Artificial Intelligence Chemistry 1 (2023) 100008

Contents lists available at ScienceDirect

Artificial Intelligence Chemistry

journal homepage: www.journals.elsevier.com/artificial-intelligence-chemistry

Orders of coupling representations as a versatile framework for machine learning from sparse data in high-dimensional spaces

Sergei Manzhos [a,*], Tucker Carrington [b,*], Manabu Ihara [a,*]

[a] School of Materials and Chemical Technology, Tokyo Institute of Technology, Ookayama 2–12-1, Meguro-ku, Tokyo 152–8552 Japan
[b] Department of Chemistry, Queen's University, 90 Bader Lane, Kingston, Ontario K7L 3N6 Canada

ARTICLE INFO

Keywords:
Neural network
Gaussian process regression
High-dimensional model representation
Potential energy surface

ABSTRACT

Machine learning (ML) techniques are already widely and increasingly used in diverse applications in science and technology, including computational chemistry. Specifically in computational chemistry, neural networks (NN) and kernel methods such as Gaussian process regressions (GPR) have been increasingly used for the construction of potential functions and functionals for density functional theory. While ML techniques have a number of advantages vs intuition-based models, notably their generality and black-box nature, they are still challenged when faced with high dimensionality of the feature space or low and uneven data density – in part because of their general nature. We review recent works using methods such as NNs and GPR as building blocks of composite methods in the framework of an expansion over orders of coupling. We introduce models using NN or GPR-based components as part of HDMR (high-dimensional model representations)-based structures. HDMR is a formalization of orders-of-coupling representations that include the many-body and N-mode representations well known in computational chemistry and allows, in particular, building all terms from one dataset of arbitrarily distributed data. The resulting HDMR-NN and HDMR-GPR combinations and NN with HDMR-GPR derived neuron activation functions not requiring non-linear optimization enhance machine learning capabilities in high dimensional spaces and or with sparse data.

1. Introduction: difficulties of common machine learning methods in high-dimensional spaces

Machine learning (ML) techniques are already widely and increasingly used in diverse applications in science and technology, including computational chemistry. ML for the construction of potential energy surfaces (PES) is already routine, and ML-based potentials are widely used in applications [1–16]. ML is firmly establishing itself as a way to construct or improve functionals for DFT (density functional theory), including exchange correlation (XC) and kinetic energy functionals (KEF) [3,17–25], and pseudopotentials [26,27]. Elsewhere, ML is increasingly used to predict materials properties from descriptors [28–46]. While ML techniques have a number of advantages vs intuition-based models, notably their generality and black-box nature, they face challenges in high dimensionality of the feature space or with low and uneven data density – in part because of their general nature. In the above applications, one typically works in a high-dimensional feature space. Interatomic potentials are functions of $3N_a$-6 to $3N_a$ coordinates in the configuration space, for an N_a -atom system. While sums over atomic contributions can be used to limit the dimensionality of the feature space for large systems [2,5,6,47–50], it is common to use dozens of descriptors. When machine learning DFT functionals from density-based descriptors, the dimensionality of the feature space D can be substantial. For example, in Refs. [18,51], terms of the fourth-order gradient expansion [52] were used as descriptors, as well as the Kohn-Sham effective potential [53], forming a seven-dimensional feature space. Adding elements of structural information and or other density-derived quantities would further expand the dimensionality [19].

Working in high-dimensional spaces exposes one to the so-called curse of dimensionality [54]. Its most obvious manifestation is the exponential scaling of the required number of datapoints of a direct product grid with dimension for a given density of sampling. Easy-to-use representations of direct product grid type (such as the Fourier expansion or DVR [55] type functions) also then require an exponentially growing with dimension number of terms. Data in high-dimensional spaces are necessarily sparse, and this sparsity cannot practically be addressed by generating more data. In one of the examples below, a

* Corresponding authors.
E-mail addresses: manzhos.s.aa@m.titech.ac.jp (S. Manzhos), tucker.carrington@queensu.ca (T. Carrington), mihara@chemeng.titech.ac.jp (M. Ihara).

https://doi.org/10.1016/j.aichem.2023.100008
Received 31 May 2023; Received in revised form 10 July 2023; Accepted 14 July 2023
Available online 17 July 2023

molecular PES in a 15-dimensional configuration space is sampled with about 50,000 single point energies. This corresponds to about 2.1 data points per degree of freedom (DOF) (of an equivalent direct product grid). Increasing the size of the dataset by a factor of 10 would only increase the sampling density to about 2.4 data per DOF. Using non-direct product representations in principle allows one to avoid the exponential scaling [56,57] but the scaling remains steep. ML methods should therefore be able to reliably work with sparse data. Data sparsity can also exist in low-dimensional settings. Other manifestations of the curse of dimensionality include the loss of the locality property of commonly used Matern type kernels [58] in high-D [59]. This can be intuitively understood on the example of the Gaussian function: its integral over the interval [-σ, σ] (where σ is the standard deviation) is about 70% of its integral over the entire space (-∞, +∞) in 1D but only about 10% in 6D. This has practical consequences, for example, for the utility of multi-zeta bases and of sparsification of matrices (such as the covariance matrix of kernel methods [60–64]) based on locality for gaining computational efficiency, as shown in Ref. [59].

Among machine learning (ML) techniques widely used in applications, and in particular in physical chemistry, materials science, and computational chemistry applications, neural networks (NN) [1,2,4,5,65] and kernel methods [60–64] are arguably among the most popular [3,21,29]. Kernel methods such as GPR (Gaussian process regression) [61,66] or KRR (kernel ridge regression) [60,67] are essentially regularized linear regressions with non-linear basis functions [67–69] but are often treated as ML methods. We refer the reader to the by now abundant literature introducing these methods [60,65–67], including in the context of computational chemistry (for example, Refs. [4,61]), and for this reason we skip introducing them here. ML methods are used to discover input-output mappings without recourse to empirical models; this black-box nature is an advantage because one does not straitjacket the solution by imposing the shape of a potential energy surface, a density functional, a wavefunction, etc., but it typically means that more data are needed to train the model than with empirical models. Even if ML methods like NN or GPR suffer less from the curse of dimensionality than e.g. tensor product polynomial approximations, because they can be viewed as expansions over non-direct product bases [70], eventually they are hit by it as the data in high-dimensional feature spaces are necessarily sparse regardless of the number of datapoints, as highlighted above; they may also be very unevenly distributed [18,51].

The universal approximator quality of an NN [57,71,72] holds assuming one has access to unlimited data. The large, and growing with the dimension of the feature space D, number of non-linear parameters of a feed-forward NN eventually leads to overfitting as well as to limitations due to CPU and memory cost. Kernel methods, being linear regressions, are more robust and scale better with dimension but may also suffer from overfitting or collapse when D becomes very large (e.g. the use Matern type kernels leads to predicted values on the test set drop to 0 as $D\rightarrow\infty$ unless the length parameter becomes so large that any advantage over simple linear regression is negated [70]). They also become costly when the number of data is large, due to the need to use the inverse of the covariance matrix [66,67,73]. Larger training data sets are typically needed in higher-D, even though the governing equations of KRR or GPR do not explicitly depend on D. A finite or low density of sampling puts a limit on the information that can be recovered about the underlying dependence from a dataset [74,75]. Ultimately this can be related to the fact that a finite-size dataset in a D-dimensional space has in general a lower dimensionality than D. The D-dimensional data may be embedded in a lower dimensional manyfold, e.g. have a lower intrinsic dimensionality [76–81]. Even if they are not embedded in a lower-dimensional manifold, the finite nature of the data set (a collection of 0-dimensional points) imparts to it fractional dimensionality, and depending on the data density, a D-dimensional function may not be recoverable [82,83]. In this case, it is reasonable to build representations with lower-dimensional functions. Lower-dimensional functions are easier to build and to use (in particular for integration).

It is convenient to think about this in terms of orders of coupling that can or cannot be recovered from a given dataset. Orders of coupling representations are known to the computational chemistry community mostly via many-body and N-mode expansions [84–86] that are used to avoid calculating multi-dimensional integrals that are a bottleneck in many computational chemistry applications, notably accurate computational spectroscopy and quantum dynamics [87,88]. They have been formalized in the form of high-dimensional model representation (HDMR) in a series of articles by Rabitz and co-workers [89–92]. In this Review, we show that combining orders of coupling representations with machine learning techniques is a potent approach that facilitates machine learning in high-dimensional spaces from sparse data. We review works using combinations of HDMR-based ideas with NN and with GPR as well as works using HDMR-GPR to construct neurons of a NN and to obtain neural networks with architectures realizing an orders of coupling representation. The review necessarily focuses on the authors' work in this area, who have developed the above techniques. We aim to demonstrate that orders of coupling representations coupled with machine learning offer a versatile and potent framework for machine learning from sparse data in high-dimensional spaces where off-the-shelf approaches might fail.

2. High-dimensional model representation as a versatile framework for machine learning from sparse data in high-dimensional spaces

2.1. Orders of coupling representations

The idea that it is useful to represent a D-dimensional function as a sum of terms that depend on one variable plus a sum of terms that depend on two variables plus a sum of terms that depend on three variables etc. is both intuitive and old. In physics and chemistry, the idea is best known as a many-body expansion (MBE). The interaction energy of a system of N_b bodies is a sum of terms [93].

$$E=\sum_{I=1}^{N_b}E_I+\sum_{I}^{N_b}\sum_{J>I}^{N_b}\Delta E_{IJ}+\sum_{I}^{N_b}\sum_{J>I}^{N_b}\sum_{K>J}^{N_b}\Delta E_{IJK}+\ldots \tag{2.1.1}$$

MBEs have been used extensively [94–97]. Here, E_I are two-body interactions dependent on the distance between particles, ΔE_{IJ} are corrections due to three-body interactions dependent on three coordinates describing mutual positions of three particles etc. In general, an orders of coupling representation in the space of some coordinates $\boldsymbol{x}\in R^D$ can be written as

$$f(\boldsymbol{x})\approx f_0+\sum_{i=1}^{D}f_i(x_i)+\sum_{1\le i<j\le D}f_{ij}\left(x_i,x_j\right)+\ldots+\sum_{\{i_1i_2\ldots i_d\}\in\{12\ldots D\}}f_{i_1i_2\ldots i_d}\left(x_{i_1},x_{i_2},\ldots,x_{i_d}\right)+\ldots \tag{2.1.2}$$

Truncating the sum of terms so that it includes all terms up to and including those that depend on d coordinates yields a representation that is exact when $d=D$ and often accurate enough when $d<D$.

When studying the motion of nuclei, it is advantageous to represent a potential energy surface (PES) as a sum of terms each of which depends on a subset of the coordinates; this is known as the N-mode representation [84,98,99]. Similar ideas have been used but with coordinates replaced by groups of coordinates [100,101]. Different authors use different names for the sum of lower dimensional terms idea. In this paper, we shall refer to it as an orders of coupling representation. The terms depending on only one coordinate include no coupling between coordinates. The terms depending on two coordinates include coupling between only two coordinates, etc. The orders of coupling expansion has the advantage that it facilitates fitting of a PES and also the advantage that it makes computing matrix elements of the PES in a basis each of whose functions is a product of univariate functions easier. In this paper, we focus on the fitting advantage. If terms with more than d coordinates are small or have a small effect on observables computed on the PES, then it is advantageous to use the order of couplings idea when fitting. It is often easier to fit many d dimensional terms than the D dimensional potential. Murrell and co-workers used the orders of coupling idea in inter-atomic distance coordinates [102]. When studying the vibrations of molecules, it is more common to use an orders of coupling representation not to fit a PES but only for computing matrix elements [103]. The orders of coupling representation reduces the dimensionality of the integrals one must commute.

In computational chemistry, an orders of coupling representation is usually created from a PES by fixing coordinates at values. For example, terms depending on only one coordinate can be obtained from the PES but setting all other coordinates to equilibrium values. Similarly, the ij-th two-coordinate term that depends on coordinates x_i and x_j is obtained from the PES by setting all other coordinates $x_k, k \neq i,j$ to the values of the so-called expansion center, which in the case of a molecular PES is usually taken to be the equilibrium molecular geometry. This orders of coupling expansion is called a cut-HDMR (high dimensional mode representation) by Rabitz et al. who developed a systematic theory of such representations [89–92,104]. An alternative is to use RS-HDMR [89,90,92]. Here "RS" stands for "random sampling" although the key idea is the ability to use any desired distribution of samples in space rather than their randomness. In principle, a PES built from RS-HDMR component functions is more accurate than its cut-HDMR counterpart because the RS-HDMR component functions minimize error in all of configuration space. A huge advantage of RS-HDMR is that all the component functions can be determined from a single set of samples of $f(\boldsymbol{x})$, for example, from a single set of ab initio points. There is no need to choose values of coordinates at which to determine cut-HDMR component functions. It the originally proposed RS-HDMR [89,90], the component functions $f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ are determined by multidimensional integrals

$$\int_D f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) f_{j_1 j_2 \ldots j_m}(x_{j_1}, x_{j_2}, \ldots, x_{j_m}) d\boldsymbol{x} = 0 \quad \{i_1, i_2, \ldots, i_d\} \neq \{j_1, j_2, \ldots, j_m\} \tag{2.1.4}$$

Weight functions can also be used in the integrals of Eqs. (2.1.3–4). The $(D-d)$-dimensional integrals of Eq. (2.1.3) could be calculated with Monte Carlo methods but remain costly in high-D. Instead, one can fit the component functions with a ML method [74,75,97,105–111]. If d is small enough, one can perform machine learning in sufficiently low-dimensional spaces to avoid or alleviate issues associated with high dimensionality such as the low density of sampling. If one has to use d-dimensional terms, in principle there is no need to carry lower-dimensional terms that can be lumped into d-dimensional terms, resulting in an approximation

$$f(\boldsymbol{x}) \approx \sum_{\{i_1 i_2 \ldots i_d\} \in \{12 \ldots D\}} f^{ML}_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \tag{2.1.5}$$

The number of terms can be further reduced if some variable combinations are unimportant (see Section 2.4). The approximation (2.1.5) in general does not preserve orthogonality of component functions (which is not required in many applications) but gains in simplicity.

2.2. Orders of coupling representations with HMDR-NN and HDMR-GPR

We first proposed combining the idea of HDMR with neural networks (HDMR-NN) [74,97,105–108] and later with Gaussian Process Regression (HDMR-GPR) [69,75,110,111]. When using the approximation of Eq. (2.1.5), each component function can be fitted to the difference between the value of the target function $f(\boldsymbol{x})$ and the sum of all other component functions:

$$f_{k_1 k_2 \ldots k_d}(x_{k_1}, x_{k_2}, \ldots, x_{k_d}) = f(\boldsymbol{x}) - \sum_{\substack{\{i_1 i_2 \ldots i_d\} \in \{1,2 \ldots D\} \\ \{i_1 i_2 \ldots i_d\} \neq \{k_1 k_2 \ldots k_d\}}} f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \tag{2.2.1}$$

This is done in cycles until convergence. If an orders of coupling expansion explicitly including lower-dimensional terms (Eq. (2.1.2)) is desired, Eq. (2.2.1) can be consecutively applied to increasing orders of coupling:

$$f_{k_1 k_2 \ldots k_d}(x_{k_1}, x_{k_2}, \ldots, x_{k_d}) = T(\boldsymbol{x}) - \sum_{\substack{\{i_1 i_2 \ldots i_d\} \in \{1,2 \ldots D\} \\ \{i_1 i_2 \ldots i_d\} \neq \{k_1 k_2 \ldots k_3\}}} f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \tag{2.2.2}$$

where

$$T(\boldsymbol{x}) = f(\boldsymbol{x}) - \left(f_0 + \sum_{i=1} f_i(x_i) + \sum_{1 \leq i < j \leq D} f_{ij}(x_i, x_j) + \ldots + \sum_{\{i_1 i_2 \ldots i_{d-1}\} \in \{12 \ldots D\}} f_{i_1 i_2 \ldots i_{d-1}}(x_{i_1}, x_{i_2}, \ldots, x_{i_{d-1}}) \right) \tag{2.2.3}$$

$$\begin{aligned} f_0 &= \int f(\boldsymbol{x}) d\boldsymbol{x} \\ f_i(x_i) &= \int f(\boldsymbol{x}) \prod_{i \neq j} dx_j - f_0 \\ f_{ij}(x_i, x_j) &= \int f(\boldsymbol{x}) \prod_{k \neq \{i,j\}} dx_k - f_i(x_i) - f_j(x_j) - f_0 \end{aligned} \tag{2.1.3}$$

The component functions are then mutually orthogonal,

is fitted first. In this case, d-dimensional terms do not include lower-dimensional contributions. The above equations allow doing ML in lower-dimensional feature spaces but necessitate using as many NN or GPR instances as there are component functions. With both NN and GPR, it is also possible to use a single NN or a single GPR realizing the HDMR structure.

With a NN, one can obtain an HDMR structure by selecting NN connectivity. Due to the high CPU cost of nonlinear optimization of NN parameters, this is relatively costly with a conventional NN. See below

(Section 2.7) about the possibility of realizing HDMR with a single NN using an unconventional NN without non-linear parameter optimization. When RS-HDMR-NN was first introduced in Ref. [74], all component functions of a given d were included into a single NN, but non-linear optimization was done in cycles on small subsets of NN parameters at a time to handle the computational cost. In that work, six-dimensional PESs of H_2O_2 and H_2CO molecules were fitted, and it was demonstrated that with a finite size of the training set, only coupling terms up to a maximum d can be determined. Using instead separate NNs for each component function is advantageous because then existing NN libraries can be easily used, and the computational cost is contained as only one low-dimensional NN is trained at a time. The idea of using a sum of NNs each of which depends on a subset of the coordinates is also used in Behler's HDNN approach [2,5,112].

With HDMR-NN, it is also easy to build the approximation of Eqs. (2.1.5) or (2.2.2) in new coordinates $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ (where $\boldsymbol{A}$ is in general rectangular). This can be achieved by introducing a linear neuron layer with d neurons connecting the D-dimensional input layer and the non-linear hidden layer(s) of the component function NNs, giving rise to redundant coordinate HDMR-NN (red-RS-HDMR-NN) [105,106]. $\boldsymbol{A}$ and $\boldsymbol{b}$ are then automatically optimized during training. The name comes from the fact that a set of all $\boldsymbol{y}$ components of all HDMR component functions forms a set of redundant coordinates. This approach has the advantage of breaking the hardwired connection between the number of component functions M, d, and D in the traditional orders of coupling representation, in which $M = C_d^D$, where C_d^D is a binomial coefficient. The combinatorial scaling of M with both d and D is a significant disadvantage of HDMR and is alleviated with red-HDMR-NN. In Ref. [106], by fitting the PESs of the hydrogen peroxide molecule and $OH+H_2$ reaction, it was shown that one can then achieve the desired accuracy either by increasing d or by increasing the number of terms with the same d. In Ref. [107], this approach was applied to fit the 12-dimensional PES of vinyl bromide. When fitting the PES to 20,000 points (randomly distributed in the 12-dimensional space), a similar test set rmse on the order of 166–167 cm^{-1} could be obtained with either 10 5D functions or 15 4D functions or 20 3D functions (for comparison, there would have been 792 5D terms with standard HDMR). In Ref. [111], this approach was also used with HDMR-GPR without automatic optimization of $\boldsymbol{A}$ and $\boldsymbol{b}$ and was shown to improve the accuracy with a given d. In Refs. [105,108], on the example of fitting a PES for N_2O on Cu(111) surface, it was demonstrated that this approach can be used to realize dimensionality reduction.

HDMR-GPR can be realized either as Eq. (2.2.1) or as a single GPR instance with an additive model of the kernel,

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_{\{i_1 i_2 \ldots i_d\} \in \{1,2\ldots D\}} k_{i_1 i_2 \ldots i_d}\left(\boldsymbol{x}_{i_1 i_2 \ldots i_d}, \boldsymbol{x}'_{i_1 i_2 \ldots i_d}\right) \tag{2.2.4}$$

With Eq. (2.2.4), one automatically obtains the representation of Eq. (2.1.5) of the target function. If lower-order terms are explicitly included in Eq. (2.2.4), the representation of Eq. (2.1.2) is obtained. This was demonstrated in Refs. [69,110] on the examples of fitting PESs of H_2CO, UF_6, and of kinetic energy densities in, respectively, 6D, 15D, and 7D. A very high-dimensional example ($D = 2346$) was treated with a 1st order additive kernel in Ref. [70]. An advantage of this approach is the ease of use of existing GPR libraries, where only a custom kernel needs to be defined. A disadvantage is a higher computational cost, especially if hyperparameters need to be optimized separately for different coordinates, because the number of terms in Eq. (2.2.4) can be large. If hyperparameters are fixed or are the same for different coordinates, the sum in Eq. (2.2.4) can be done efficiently. HDMR-GPR approach with separate GPR instances (Eq. (2.2.1-2)) was realized in Refs. [75,111] where it was applied to the PESs of H_2O, H_2CO, and UF_6 as well as to fitting kinetic energy densities and financial market data. This has the advantage of relying on existing GPR and kernel implementations and of containing the computational cost as only one low-dimensional GPR instance is trained at a time. Both approaches, Eq. (2.2.4) and Eq. (2.2.1-2), allow avoiding the use of high-dimensional product-like kernels that can lead to GPR collapse when $D \to \infty$ [70]. Available Python implementation of HDMR-GPR via Eq. (2.2.1-2) [111] also allows linearly transforming coordinates before applying HDMR-GPR. All HDMR-GPR applications in computational chemistry cited above used test sets much larger than the training sets to reliably gauge the global quality of the approximations.

2.3. Facilitation of hyperparameter optimization with HDMR

The quality of Gaussian process regression models stands and falls on the appropriate selection of hyperparameters. While GPR is often presented as a nonparametric approach, hyperparameters such as the length parameters l of Matern type kernels [58].

$$k(\boldsymbol{x},\boldsymbol{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\ \frac{|\boldsymbol{x}-\boldsymbol{x}'|}{l}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\ \frac{|\boldsymbol{x}-\boldsymbol{x}'|}{l}\right) \tag{2.3.1}$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and σ^2 is the prefactor (often set to the variance of the target), *are* nonlinear parameters even though they are relatively few (compared e.g. to the number of nonlinear parameters of an NN). Their incorrect choice leads to overfitting. Hyperparameters are often optimized by maximizing the log likelihood function:

$$\max\left(\frac{1}{2}\ln|\boldsymbol{K}| - \frac{1}{2}\boldsymbol{t}\boldsymbol{K}^{-1}\boldsymbol{t} - \frac{M}{2}\ln(2\pi)\right) \tag{2.3.2}$$

where $\boldsymbol{K}$ is the covariance matrix and $\boldsymbol{t}$ the vector of M target values of the training set entering the defining equations of GPR [66]. This is known as the MLE (maximum likelihood estimator) [113]. Various other methods have been developed, testifying to the importance of the issue of hyperparameter optimization [113–120]. All of them determine hyperparameters from available data. When the data density is very low, there may not be enough information in the data to determine hyperparameters assuring the best accuracy of the model in all space.

Regardless of the number of available datapoints, it is useful to approach the issue of hyperparameter selection as the issue of maximizing the completeness of the basis of the regularized linear regression that GPR is. In Ref. [69], by pointing out that GPR is a regularized linear regression over the basis $b_n(\boldsymbol{x}) = k\left(\boldsymbol{x}, \boldsymbol{x}^{(n)}\right)$ formed by kernel functions (where $\boldsymbol{x}^{(n)}$ are training points, $n = 1, \ldots, M$) [67],

$$f(\boldsymbol{x}) = \sum_{n=1}^{M} b_n(\boldsymbol{x}) c_n \tag{2.3.3}$$

where c_n are coefficients, it was shown that when data density is low, it is advantageous to set the hyperparameters empirically from the perspective of the completeness of the basis rather than to optimize them automatically. In our experience, when using large test sets (much larger than the training set) [59,69,73,75,110,111,121], optimal length parameters (those providing the best test set error) were always larger than those found by automatic optimization. We documented a spectacular failure of MLE when optimizing hyperparameters in a 15-dimensional GPR from up to 10,000 training datapoints in Ref. [69]. When a large test set is available, optimal hyperparameters can be found by a manual scan if they are not many (e.g. when using an isotropic kernel with normalized features). When it is not available, the issue of finding optimal parameters can be severe when the data are sparse.

Terms of a low-order HDMR model can be determined from rather few data [74]. In the example of a 15-dimensional problem of Ref. [69] cited above, as few as 500 points were sufficient to reliably determine the component functions of a 1st order additive model $f(\boldsymbol{x}) = \sum_{i=1}^{D} f_i(x_i)$. A low-order HDMR model can therefore be built from available data and then used to generate any number of samples - ersatz "test points" that can be used to optimize the hyperparameters of a higher-dimensional model. This procedure was explored in Ref. [69] when fitting the PES

of the UF_6 molecule. The hyperparameters optimized based on the ersatz points sampled from the additive model somewhat differed from those that would have been obtained had a large test set of samples of the original function been available, but they were sufficiently good. When using 5000 training and 40,000 "real" test points and an isotropic squared exponential kernel with normalized features, the lowest test set error was 40.1 cm^{-1} and 47.8 cm^{-1} with length parameters $l = 4.0$ and $l = 4.5$, respectively. The lowest test set error on ersatz test points (i.e. those sampled *in 15D* from the 1st order additive model) was 1.6 cm^{-1} and 1.7 cm^{-1} (errors when fitting the ersatz model are expectedly low) with $l = 4.0$ and $l = 4.5$, respectively. That is, a low-order HDMR model whose hyperparameters are easier to determine from sparse data can serve to derive near-optimal hyperparameters for a high-dimensional model.

2.4. HMDR-GPR as an alternative to ARD (automated relevance determination): generating insight with a general method

GPR can be used to evaluate the relative importance of features in a procedure known as automated relevance determination (ARD) [66]. In it, one uses an anisotropic squared exponential kernel,

$$k(\boldsymbol{x},\boldsymbol{x}') = \prod_{i=1}^{D} \exp\left(-\frac{(x_i - x_i')^2}{2l_i^2}\right) \quad (2.4.1)$$

with length parameters l_i optimized, for example, with the MLE [113] method. Variables with low l_i are deemed to be more important than those with high l_i. When the dimensionality is high and data density is low, optimization of hyperparameters of an anisotropic kernel is computationally costly and may be unreliable [69]. HDMR-GPR offers and alternative to ARD approach to evaluating the importance of variables; moreover, it allows evaluating the importance of different combinations of variables, by the magnitude of corresponding component functions (defined, for example, as the variance of $f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$). This can be used to discard some of the coupling terms and thereby palliate a key drawback of HMDR – the combinatorial growth of the number of terms with d and D. The magnitude of component functions can be evaluated with *or without* the costly automated hyperparameter optimization that is required by ARD.

In Ref. [111], the importance of different sets of variables was analyzed based on the magnitude of component functions when fitting kinetic energy densities (KED) of solid materials (Li, Mg, Al) as a function of density-dependent variables

$$\{\tau_{TF}, \tau_{TF}p, \tau_{TF}q, \tau_{TF}p^2, \tau_{TF}pq, \tau_{TF}q^2, \rho V_{eff}\} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} \quad (2.4.2)$$

where $\tau_{TF} = \frac{3(3\pi^2)^{\frac{2}{3}}}{10}\rho^{\frac{5}{3}}(\boldsymbol{r})$ is the Thomas-Fermi KED, and p an q are scaled [122] gradient and Laplacian of the electron density $\rho(\boldsymbol{r})$, $p = \frac{|\nabla\rho|^2}{4(3\pi^2)^{2/3}\rho^{8/3}}$.

and $q = \frac{\Delta\rho}{4(3\pi^2)^{2/3}\rho^{5/3}}$, respectively, and V_{eff} is the density-dependent Kohn-Sham effective potential. The first six descriptors are terms of the 4th order gradient expansion [52]. These descriptors were previously shown to be effective for the regression of KED [18,51]. The length parameters of isotropic (but different for different component functions) squared exponential kernels were optimized when using different numbers of training data points. The authors found that depending on the training dataset size (sampling density) different sets of coordinates were meaningful. For some component functions, the optimal length parameter became very large and simultaneously the magnitude of the corresponding component function tended to 0, while for other component functions the optimal length parameter was on the order of 1 (inputs were scaled on a unit cube) and the component function had a significant amplitude. For example, in Table 1 we show the resulting length parameter values and component function variances of third-order terms obtained in Ref. [111] when learning the KED of Li, Al, and Mg simultaneously from a training set of 5000 points. Only 3 terms out of 35 (those dependent on (x_1, x_3, x_7), (x_2, x_3, x_4), and (x_2, x_3, x_7)) were meaningful while the rest could be omitted. See Ref. [111] for the results for other d.

This approach not only allows pruning coupling terms, but it also provides information on the relative importance of combinations of features that could be used elsewhere e.g. for deriving analytic models. This is insight obtained with a method that remains general.

Table 1
The length parameter l and the variance *var* of the 3rd order HDMR-GPR component functions $f_{i,j,k}(x_i, x_j, x_k)$ when fitting kinetic energy densities of Al, Mg, and Si to 5000 data. The maximum allowed value of l was 10^5. Target and features were scaled to [0,1] before fitting.

features	$\mathrm{var}\{f_{i_1,i_2,i_3}\}$	l	features	$\mathrm{var}\{f_{i_1,i_2,i_3}\}$	l
x_1, x_2, x_3	5.25×10^{-11}	10^5	x_2, x_3, x_7	0.0109	4.000
x_1, x_2, x_4	3.45×10^{-10}	10^5	x_2, x_4, x_5	2.04×10^{-10}	10^5
x_1, x_2, x_5	1.65×10^{-10}	10^5	x_2, x_4, x_6	2.49×10^{-10}	10^5
x_1, x_2, x_6	1.57×10^{-10}	10^5	x_2, x_4, x_7	2.91×10^{-10}	10^5
x_1, x_2, x_7	1.91×10^{-10}	10^5	x_2, x_5, x_6	2.26×10^{-10}	10^5
x_1, x_3, x_4	4.20×10^{-10}	10^5	x_2, x_5, x_7	1.82×10^{-10}	10^5
x_1, x_3, x_5	9.00×10^{-11}	10^5	x_2, x_6, x_7	1.54×10^{-10}	10^5
x_1, x_3, x_6	9.74×10^{-11}	10^5	x_3, x_4, x_5	2.94×10^{-10}	10^5
x_1, x_3, x_7	0.0151	0.227	x_3, x_4, x_6	3.37×10^{-10}	10^5
x_1, x_4, x_5	6.64×10^{-10}	10^5	x_3, x_4, x_7	3.68×10^{-10}	10^5
x_1, x_4, x_6	5.56×10^{-10}	10^5	x_3, x_5, x_6	1.73×10^{-10}	10^5
x_1, x_4, x_7	5.26×10^{-10}	10^5	x_3, x_5, x_7	9.99×10^{-11}	10^5
$x_1, x_5, x6$	1.89×10^{-10}	10^5	x_3, x_6, x_7	8.83×10^{-11}	10^5
x_1, x_5, x_7	1.52×10^{-10}	10^5	x_4, x_5, x_6	1.99E-10	10^5
x_1, x_6, x_7	4.93×10^{-11}	10^5	x_4, x_5, x_7	2.31×10^{-10}	10^5
x_2, x_3, x_4	0.00652	11.1	x_4, x_6, x_7	2.71×10^{-10}	10^5
x_2, x_3, x_5	4.47×10^{-10}	10^5	x_5, x_6, x_7	2.37×10^{-10}	10^5
x_2, x_3, x_6	3.01×10^{-10}	10^5			

2.5. Imputation of missing data

An orders of coupling representation can be used to impute missing data. Data imputation is an important issue, as in many applications some of the data may be missing (e.g. clinical data that are often aggregated from different institutions not all recording the same parameters, or sensorial data where some of the sensors may not be working some of the time). If data are abundant, one may be tempted to only retain complete data rows. This strategy is perilous when D is high. For example, if one collects 10,000 100-dimensional instances for a total of 1000,000 entries of $\boldsymbol{X}$ (matrix of all $\boldsymbol{x}$ instances), and assuming that 1% of elements X_{ij} (where the rows indexed by i are points and columns indexed by j are dimensions, $j = 1, \ldots, D$) are missing with the missing values distributed randomly across the matrix, retaining only complete rows would result in discarding almost all of the data! Imputing data values in principle allows palliating this problem, but approaches such as imputing missing values based on data distribution [123] are of limited accuracy.

An HDMR model allows imputing missing values rather accurately in certain situations. For example, to the extent that an additive (separable) model

$$f(\boldsymbol{x}) = \sum_{i}^{D} f_i(x_i) \quad (2.5.1)$$

is accurate, and when one $x_i^{(m)}$ out of D is missing, imputation is accurate [111] as long as there are enough complete data rows to define the additive model. As highlighted above, much fewer data points are needed to reliably construct the component functions of the first order model [69]. Once they are constructed, the missing values $x_i^{(m)}$ in the m-th datapoint are reconstructed as

$$x_i^{(m)} = f_i^{-1}\left(x_i^{(m)}\right) \tag{2.5.2}$$

where

$$f_i\left(x_i^{(m)}\right) = f\left(x_i^{(m)}\right) - \sum_{\substack{j=1,\\ j\neq i}}^{D} f_j\left(x_j^{(m)}\right) \tag{2.5.3}$$

is the reconstructed value of the component function for the i-th variable at the m-th datapoint (i.e. in Eq. (2.5.2-3), $f_i^{-1}(x_i^{(m)})$ is the known value of f_i^{-1} for $\boldsymbol{x}^{(m)}$ whose i-th component is unknown). The imputation naturally becomes inaccurate when f_i^{-1} is singular. It becomes ambiguous when f_i^{-1} is multivalued. In this case Eq. (2.5.2) still allows narrowing down the choices of possible values, and heuristics can be used to select the most likely of them. In Ref. [111], imputation with this method was demonstrated for various multivariate polynomial functions $f(\boldsymbol{x})$.

2.6. HMDR-GPR as a way to get optimal neuron activation functions

In Ref. [124], a method was presented for constructing a neural network with neuron activation functions that are optimal for a given problem and are individual to each neuron. The activation functions are built from an additive Gaussian process [109]. It was recognized that a single hidden layer NN with a linear output neuron, which has the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} c_i \sigma_i(\boldsymbol{w}_i\boldsymbol{x} + b_i) = \sum_{i=1}^{N} f_i(y_i) \tag{2.6.1}$$

is an additive model (1st order HDMR-GPR) in redundant coordinates $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ (output neuron and its weight and bias are omitted without loss of generality as they can be moved to the left hand side). Once the components of $\boldsymbol{y}$ are defined, one can use a first-order HDMR-GPR to construct $f_i(y_i)$. This is conveniently done in a single step by using a first order additive kernel, $k(\boldsymbol{y},\boldsymbol{y}') = \sum_{i=1}^{N} k(y_i,y_i')$. The definition of the elements of $\boldsymbol{y}$, i.e. weights and biases, is done in traditional NNs via non-linear fitting. It is to facilitate the non-linear fitting with approaches such as backpropagation [125,126] that the same simple analytic form of σ_i (e.g. sigmoid) is typically used for all neurons. In Ref. [124], the authors have shown that weights can be defined with a rule-based method. Specifically, any desired number of redundant coordinates y_i can be defined by using a pseudorandom Sobol sequence [127]: $y_i = \boldsymbol{x}^T\boldsymbol{s}_i$ where s_i is the i-th element (vector) of a D-dimensional Sobol sequence. Note that there is no need to consider the biases as they are automatically subsumed into f_i when using HDMR-GPR. To test the method, the authors fitted the PESs of H_2O and H_2CO in the spectroscopically relevant region. They showed that the use of weights fixed by rules does not prevent obtaining a fit quality on par with traditional NN and GPR approaches [75,121] (the rmse on a large test set is below 0.5 cm^{-1} when fitting 1000 points for H_2O and about 20 cm^{-1} when fitting 1000 points for H_2CO). Importantly, it allows avoiding overfitting when the number of neurons increases beyond the optimal. This is ultimately due to the avoidance of non-linear parameter optimization. It is obviating the need for non-linear parameter optimization that makes it easy to use different activation functions for different neurons. As the rule-based weights are in general not optimal, one may need to use more neurons than in a fully optimized traditional NN, but the absence of weights optimization means that this is not costly; more important is the fact that one avoids overfitting while achieving a test set error which is as good as with a fully optimized traditional NN. One essentially obtains a method with an expressive power of an NN but which is as robust as a linear regression. Different rules to define weights can be explored. Extensions of this approach towards NNs with multivariate neuron activation functions, which have been proposed in various applications [128–130], are possible (i.e. multivariate neurons constructed from higher orders of HDMR-GPR) and are awaiting to be explored.

2.7. HDMR-NN via HDMR-GPR

The method discussed in the previous section can be naturally extended to build an NN with an architecture realizing an orders of coupling representation. HDMR-NN can be realized either as a single NN with connectivity realizing HDMR or as separate NN instances. A single NN realizing an orders of coupling representation is attractive due to the simplicity of the concept, but is computationally costly as non-linear parameters of all component functions need to be optimized simultaneously. If single hidden layer NNs are used for component functions, then using NNs with weights fixed by rules for component functions dispenses with this issue. Coupling terms can then be easily defined by using d-dimensional pseudorandom sequences with $d < D$: redundant coordinates $\boldsymbol{y}^{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) = \boldsymbol{W}\boldsymbol{x}$ can be defined as $y_i^{i_1 i_2 \ldots i_d} = \boldsymbol{x}^T\boldsymbol{s}_i$ where $\boldsymbol{s}_i$ is a vector whose d elements are taken from a d-dimensional Sobol sequence, and the other $(D-d)$ elements are zero [131]. The choice of the d non-zero elements determines the selection of a particular coupling term dependent on $(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$, and the number of the elements of the d-dimensional Sobol sequences $N_{i_1 i_2 \ldots i_d}$ defines the number of neurons of each component function NN. All y_i dependent on different selections of $(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ are concatenated into a single vector of redundant coordinates $\boldsymbol{y}$, and all terms of the final approximation

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{\{i_1 i_2 \ldots i_d\} \in \{1,2 \ldots D\}} f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \\ &= \sum_{i_1 i_2 \ldots i_d} \sum_{j=1}^{N_{i_1 i_2 \ldots i_d}} \sigma_{j,i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \end{aligned} \tag{2.7.1}$$

are found from a single first-order HDMR-GPR calculation in one step, by using a first order additive kernel, $k(\boldsymbol{y},\boldsymbol{y}') = \sum_{i=1}^{N} k(y_i,y_i')$. Eq. (2.7.1) is a HMDR-NN [74,97,105–108] where $f_{i_1 i_2 \ldots i_d}(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ are single hidden layer NNs with customized neuron activation functions $\sigma_{j,i_1 i_2 \ldots i_d}$.

In Ref. [131], this approach was used to fit the PES of the formaldehyde molecule and was shown to be competitive with other approaches to build RS-HMDR type models [75,110]. Table 2 compares the rmse of the PES of H_2CO on a large test set achieved with orders of coupling representations of different orders built using Eq. (2.7.1) in Ref. [131] and the RS-HDMR-GPR approach discussed in Section 2.2. The performance is similar.

3. Conclusions

It is challenging to work with data in a multi-dimensional space. One of the most important goals of scientists working with data in many dimensions is to build a function that can be computed at any point in the full dimensional space that reproduces or nearly reproduces values at a set of known data points. The problem is ubiquitous. It is now

Table 2
Comparison of test set rmse values achieved with different orders d of HDMR-NN (GPR) in Ref. [131] using Eq. (2.7.1) and reference HDMR-GPR calculations from Ref. [75], when fitting the PES of H_2CO to the same data. 1000 training points and 10,000 test points were used.

d	No. of coupling terms	Test rmse, cm^{-1}, with Eq. (2.7.1) (Ref. [131])	Test rmse, cm^{-1}, with RS-HDMR-GPR (Ref. [75])
1	6	1302.7	1315.6
2	15	385.5	410.0
3	20	20.4	29.1
4	15	14.6	16.1
5	6	16.5	14.4
6	1	19.6	23.8

common to use machine-learning methods to solve the problem. In chemistry, examples of such applications include the fitting of potential energy surfaces, functionals, refitting of errors of ab initio methods (so-called Δ-learning), fitting of properties of molecules and materials as function of descriptors, etc. ML methods impose no (or few) constraints on the fitting function. This has the advantage of conveying generality but the disadvantage that because no prior knowledge is used more data points are required. Given enough points and enough computer power, the ML task is in principle straightforward, and many possible algorithms can be used. The problem becomes interesting and challenging when the multi-dimensional volume of the space in which a representation of the function is desired is so large that the density of data points is small. When the dimensionality of the space is small it is always possible to simply use more data points. When the dimensionality of the space is large, it is not possible to mitigate the problem by using enough points. In large dimensions, sampling is intrinsically sparse.

In this paper, we reviewed various options for using low-dimensional structure in conjunction with ML methods to facilitate representing functions in many dimensions. They are all based on what Rabitz et al. have called a high-dimensional model representation (HDMR). A D-dimensional function can be represented with a single D-dimensional function or as a sum of d-dimensional functions, where $d < D$. The disadvantage of using d-dimensional functions is that there is more than one of them; the advantage is that it is easier to determine lower-dimensional functions. In fact, the number of data points is linked to the determinability of the parameters of d-dimensional fitting functions: the smaller the number of data points the harder it is to determine functions for which d is larger. ML methods can be used to determine the lower dimensional functions in an HDMR. We have discussed using both NN and GPR methods for this purpose, respectively HDMR-NN and HDMR-GPR methods.

Because low-dimensional HDMR terms are easier to determine from few data, they can be used to effectively optimize hyperparameters, including hyperparameters usable for higher-dimensional terms. The HDMR structure allows evaluating the importance of variables and of specific sets of variables based on the magnitude of HDMR component functions, in an alternative approach to ARD. Contrary to ARD, this can be done without automatic hyperparameter optimization, which may be costly and unreliable in high-D. This effectively generates elements of insight with a general method. The number of HDMR terms can be reduced by discarding low-magnitude component functions. HDMR ideas (specifically HDMR-GPR) can also be used to create a NN with customized neuron activation functions that are different for different neurons and in a way that avoids nonlinear parameter optimization and with it the dangers of overfitting when the number of neurons is increased. They can also be used to create a NN that has an architecture realizing HDMR. The HDMR structure also allows imputing missing data; the imputation can be very accurate when a low-order HDMR model is accurate.

In general, the idea of exploiting the fact that it is easier to fit lower-dimensional functions to build a higher-dimensional function from sparse data seems promising for further developments and applications in and beyond computational chemistry, as HDMR-based ML approaches exposed here are general.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by JST-Mirai Program Grant Number JPMJMI22H1, Japan and NSERC of Canada. The funders had no involvement in the design of this study.

References

[1] S. Manzhos, T. Carrington, Neural network potential energy surfaces for small molecules and reactions, Chem. Rev. 121 (2021) 10187–10217, https://doi.org/10.1021/acs.chemrev.0c00665.
[2] J. Behler, Constructing high-dimensional neural network potentials: a tutorial review, Int. J. Quantum Chem. 115 (2015) 1032–1050, https://doi.org/10.1002/qua.24890.
[3] H. Kulik, T. Hammerschmidt, J. Schmidt, S. Botti, M.A.L. Marques, M. Boley, M. Scheffler, M. Todorović, P. Rinke, C. Oses, A. Smolyanyuk, S. Curtarolo, A. Tkatchenko, A. Bartok, S. Manzhos, M. Ihara, T. Carrington, J. Behler, O. Isayev, M. Veit, A. Grisafi, J. Nigam, M. Ceriotti, K.T. Schütt, J. Westermayr, M. Gastegger, R. Maurer, B. Kalita, K. Burke, R. Nagai, R. Akashi, O. Sugino, J. Hermann, F. Noé, S. Pilati, C. Draxl, M. Kuban, S. Rigamonti, M. Scheidgen, M. Esters, D. Hicks, C. Toher, P. Balachandran, I. Tamblyn, S. Whitelam, C. Bellinger, L.M. Ghiringhelli, Roadmap on machine learning in electronic structure, Electron. Struct. (2022), https://doi.org/10.1088/2516-1075/ac572f.
[4] S. Manzhos, R. Dawes, T. Carrington, Neural network-based approaches for building high dimensional and quantum dynamics-friendly potential energy surfaces, Int. J. Quantum Chem. 115 (2015) 1012–1020, https://doi.org/10.1002/qua.24795.
[5] J. Behler, Perspective: machine learning potentials for atomistic simulations, J. Chem. Phys. 145 (2016), 170901, https://doi.org/10.1063/1.4966192.
[6] J. Behler, First principles neural network potentials for reactive simulations of large molecular and condensed systems, Angew. Chem. Int. Ed. 56 (2017) 12828–12840, https://doi.org/10.1002/anie.201703114.
[7] B.J. Braams, J.M. Bowman, Permutationally invariant potential energy surfaces in high dimensionality, Int. Rev. Phys. Chem. 28 (2009) 577–606, https://doi.org/10.1080/01442350903234923.
[8] H. Ghorbanfekr, J. Behler, F.M. Peeters, Insights into water permeation through hBN nanocapillaries by Ab initio machine learning molecular dynamics simulations, J. Phys. Chem. Lett. 11 (2020) 7363–7370, https://doi.org/10.1021/acs.jpclett.0c01739.
[9] E. Bosoni, D. Campi, D. Donadio, G.C. Sosso, J. Behler, M. Bernasconi, Atomistic simulations of thermal conductivity in GeTe nanowires, J. Phys. D: Appl. Phys. 53 (2019), 054001, https://doi.org/10.1088/1361-6463/ab5478.
[10] S. Gabardi, E. Baldi, E. Bosoni, D. Campi, S. Caravati, G.C. Sosso, J. Behler, M. Bernasconi, Atomistic simulations of the crystallization and aging of GeTe nanowires, J. Phys. Chem. C. 121 (2017) 23827–23838, https://doi.org/10.1021/acs.jpcc.7b09862.
[11] M.L. Paleico, J. Behler, Global optimization of copper clusters at the ZnO($10\bar{1}0$) surface using a DFT-based neural network potential and genetic algorithms, J. Chem. Phys. 153 (2020), 054704, https://doi.org/10.1063/5.0014876.
[12] J. Weinreich, A. Römer, M.L. Paleico, J. Behler, Properties of α-Brass Nanoparticles. 1. Neural Network Potential Energy Surface, J. Phys. Chem. C. 124 (2020) 12682–12695, https://doi.org/10.1021/acs.jpcc.0c00559.
[13] N. Gerrits, K. Shakouri, J. Behler, G.-J. Kroes, Accurate probabilities for highly activated reaction of polyatomic molecules on surfaces using a high-dimensional neural network potential: CHD3 + Cu(111), J. Phys. Chem. Lett. 10 (2019) 1763–1768, https://doi.org/10.1021/acs.jpclett.9b00560.
[14] S. Kondati Natarajan, J. Behler, Neural network molecular dynamics simulations of solid–liquid interfaces: water at low-index copper surfaces, Phys. Chem. Chem. Phys. 18 (2016) 28704–28725, https://doi.org/10.1039/C6CP05711J.
[15] J. Behler, Representing potential energy surfaces by high-dimensional neural network potentials, J. Phys.: Condens. Matter 26 (2014), 183001, https://doi.org/10.1088/0953-8984/26/18/183001.
[16] B. Kolb, P. Marshall, B. Zhao, B. Jiang, H. Guo, Representing global reactive potential energy surfaces using gaussian processes, J. Phys. Chem. A 121 (2017) 2552–2557, https://doi.org/10.1021/acs.jpca.7b01182.
[17] K. Yao, J. Parkhill, Kinetic energy of hydrocarbons as a function of electron density and convolutional neural networks, J. Chem. Theory Comput. 12 (2016) 1139–1147, https://doi.org/10.1021/acs.jctc.5b01011.
[18] P. Golub, S. Manzhos, 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. 21 (2018) 378–395, https://doi.org/10.1039/C8CP06433D.
[19] J. Seino, R. Kageyama, M. Fujinami, Y. Ikabata, H. Nakai, Semi-local machine-learned kinetic energy density functional demonstrating smooth potential energy curves, Chem. Phys. Lett. 734 (2019), 136732, https://doi.org/10.1016/j.cplett.2019.136732.
[20] M. Fujinami, R. Kageyama, J. Seino, Y. Ikabata, H. Nakai, Orbital-free density functional theory calculation applying semi-local machine-learned kinetic energy density functional and kinetic potential, Chem. Phys. Lett. 748 (2020), 137358, https://doi.org/10.1016/j.cplett.2020.137358.
[21] S. Manzhos, Machine learning for the solution of the Schrödinger equation, Mach. Learn.: Sci. Technol. 1 (2020), 013002, https://doi.org/10.1088/2632-2153/ab7d30.
[22] C. Duan, F. Liu, A. Nandy, H.J. Kulik, Putting density functional theory to the test in machine-learning-accelerated materials discovery, J. Phys. Chem. Lett. 12 (2021) 4628–4637, https://doi.org/10.1021/acs.jpclett.1c00631.
[23] M. Bogojeski, L. Vogt-Maranto, M.E. Tuckerman, K.-R. Müller, K. Burke, Quantum chemical accuracy from density functional approximations via machine learning, Nat. Commun. 11 (2020) 5223, https://doi.org/10.1038/s41467-020-19093-1.
[24] R. Pederson, B. Kalita, K. Burke, Machine learning and density functional theory, Nat. Rev. Phys. 4 (2022) 357–358, https://doi.org/10.1038/s42254-022-00470-2.

[25] F. Brockherde, L. Vogt, L. Li, M.E. Tuckerman, K. Burke, K.-R. Müller, Bypassing the Kohn-Sham equations with machine learning, Nat. Commun. 8 (2017) 872, https://doi.org/10.1038/s41467-017-00839-3.

[26] F. Legrain, S. Manzhos, Highly accurate local pseudopotentials of Li, Na, and Mg for orbital free density functional theory, Chem. Phys. Lett. 622 (2015) 99–103, https://doi.org/10.1016/j.cplett.2015.01.016.

[27] J. Lüder, S. Manzhos, Nonparametric local pseudopotentials with machine learning: a tin pseudopotential built using gaussian process regression, J. Phys. Chem. A 124 (2020) 11111–11124, https://doi.org/10.1021/acs.jpca.0c05723.

[28] F. Li, X. Peng, Z. Wang, Y. Zhou, Y. Wu, M. Jiang, M. Xu, Machine learning (ML)-assisted design and fabrication for solar cells, Energy Environ. Mater. 2 (2019) 280–291, https://doi.org/10.1002/eem2.12049.

[29] S. Manzhos, M. Ihara, 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, Physchem 2 (2022) 72–95, https://doi.org/10.3390/physchem2020006.

[30] Q. Tong, P. Gao, H. Liu, Y. Xie, J. Lv, Y. Wang, J. Zhao, Combining machine learning potential and structure prediction for accelerated materials design and discovery, J. Phys. Chem. Lett. 11 (2020) 8710–8720, https://doi.org/10.1021/acs.jpclett.0c02357.

[31] W.P. Walters, R. Barzilay, Applications of deep learning in molecule generation and molecular property prediction, Acc. Chem. Res. 54 (2021) 263–270, https://doi.org/10.1021/acs.accounts.0c00699.

[32] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, C. Kim, Machine learning in materials informatics: recent applications and prospects, Npj Comput. Mater. 3 (2017) 1–13, https://doi.org/10.1038/s41524-017-0056-5.

[33] A.Y.-T. Wang, R.J. Murdock, S.K. Kauwe, A.O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson, T.D. Sparks, Machine learning for materials scientists: an introductory guide toward best practices, Chem. Mater. 32 (2020) 4954–4965, https://doi.org/10.1021/acs.chemmater.0c01907.

[34] K.T. Butler, D.W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science, Nature 559 (2018) 547–555, https://doi.org/10.1038/s41586-018-0337-2.

[35] S.M. Moosavi, K.M. Jablonka, B. Smit, The role of machine learning in the understanding and design of materials, J. Am. Chem. Soc. 142 (2020) 20273–20287, https://doi.org/10.1021/jacs.0c09105.

[36] N. Meftahi, M. Klymenko, A.J. Christofferson, U. Bach, D.A. Winkler, S.P. Russo, Machine learning property prediction for organic photovoltaic devices, Npj Comput. Mater. 6 (2020) 1–8, https://doi.org/10.1038/s41524-020-00429-w.

[37] A. Mahmood, J.-L. Wang, Machine learning for high performance organic solar cells: current scenario and future prospects, Energy Environ. Sci. 14 (2021) 90–105, https://doi.org/10.1039/D0EE02838J.

[38] C.-I. Wang, I. Joanito, C.-F. Lan, C.-P. Hsu, Artificial neural networks for predicting charge transfer coupling, J. Chem. Phys. 153 (2020), 214113, https://doi.org/10.1063/5.0023697.

[39] X. Rodríguez-Martínez, E. Pascual-San-José, M. Campoy-Quiles, Accelerating organic solar cell material's discovery: high-throughput screening and big data, Energy Environ. Sci. 14 (2021) 3301–3322, https://doi.org/10.1039/D1EE00559F.

[40] M. Srivastava, J.M. Howard, T. Gong, M. Rebello Sousa Dias, M.S. Leite, Machine learning roadmap for perovskite photovoltaics, J. Phys. Chem. Lett. 12 (2021) 7866–7877, https://doi.org/10.1021/acs.jpclett.1c01961.

[41] I.A. Moses, R.P. Joshi, B. Ozdemir, N. Kumar, J. Eickholt, V. Barone, Machine learning screening of metal-ion battery electrode materials, ACS Appl. Mater. Interfaces 13 (2021) 53355–53362, https://doi.org/10.1021/acsami.1c04627.

[42] A. Chen, X. Zhang, L. Chen, S. Yao, Z. Zhou, A machine learning model on simple features for CO_2 reduction electrocatalysts, J. Phys. Chem. C. 124 (2020) 22471–22478, https://doi.org/10.1021/acs.jpcc.0c05964.

[43] P. Schlexer Lamoureux, K.T. Winther, J.A. Garrido Torres, V. Streibel, M. Zhao, M. Bajdich, F. Abild-Pedersen, T. Bligaard, Machine learning for computational heterogeneous catalysis, ChemCatChem 11 (2019) 3581–3601, https://doi.org/10.1002/cctc.201900595.

[44] S. Back, J. Yoon, N. Tian, W. Zhong, K. Tran, Z.W. Ulissi, Convolutional neural network of atomic surface structures to predict binding energies for high-throughput screening of catalysts, J. Phys. Chem. Lett. 10 (2019) 4401–4408, https://doi.org/10.1021/acs.jpclett.9b01428.

[45] T. Toyao, Z. Maeno, S. Takakusagi, T. Kamachi, I. Takigawa, K. Shimizu, Machine learning for catalysis informatics: recent applications and prospects, ACS Catal. 10 (2020) 2260–2297, https://doi.org/10.1021/acscatal.9b04186.

[46] D. Wu, J. Zhang, M.-J. Cheng, Q. Lu, H. Zhang, Machine learning investigation of supplementary adsorbate influence on copper for enhanced electrochemical CO_2, Reduct. Perform., J. Phys. Chem. C. 125 (2021) 15363–15372, https://doi.org/10.1021/acs.jpcc.1c05004.

[47] Y. Zhang, C. Hu, B. Jiang, Embedded atom neural network potentials: efficient and accurate machine learning with a physically inspired representation, J. Phys. Chem. Lett. 10 (2019) 4962–4967, https://doi.org/10.1021/acs.jpclett.9b02037.

[48] S.A. Ghasemi, A. Hofstetter, S. Saha, S. Goedecker, Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network, Phys. Rev. B 92 (2015), 045131, https://doi.org/10.1103/PhysRevB.92.045131.

[49] O.T. Unke, M. Meuwly, PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges, J. Chem. Theory Comput. 15 (2019) 3678–3693, https://doi.org/10.1021/acs.jctc.9b00181.

[50] N. Lubbers, J.S. Smith, K. Barros, Hierarchical modeling of molecular energies using a deep neural network, J. Chem. Phys. 148 (2018), 241715, https://doi.org/10.1063/1.5011181.

[51] S. Manzhos, P. Golub, Data-driven kinetic energy density fitting for orbital-free DFT: Linear vs Gaussian process regression, J. Chem. Phys. 153 (2020), 074104, https://doi.org/10.1063/5.0015042.

[52] C.H. Hodges, Quantum Corrections to the Thomas–Fermi Approximation—The Kirzhnits Method, Can. J. Phys. 51 (1973) 1428–1437, https://doi.org/10.1139/p73-189.

[53] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (1965) A1133–A1138, https://doi.org/10.1103/PhysRev.140.A1133.

[54] D.L. Donoho, High-dimensional data analysis: The curses and blessings of dimensionality, in: AMS Conference on Math Challenges of the 21st Century, AMS, 2000.

[55] J.C. Light, T. Carrington Jr., Discrete-Variable Representations and their Utilization, in: Advances in Chemical Physics, John Wiley & Sons, Ltd, 2000, pp. 263–310, https://doi.org/10.1002/9780470141731.ch4.

[56] J.Ignacio Mulero-Martínez, Functions bandlimited in frequency are free of the curse of dimensionality, Neurocomputing 70 (2007) 1439–1452, https://doi.org/10.1016/j.neucom.2006.05.010.

[57] Y. Liao, S.-C. Fang, H.L.W. Nuttle, Relaxed conditions for radial-basis function networks to be universal approximators, Neural Netw. 16 (2003) 1019–1028, https://doi.org/10.1016/S0893-6080(02)00227-7.

[58] M.G. Genton, Classes of kernels for machine learning: a statistics perspective, J. Mach. Learn. Res. 2 (2001) 299–312.

[59] S. Manzhos, M. Ihara, The loss of the property of locality of the kernel in high-dimensional Gaussian process regression on the example of the fitting of molecular potential energy surfaces, J. Chem. Phys. 158 (2023), 044111, https://doi.org/10.1063/5.0136156.

[60] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2001) 181–201, https://doi.org/10.1109/72.914517.

[61] V.L. Deringer, A.P. Bartók, N. Bernstein, D.M. Wilkins, M. Ceriotti, G. Csányi, Gaussian process regression for materials and molecules, Chem. Rev. 121 (2021) 10073–10141, https://doi.org/10.1021/acs.chemrev.1c00022.

[62] L. Li, J.C. Snyder, I.M. Pelaschier, J. Huang, U.-N. Niranjan, P. Duncan, M. Rupp, K.-R. Müller, K. Burke, Understanding machine-learned density functionals, Int. J. Quantum Chem. 116 (2016) 819–833, https://doi.org/10.1002/qua.25040.

[63] B. Kalita, L. Li, R.J. McCarty, K. Burke, Learning to approximate density functionals, Acc. Chem. Res. 54 (2021) 818–826, https://doi.org/10.1021/acs.accounts.0c00742.

[64] A. Christianen, T. Karman, R.A. Vargas-Hernández, G.C. Groenenboom, R. V. Krems, Six-dimensional potential energy surface for NaK–NaK collisions: Gaussian process representation with correct asymptotic form, J. Chem. Phys. 150 (2019), 064106, https://doi.org/10.1063/1.5082740.

[65] G. Montavon, G.B. Orr, K.-R. Mueller, Neural Networks: Tricks of the Trade. Berlin Heidelberg, 2nd ed..,, Springer,, 2012, https://doi.org/10.1007/978-3-642-35289-8.

[66] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge MA, USA, 2006. http://www.gaussianprocess.org/gpml/ (accessed June 19, 2021).

[67] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, Singapore, 2006.

[68] S. Manzhos, M. Ihara, Rectangularization of Gaussian process regression for optimization of hyperparameters, ArXiv:2112.02467 [Cs, Math]. (2021). ⟨http://arxiv.org/abs/2112.02467⟩ (accessed February 8, 2022).

[69] S. Manzhos, M. Ihara, Optimization of hyperparameters of Gaussian process regression with the help of a low-order high-dimensional model representation: application to a potential energy surface, J. Math. Chem. 61 (2023) 7–20, https://doi.org/10.1007/s10910-022-01407-x.

[70] S. Manzhos, S. Tsuda, M. Ihara, Machine learning in computational chemistry: interplay between (non)linearity, basis sets, and dimensionality, Phys. Chem. Chem. Phys. 25 (2023) 1546–1555, https://doi.org/10.1039/D2CP04155C.

[71] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Netw. 3 (1990) 551–560, https://doi.org/10.1016/0893-6080(90)90005-6.

[72] F. Scarselli, A.Chung Tsoi, Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results, Neural Netw. 11 (1998) 15–37, https://doi.org/10.1016/S0893-6080(97)00097-X.

[73] N. Yang, S. Hill, S. Manzhos, T. Carrington, A local Gaussian Processes method for fitting potential surfaces that obviates the need to invert large matrices, J. Mol. Spectrosc. 393 (2023), 111774, https://doi.org/10.1016/j.jms.2023.111774.

[74] S. Manzhos, T. Carrington, A random-sampling high dimensional model representation neural network for building potential energy surfaces, J. Chem. Phys. 125 (2006), 084109, https://doi.org/10.1063/1.2336223.

[75] M.A. Boussaidi, O. Ren, D. Voytsekhovsky, S. Manzhos, 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 124 (2020) 7598–7607, https://doi.org/10.1021/acs.jpca.0c05935.

[76] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science (2006), https://doi.org/10.1126/science.1127647.

[77] S. Manzhos, K. Yamashita, A model for the dissociative adsorption of N2O on Cu (100) using a continuous potential energy surface, Surf. Sci. 604 (2010) 555–561, https://doi.org/10.1016/j.susc.2009.12.025.

[78] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319, https://doi.org/10.1162/089976698300017467.

[79] Modern Multidimensional Scaling, Springer, New York, NY, 2005. https://doi.org/10.1007/0–387-28981-X.

[80] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323, https://doi.org/10.1126/science.290.5500.2319.

[81] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326, https://doi.org/10.1126/science.290.5500.2323.

[82] F. Hausdorff, Dimension und äußeres Maß, Math. Ann. 79 (1918) 157–179, https://doi.org/10.1007/BF01457179.

[83] S. Kak, Information theory and dimensionality of space, Sci. Rep. 10 (2020) 20733, https://doi.org/10.1038/s41598-020-77855-9.

[84] S. Carter, S.J. Culik, J.M. Bowman, Vibrational self-consistent field method for many-mode systems: a new approach and application to the vibrations of CO adsorbed on Cu(100,), J. Chem. Phys. 107 (1997) 10458–10469, https://doi.org/10.1063/1.474210.

[85] K. Raghavachari, A. Saha, Accurate composite and fragment-based quantum chemical models for large molecules, Chem. Rev. 115 (2015) 5643–5677, https://doi.org/10.1021/cr500606e.

[86] S. Carter, J.M. Bowman, N.C. Handy, Extensions and tests of "multimode": a code to obtain accurate vibration/rotation energies of many-mode molecules, Theor. Chem. Acc. 100 (1998) 191–198, https://doi.org/10.1007/s002140050379.

[87] J.M. Bowman, T. Carrington, H.-D. Meyer, Variational quantum approaches for computing vibrational energies of polyatomic molecules, Mol. Phys. 106 (2008) 2145–2182, https://doi.org/10.1080/00268970802258609.

[88] M.H. Beck, A. Jäckle, G.A. Worth, H.-D. Meyer, The multiconfiguration time-dependent Hartree (MCTDH) method: a highly efficient algorithm for propagating wavepackets, Phys. Rep. 324 (2000) 1–105, https://doi.org/10.1016/S0370-1573(99)00047-2.

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

[90] H. Rabitz, Ö.F. Aliş, General foundations of high-dimensional model representations, J. Math. Chem. 25 (1999) 197–233, https://doi.org/10.1023/A:1019188517934.

[91] Ö.F. Alış, H. Rabitz, Efficient implementation of high dimensional model representations, J. Math. Chem. 29 (2001) 127–142, https://doi.org/10.1023/A:1010979129659.

[92] G. Li, S.-W. Wang, H. Rabitz, Practical approaches to construct RS-HDMR component functions, J. Phys. Chem. A 106 (2002) 8721–8733, https://doi.org/10.1021/jp014567t.

[93] D. Hankins, J.W. Moskowitz, F.H. Stillinger, Water molecule interactions, J. Chem. Phys. 53 (2003) 4544–4554, https://doi.org/10.1063/1.1673986.

[94] E. Clementi, W. Kołos, G.C. Lie, G. Ranghino, Nonadditivity of interaction in water trimers, Int. J. Quantum Chem. 17 (1980) 377–398, https://doi.org/10.1002/qua.560170302.

[95] S.S. Xantheas, Ab initio studies of cyclic water clusters (H2O)n, n=1–6. II, Anal. many-body Interact., J. Chem. Phys. 100 (1994) 7523–7534, https://doi.org/10.1063/1.466846.

[96] G.A. Cisneros, K.T. Wikfeldt, L. Ojamäe, J. Lu, Y. Xu, H. Torabifard, A.P. Bartók, G. Csányi, V. Molinero, F. Paesani, Modeling molecular interactions in water: from pairwise to many-body potential energy functions, Chem. Rev. 116 (2016) 7501–7528, https://doi.org/10.1021/acs.chemrev.5b00644.

[97] S. Manzhos, K. Nakai, K. Yamashita, Three-body interactions in clusters CO–(pH2)n, Chem. Phys. Lett. 493 (2010) 229–233, https://doi.org/10.1016/j.cplett.2010.05.055.

[98] G. Rauhut, Efficient calculation of potential energy surfaces for the generation of vibrational wave functions, J. Chem. Phys. 121 (2004) 9313–9322, https://doi.org/10.1063/1.1804174.

[99] P. Meier, M. Neff, G. Rauhut, Accurate vibrational frequencies of borane and its isotopologues, J. Chem. Theory Comput. 7 (2011) 148–152, https://doi.org/10.1021/ct1004752.

[100] Y. Scribano, D.M. Benoit, Iterative active-space selection for vibrational configuration interaction calculations using a reduced-coupling VSCF basis, Chem. Phys. Lett. 458 (2008) 384–387, https://doi.org/10.1016/j.cplett.2008.05.001.

[101] O. Vendrell, F. Gatti, D. Lauvergnat, H.-D. Meyer, Full-dimensional (15-dimensional) quantum-dynamical simulation of the protonated water dimer. I. Hamiltonian setup and analysis of the ground vibrational state, J. Chem. Phys. 127 (2007), 184302, https://doi.org/10.1063/1.2787588.

[102] J.N. Murrell, S. Carter, S.C. Farantos, P. Huxley, A.J.C. Varandas, Molecular Potential Energy Functions, Wiley, 1984.

[103] J.M. Bowman, S. Carter, X. Huang, MULTIMODE: a code to calculate rovibrational energies of polyatomic molecules, Int. Rev. Phys. Chem. 22 (2003) 533–549, https://doi.org/10.1080/0144235031000124163.

[104] H. Rabitz, Ö.F. Aliş, J. Shorter, K. Shim, Efficient input—output model representations, Comput. Phys. Commun. 117 (1999) 11–20, https://doi.org/10.1016/S0010-4655(98)00152-0.

[105] S. Manzhos, K. Yamashita, T. Carrington, Fitting sparse multidimensional data with low-dimensional terms, Comput. Phys. Commun. 180 (2009) 2002–2012, https://doi.org/10.1016/j.cpc.2009.05.022.

[106] S. Manzhos, T. Carrington, Using redundant coordinates to represent potential energy surfaces with lower-dimensional functions, J. Chem. Phys. 127 (2007), 014103, https://doi.org/10.1063/1.2746846.

[107] S. Manzhos, T. Carrington, Using neural networks, optimized coordinates, and high-dimensional model representations to obtain a vinyl bromide potential surface, J. Chem. Phys. 129 (2008), 224104, https://doi.org/10.1063/1.3021471.

[108] S. Manzhos, K. Yamashita, T. Carrington, Extracting Functional Dependence from Sparse Data Using Dimensionality Reduction: Application to Potential Energy Surface Construction, in: A.N. Gorban, D. Roose (Eds.), Coping with Complexity: Model Reduction and Data Analysis, Springer, Berlin, Heidelberg, 2011, pp. 133–149, https://doi.org/10.1007/978-3-642-14941-2_7.

[109] D. Duvenaud, H. Nickisch, C.E. Rasmussen, Additive Gaussian Processes, in: Advances in Neural Information Processing Systems, 2011: pp. 226–234. https://arxiv.org/abs/1112.4394v1 (accessed November 24, 2021).

[110] S. Manzhos, E. Sasaki, M. Ihara, 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. 3 (2022) 01LT02, https://doi.org/10.1088/2632-2153/ac4949.

[111] O. Ren, M.A. Boussaidi, D. Voytsekhovsky, M. Ihara, S. Manzhos, 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, Comput. Phys. Commun. 271 (2022), 108220, https://doi.org/10.1016/j.cpc.2021.108220.

[112] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, J. Chem. Phys. 134 (2011), 074106, https://doi.org/10.1063/1.3553717.

[113] I.J. Myung, Tutorial on maximum likelihood estimation, J. Math. Psychol. 47 (2003) 90–100, https://doi.org/10.1016/S0022-2496(02)00028-7.

[114] E. Brochu, V.M. Cora, N. de Freitas, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning, ArXiv:1012.2599 [Cs]. (2010). ⟨http://arxiv.org/abs/1012.2599⟩ (accessed January 5, 2022).

[115] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

[116] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[117] M. Fischetti, M. Stringher, Embedded hyper-parameter tuning by Simulated Annealing, ArXiv:1906.01504 [Cs, Math, Stat]. (2019). http://arxiv.org/abs/1906.01504 (accessed January 6, 2022).

[118] H. Alibrahim, S.A. Ludwig, Hyperparameter optimization: comparing genetic algorithm against grid search and Bayesian optimization, 2021 IEEE Congr. Evolut. Comput. (CEC) (2021) 1551–1559, https://doi.org/10.1109/CEC45853.2021.9504761.

[119] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, J. Mach. Learn. Res. 18 (2018) 1–52.

[120] S. Falkner, A. Klein, F. Hutter, BOHB: Robust and Efficient Hyperparameter Optimization at Scale, ArXiv:1807.01774 [Cs, Stat]. (2018). ⟨http://arxiv.org/abs/1807.01774⟩ (accessed January 6, 2022).

[121] A. Kamath, R.A. Vargas-Hernández, R.V. Krems, T. Carrington, S. Manzhos, Neural networks vs Gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy, J. Chem. Phys. 148 (2018), 241702, https://doi.org/10.1063/1.5003074.

[122] R.J. Bartlett, D.S. Ranasinghe, The power of exact conditions in electronic structure theory, Chem. Phys. Lett. 669 (2017) 54–70, https://doi.org/10.1016/j.cplett.2016.12.017.

[123] H. Kang, The prevention and handling of the missing data, Korean J. Anesth. 64 (2013) 402–406, https://doi.org/10.4097/kjae.2013.64.5.402.

[124] S. Manzhos, M. Ihara, Neural network with optimal neuron activation functions based on additive Gaussian process regression, ArXiv 2301 (2023) 05567, https://doi.org/10.48550/arXiv.2301.05567.

[125] R. Rojas, The Backpropagation Algorithm, in: R. Rojas (Ed.), Neural Networks: A Systematic Introduction, Springer, Berlin, Heidelberg, 1996, pp. 149–182, https://doi.org/10.1007/978-3-642-61068-4_7.

[126] B. Widrow, M.A. Lehr, 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation, Proc. IEEE 78 (1990) 1415–1442, https://doi.org/10.1109/5.58323.

[127] I.M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, USSR Comput. Math. Math. Phys. 7 (1967) 86–112, https://doi.org/10.1016/0041-5553(67)90144-9.

[128] Y. Nakamura, K. Torii, T. Munakata, Neural-network model composed of multidimensional spin neurons, Phys. Rev. E 51 (1995) 1538–1546, https://doi.org/10.1103/PhysRevE.51.1538.

[129] M. Solazzi, A. Uncini, Adaptive multidimensional spline neural network for digital equalization, Neural Netw. Signal Process. X. Proc. 2000 IEEE Signal

Process. Soc. Workshop (Cat. No. 00TH8501) vol.2 (2000) 729–735, https://doi.org/10.1109/NNSP.2000.890152.

[130] R.S. Wedemann, A.R. Plastino, Associative Memory Networks with Multidimensional Neurons, in: Artificial Neural Networks and Machine Learning – ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2022, pp. 503–514, https://doi.org/10.1007/978-3-031-15919-0_42.

[131] S. Manzhos, M. Ihara, Orders-of-coupling representation with a single neural network with optimal neuron activation functions and without nonlinear parameter optimization, ArXiv 2302 (2023) 12013v1, https://doi.org/10.48550/arXiv.2302.12013.