, 20733, doi:10.1038/s41598-020-77855-9.
85. Montavon, G.; Orr, G.B.; Mueller, K.-R. *Neural Networks: Tricks of the Trade*; 2nd ed.; Springer: Berlin Heidelberg, 2012; ISBN 978-3-642-35288-1.
86. Kolmogorov, A.N. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. *Dokl. Akad. Nauk SSSR* **1957**, *114*, 953–856.
87. Sprecher, D.A. A Numerical Implementation of Kolmogorov's Superpositions II. *Neural Networks* **1997**, *10*, 447–457, doi:10.1016/S0893-6080(96)00073-1.
88. Sprecher, D.A. A Numerical Implementation of Kolmogorov's Superpositions. *Neural Networks* **1996**, *9*, 765–772, doi:10.1016/0893-6080(95)00081-X.
89. Sprecher, D.A.; Draghici, S. Space-Filling Curves and Kolmogorov Superposition-Based Neural Networks. *Neural Networks* **2002**, *15*, 57–67, doi:10.1016/S0893-6080(01)00107-1.
90. Nees, M. Approximative Versions of Kolmogorov's Superposition Theorem, Proved Constructively. *Journal of Computational and Applied Mathematics* **1994**, *54*, 239–250, doi:10.1016/0377-0427(94)90179-1.
91. Katsuura, H.; Sprecher, D.A. Computational Aspects of Kolmogorov's Superposition Theorem. *Neural Networks* **1994**, *7*, 455–461, doi:10.1016/0893-6080(94)90079-5.
92. Sprecher, D.A. A Universal Mapping for Kolmogorov's Superposition Theorem. *Neural Networks* **1993**, *6*, 1089–1094, doi:10.1016/S0893-6080(09)80020-8.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

93. Kůrková, V. Kolmogorov's Theorem and Multilayer Neural Networks. *Neural Networks* **1992**, *5*, 501–506, doi:10.1016/0893-6080(92)90012-8.
94. Hornik, K. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* **1991**, *4*, 251–257, doi:10.1016/0893-6080(91)90009-T.
95. Hornik, K.; Stinchcombe, M.; White, H. Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks. *Neural Networks* **1990**, *3*, 551–560, doi:10.1016/0893-6080(90)90005-6.
96. Gorban, A.N. Approximation of Continuous Functions of Several Variables by an Arbitrary Nonlinear Continuous Function of One Variable, Linear Functions, and Their Superpositions. *Applied Mathematics Letters* **1998**, *11*, 45–49, doi:10.1016/S0893-9659(98)00032-9.
97. Manzhos, S.; Carrington, T. Using Neural Networks to Represent Potential Surfaces as Sums of Products. *J. Chem. Phys.* **2006**, *125*, 194105, doi:10.1063/1.2387950.
98. Beck, M.H.; Jäckle, A.; Worth, G.A.; Meyer, H.-D. The Multiconfiguration Time-Dependent Hartree (MCTDH) Method: A Highly Efficient Algorithm for Propagating Wavepackets. *Physics Reports* **2000**, *324*, 1–105, doi:10.1016/S0370-1573(99)00047-2.
99. Schmitt, M. On the Complexity of Computing and Learning with Multiplicative Neural Networks. *Neural Computation* **2002**, *14*, 241–301, doi:10.1162/08997660252741121.
100. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge MA, USA, 2006; ISBN 0-262-18253-X.
101. Genton, M.G. Classes of Kernels for Machine Learning: A Statistics Perspective. *Journal of Machine Learning Research* **2001**, *2*, 299–312.
102. Smola, A.; Bartlett, P. Sparse Greedy Gaussian Process Regression. In Proceedings of the Advances in Neural Information Processing Systems; MIT Press, 2001; Vol. 13.
103. Kamath, A.; Vargas-Hernández, R.A.; Krems, R.V.; Carrington, T.; Manzhos, S. Neural Networks vs Gaussian Process Regression for Representing Potential Energy Surfaces: A Comparative Study of Fit Quality and Vibrational Spectrum Accuracy. *J. Chem. Phys.* **2018**, *148*, 241702, doi:10.1063/1.5003074.
104. Neal, R.M. Bayesian Learning for Neural Networks. Ph.D. Thesis, University of Toronto: Toronto, 1995.
105. Boussaidi, M.A.; Ren, O.; Voytsekhovsky, D.; Manzhos, S. Random Sampling High Dimensional Model Representation Gaussian Process Regression (RS-HDMR-GPR) for Multivariate Function Representation: Application to Molecular Potential Energy Surfaces. *J. Phys. Chem. A* **2020**, *124*, 7598–7607, doi:10.1021/acs.jpca.0c05935.
106. Ren, O.; Boussaidi, M.A.; Voytsekhovsky, D.; Ihara, M.; Manzhos, S. Random Sampling High Dimensional Model Representation Gaussian Process Regression (RS-HDMR-GPR) for Representing Multidimensional Functions with Machine-Learned Lower-Dimensional Terms Allowing Insight with a General Method. *Computer Physics Communications* **2021**, 108220, doi:10.1016/j.cpc.2021.108220.
107. Manzhos, S.; Ihara, M. Rectangularization of Gaussian Process Regression for Optimization of Hyperparameters. *arXiv:2112.02467 [cs, math]* **2021**.
108. Li, G.; Rosenthal, C.; Rabitz, H. High Dimensional Model Representations. *J. Phys. Chem. A* **2001**, *105*, 7765–7777, doi:10.1021/jp010450t.
109. Rabitz, H.; Aliş, Ö.F. General Foundations of High-dimensional Model Representations. *Journal of Mathematical Chemistry* **1999**, *25*, 197–233, doi:10.1023/A:1019188517934.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

110. Alış, Ö.F.; Rabitz, H. Efficient Implementation of High Dimensional Model Representations. *Journal of Mathematical Chemistry* **2001**, *29*, 127–142, doi:10.1023/A:1010979129659.

111. Fisher, R.A. On the “Probable Error” of a Coefficient of Correlation Deduced from a Small Sample. *Metron* **1921**, *1*, 3–32.

112. Sobol′, I.M. Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. *Mathematics and Computers in Simulation* **2001**, *55*, 271–280, doi:10.1016/S0378-4754(00)00270-6.

113. Li, G.; Hu, J.; Wang, S.-W.; Georgopoulos, P.G.; Schoendorf, J.; Rabitz, H. Random Sampling-High Dimensional Model Representation (RS-HDMR) and Orthogonality of Its Different Order Component Functions. *J. Phys. Chem. A* **2006**, *110*, 2474–2485, doi:10.1021/jp054148m.

114. Wang, S.-W.; Georgopoulos, P.G.; Li, G.; Rabitz, H. Random Sampling−High Dimensional Model Representation (RS−HDMR) with Nonuniformly Distributed Variables: Application to an Integrated Multimedia/Multipathway Exposure and Dose Model for Trichloroethylene. *J. Phys. Chem. A* **2003**, *107*, 4707–4716, doi:10.1021/jp022500f.

115. Manzhos, S.; Ihara, M. On the Optimization of Hyperparameters in Gaussian Process Regression with the Help of Low-Order High-Dimensional Model Representation. *arXiv:2112.01374 [cs, math, stat]* **2022**.

116. Manzhos, S.; Yamashita, K.; Carrington, T. Extracting Functional Dependence from Sparse Data Using Dimensionality Reduction: Application to Potential Energy Surface Construction. In Proceedings of the Coping with Complexity: Model Reduction and Data Analysis; Gorban, A.N., Roose, D., Eds.; Springer: Berlin, Heidelberg, 2011; pp. 133–149.

117. Manzhos, S.; Carrington, T. Using Redundant Coordinates to Represent Potential Energy Surfaces with Lower-Dimensional Functions. *J. Chem. Phys.* **2007**, *127*, 014103, doi:10.1063/1.2746846.

118. Manzhos, S.; Yamashita, K.; Carrington, T. Fitting Sparse Multidimensional Data with Low-Dimensional Terms. *Computer Physics Communications* **2009**, *180*, 2002–2012, doi:10.1016/j.cpc.2009.05.022.

119. Manzhos, S.; Sasaki, E.; Ihara, M. Easy Representation of Multivariate Functions with Low-Dimensional Terms via Gaussian Process Regression Kernel Design: Applications to Machine Learning of Potential Energy Surfaces and Kinetic Energy Densities from Sparse Data. *Mach. Learn.: Sci. Technol.* **2022**, *3*, 01LT02, doi:10.1088/2632-2153/ac4949.

120. Manzhos, S.; Carrington, T. A Random-Sampling High Dimensional Model Representation Neural Network for Building Potential Energy Surfaces. *J. Chem. Phys.* **2006**, *125*, 084109, doi:10.1063/1.2336223.

121. Duvenaud, D.; Nickisch, H.; Rasmussen, C.E. Additive Gaussian Processes. In Proceedings of the Advances in Neural Information Processing Systems; December 19 2011; pp. 226–234.

122. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, doi:10.1126/science.1127647.

123. Manzhos, S.; Yamashita, K. A Model for the Dissociative Adsorption of N2O on Cu(100) Using a Continuous Potential Energy Surface. *Surface Science* **2010**, *604*, 555–561, doi:10.1016/j.susc.2009.12.025.

124. Wolfsberg, M.; Van Hook, A.; Paneth, P.; Rebelo, L.P.N. *Isotope Effects*; Springer: Dordrecht, 2009;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 March 2022 doi:10.20944/preprints202203.0007.v1

125. Schneider, E.; Carlsen, B.; Tavrides, E.; van der Hoeven, C.; Phathanapirom, U. Measures of the Environmental Footprint of the Front End of the Nuclear Fuel Cycle. *Energy Economics* **2013**, *40*, 898–910, doi:10.1016/j.eneco.2013.01.002.

126. Parvin, P.; Sajad, B.; Silakhori, K.; Hooshvar, M.; Zamanipour, Z. Molecular Laser Isotope Separation versus Atomic Vapor Laser Isotope Separation. *Progress in Nuclear Energy* **2004**, *44*, 331–345, doi:10.1016/j.pnueene.2004.07.002.

127. Ronander, E.; Strydom, H.J.; Botha, L.R. High-Pressure Continuously Tunable CO2 Lasers and Molecular Laser Isotope Separation. *Pramana* **2014**, *82*, 49–58.

128. McDowell, R.S.; Sherman, R.J.; Asprey, L.B.; Kennedy, R.C. Vibrational Spectrum and Force Field of Molybdenum Hexafluoride. *J. Chem. Phys.* **1975**, *62*, 3974–3978, doi:10.1063/1.430320.

129. Koh, Y.W.; Westerman, K.; Manzhos, S. A Computational Study of Adsorption and Vibrations of UF6 on Graphene Derivatives: Conditions for 2D Enrichment. *Carbon* **2015**, *81*, 800–806, doi:10.1016/j.carbon.2014.10.027.

130. Manzhos, S.; Carrington, T.; Laverdure, L.; Mosey, N. Computing the Anharmonic Vibrational Spectrum of UF6 in 15 Dimensions with an Optimized Basis Set and Rectangular Collocation. *J. Phys. Chem. A* **2015**, *119*, 9557–9567, doi:10.1021/acs.jpca.5b07627.

131. Berezin, A.G.; Malyugin, S.L.; Nadezhdinskii, A.I.; Namestnikov, D.Yu.; Ponurovskii, Ya.Ya.; Stavrovskii, D.B.; Shapovalov, Yu.P.; Vyazov, I.E.; Zaslavskii, V.Ya.; Selivanov, Yu.G.; et al. UF6 Enrichment Measurements Using TDLS Techniques. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* **2007**, *66*, 796–802, doi:10.1016/j.saa.2006.10.037.

132. Sobol', I.M. On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *USSR Computational Mathematics and Mathematical Physics* **1967**, *7*, 86–112, doi:10.1016/0041-5553(67)90144-9.

133. Hodges, C.H. Quantum Corrections to the Thomas–Fermi Approximation—The Kirzhnits Method. *Can. J. Phys.* **1973**, *51*, 1428–1437, doi:10.1139/p73-189.

134. Manzhos, S.; Dawes, R.; Carrington, T. Neural Network-Based Approaches for Building High Dimensional and Quantum Dynamics-Friendly Potential Energy Surfaces. *International Journal of Quantum Chemistry* **2015**, *115*, 1012–1020, doi:10.1002/qua.24795.

135. Manzhos, S.; Carrington, T. Neural Network Potential Energy Surfaces for Small Molecules and Reactions. *Chem. Rev.* **2021**, *121*, 10187–10217, doi:10.1021/acs.chemrev.0c00665.

136. Bartlett, R.J.; Ranasinghe, D.S. The Power of Exact Conditions in Electronic Structure Theory. *Chemical Physics Letters* **2017**, *669*, 54–70, doi:10.1016/j.cplett.2016.12.017.

137. Fermi, E. Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente. *Z. Physik* **1928**, *48*, 73–79, doi:10.1007/BF01351576.

138. Weizsäcker, C.F. v. Zur Theorie der Kernmassen. *Z. Physik* **1935**, *96*, 431–458, doi:10.1007/BF01337700.