Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Improving sample and feature selection with principal covariates regression

To cite this article: Rose K Cersonsky *et al* 2021 *Mach. Learn.: Sci. Technol.* **2** 035038

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035038 https://doi.org/10.1088/2632-2153/abfe7c

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
18 December 2020

REVISED
23 April 2021

ACCEPTED FOR PUBLICATION
6 May 2021

PUBLISHED
14 July 2021

PAPER

Improving sample and feature selection with principal covariates regression

Rose K Cersonsky[1,*], Benjamin A Helfrecht[1], Edgar A Engel[2], Sergei Kliavinek[1] and Michele Ceriotti[1,*]

[1] Laboratory of Computational Science and Modeling, IMX, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
[2] TCM Group, Cavendish Laboratory, University of Cambridge, J.J. Thomson Avenue, Cambridge CB3 0HE, United Kingdom
[*] Authors to whom any correspondence should be addressed.

E-mail: rose.cersonsky@epfl.ch and michele.ceriotti@epfl.ch

Keywords: machine learning, feature selection, sample selection, farthest point sampling, materials science, physical chemistry, semi-supervised learning

Supplementary material for this article is available online

Abstract

Selecting the most relevant features and samples out of a large set of candidates is a task that occurs very often in the context of automated data analysis, where it improves the computational performance and often the transferability of a model. Here we focus on two popular subselection schemes applied to this end: CUR decomposition, derived from a low-rank approximation of the feature matrix, and farthest point sampling (FPS), which relies on the iterative identification of the most diverse samples and discriminating features. We modify these unsupervised approaches, incorporating a supervised component following the same spirit as the principal covariates (PCov) regression method. We show how this results in selections that perform better in supervised tasks, demonstrating with models of increasing complexity, from ridge regression to kernel ridge regression and finally feed-forward neural networks. We also present adjustments to minimise the impact of any subselection when performing unsupervised tasks. We demonstrate the significant improvements associated with PCov-CUR and PCov-FPS selections for applications to chemistry and materials science, typically reducing by a factor of two the number of features and samples required to achieve a given level of regression accuracy.

1. Introduction

In recent years, machine learning (ML) models have found application across a vast breadth of scientific fields, from economics [1–4] to medical diagnostics [5–8] to sensing [9–11] and computational chemistry [12–14]. Data-driven modelling is often discussed in a 'big data' context, where the computational cost of a ML model is of secondary importance and data is inexpensive. Nevertheless, many applications benefit significantly from reducing data requirements or accelerating training and prediction. The search for a balance between the complexity of the model, the amount of training data, and the accuracy of predictions has given rise to a subclass of ML schemes focused on subselection, wherein a subset of samples or descriptors is identified that minimises the corresponding degradation of accuracy [15, 16].

The objective of sample selection is to identify the most significant data points, effectively pruning the redundant samples and identifying ideal candidates for costlier reference calculations or analysis steps [17]. Methods may seek to find a *core-set* that is representative of the entire sample space, e.g. through Voronoi tessellations [18], committee models [19, 20] or random forests [21], or to temper the error in representing outlier or border samples, such as with sensitivity heuristics [22–24] or nearest neighbour analysis [25]. Conversely, in feature selection, one determines an information-rich subset of a large list of possible descriptors. This motivation is akin to traditional dimensionality reduction techniques like principal components analysis (PCA), which construct *new* features as combinations of the original inputs. Feature selection, on the other hand, preserves the original feature space, which can be valuable whenever descriptors hold conceptual value, e.g. sensors for autonomous robots [26], medical markers for diagnostic classification [5, 27–29], or where evaluating a large number of features for new samples is costly.

Most subselection methods are *unsupervised* and seek to exploit the diversity of the selections to maximise the corresponding variance. For example, in farthest point sampling (FPS) one relies on the diversity of the selected vectors as measured by the mutual Euclidean distance. Selection methods based on the CUR decomposition, instead, choose the columns and/or rows of the feature matrix that provide the best low-rank approximation of the original matrix. In unsupervised selection models, the preservation of pertinent information for supervised tasks is not guaranteed, particularly in the case of poor representations or non-linear relationships between features and targets. Thus, in supervised tasks, it may be attractive to use supervised selections that employ knowledge of the regression targets to influence the choice of the samples or features.

Inspired by the principal covariates regression (PCovR) method [30], we propose a modification to the FPS and CUR approaches that combines the unsupervised component with an explicit estimation of the performance of the subselection in the context of property regression. We demonstrate the superior ability of these algorithms to select features or samples for supervised learning and discuss minor modifications that improve the performance of these subsets within unsupervised tasks. To demonstrate, we employ datasets from the atomic-scale study of molecules and materials containing features that encode structural and compositional information and are used to predict properties such as the magnetic chemical shieldings of nuclei [31, 32], energy, or the forces acting on atoms [33–35].

2. Methods

We assume that the reader is familiar with simple linear and kernel ML methods; those unfamiliar with these methods or wishing further explanation may refer to Helfrecht *et al* [36], which contains a pedagogic discussion of these methods using a similar notation.

2.1. Notation

2.1.1. $\boldsymbol{X}$ *and* $\boldsymbol{Y}$

For each system, we describe inputs by a $n_{\text{samples}} \times n_{\text{features}}$ matrix $\mathbf{X}$, where each row vector $\mathbf{x}$ contains as its entries the features of the corresponding sample. We also assume that the $n_{\text{samples}} \times n_{\text{properties}}$ matrix $\mathbf{Y}$ consists of rows (denoted $\mathbf{y}$) containing the properties corresponding to the samples in $\mathbf{X}$. Furthermore, we assume that $\mathbf{X}$ and $\mathbf{Y}$ are standardised, i.e. centred by their column means and scaled such that $\mathbf{X}$ has unit variance and each column of $\mathbf{Y}$ has variance equal to $(1/n_{\text{properties}})$. The standardisation step is not essential but ensures that the features and properties have variance on the order of 1.

2.1.2. Projectors and latent space

We use $\mathbf{T}$ to indicate a projection of data into a lower-dimensional latent space. $\mathbf{P}_{AB}$ denotes a projector from one space $\mathbf{A}$ to another space $\mathbf{B}$.

2.1.3. Matrix slices

For a general matrix $\mathbf{A}$, we denote a general subset of the elements of $\mathbf{A}$ as $\mathbf{A}_{*}$. The feature-selected subset of $\mathbf{A}$ is given as $\mathbf{A}_{\mathbf{c}}$, consisting of the M features found in columns $\mathbf{c} = (c_1, c_2, \ldots c_M)$. The sample-selected subset of $\mathbf{A}$ is given as $\mathbf{A}_{\mathbf{r}}$, consisting of the M samples found in rows $\mathbf{r} = (r_1, r_2, \ldots r_M)$. We indicate the ith row as $\mathbf{a}_i$ and the jth column as $\mathbf{A}_j$, while the jth element in the ith row is given by A_{ij}.

2.1.4. Accents and operations

We use $\hat{\mathbf{A}}$ to indicate an approximation of $\mathbf{A}$ and $\tilde{\mathbf{A}}$ for an augmentation (a matrix that is analogous to $\mathbf{A}$ but is modified to incorporate other information). We use $\mathbf{U}_A$ and $\boldsymbol{\Lambda}_A$ to represent the eigenvectors and eigenvalues of a matrix $\mathbf{A} = \mathbf{U}_A \boldsymbol{\Lambda}_A \mathbf{U}_A^T$, where $\mathbf{U}_A$ contains the eigenvectors as columns. We denote the pseudoinverse of a regularised, non-invertible matrix as $\mathbf{A}^{-}$, which is equal to $\left(\mathbf{A}^T\mathbf{A} + \lambda \mathbf{I}\right)^{-1}\mathbf{A}^T$, where λ is an appropriate conditioning or regularising parameter.

2.1.5. Loss measures

We will report different loss measures, either the relative loss $\ell_A = \frac{\|\mathbf{A} - \hat{\mathbf{A}}\|^2}{\|\mathbf{A}\|^2}$, or where we wish to give the regression loss in concrete units, the root-mean-squared error (RMSE).

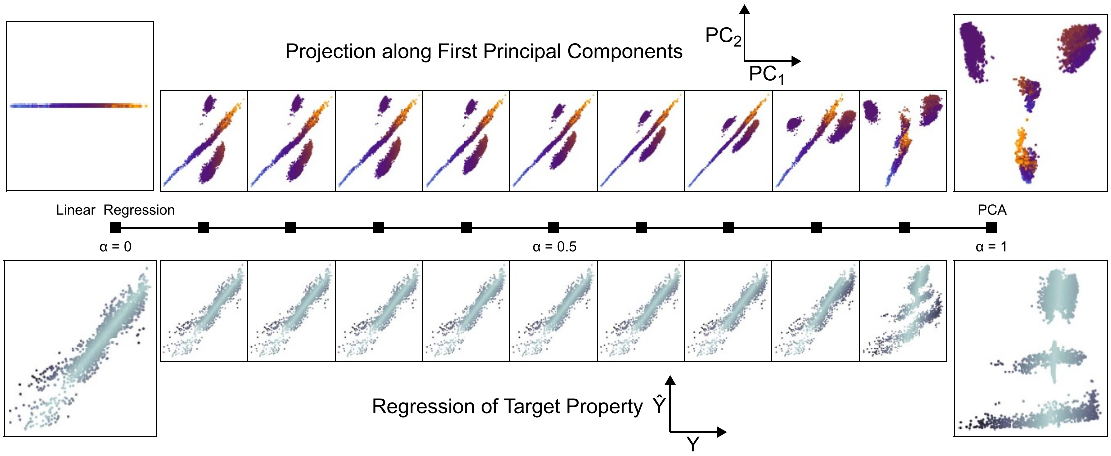


Figure 1. Example projections and regressions using the PCovR model. PCovR provides a projection into latent space which optimises a combined regression and projection loss, as weighted by the parameter α, where $\alpha = 0$ corresponds to pure linear regression and $\alpha = 1$ corresponds to PCA. Projections (top row) are given across the first two principal components and coloured by the corresponding property. $\mathbf{Y}$ vs $\hat{\mathbf{Y}}$ regression plots (bottom row) are coloured by regression error, with darker corresponding to a higher error. Intermediate values of α provide a projection which performs better in both tasks than either linear regression or PCA.

2.2. Principal covariates regression

PCovR [30] is an algorithm used to generate a latent space projection $\mathbf{T}$ that minimises a combined PCA and linear regression (LR)-like loss

$$\ell = \alpha \frac{\|\mathbf{X} - \mathbf{T}\mathbf{P}_{TX}\|^2}{\|\mathbf{X}\|^2} + (1-\alpha)\frac{\|\mathbf{Y} - \mathbf{T}\mathbf{P}_{TY}\|^2}{\|\mathbf{Y}\|^2}, \tag{1}$$

where α is a mixing parameter that determines the relative weight of the two components. Setting $\alpha = 0.0$ corresponds to LR, and $\alpha = 1.0$ corresponds to PCA.

In *sample-space PCovR*, the latent projection is determined by the modified Gram matrix of size $n_{\text{samples}} \times n_{\text{samples}}$:

$$\tilde{\mathbf{K}} = \alpha \mathbf{X}\mathbf{X}^T + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T, \tag{2}$$

where $\hat{\mathbf{Y}}$ is the result of an appropriate regression approximation of $\mathbf{Y}$ to avoid over-fitting.

However, computing the eigendecomposition $\tilde{\mathbf{K}}$ is intractable for large numbers of samples; therefore, when $n_{\text{features}} < n_{\text{samples}}$, it is advantageous to perform *feature-space PCovR*, where an equivalent latent projection is determined by the eigendecomposition of a modified covariance matrix of size $n_{\text{features}} \times n_{\text{features}}$:

$$\tilde{\mathbf{C}} = \mathbf{C}^{-1/2}\mathbf{X}^T\tilde{\mathbf{K}}\mathbf{X}\mathbf{C}^{-1/2}. \tag{3}$$

Examples of the projections and regressions obtained using PCovR, performed on the NMR Chemical Shieldings of the CSD-1000R dataset [32], are shown in figure 1. In the $\alpha = 0.0$ case, the projection is equivalent to the regression weight(s), and the second principal component is zero, as this dataset has $n_{\text{properties}} = 1$. In the $\alpha = 1.0$ case, the projection distinguishes the clusters (that are associated with the chemical identity of the atoms, namely H, C, N, O) but fails to regress the properties. For most PCovR models, an intermediate value of $\alpha \approx 0.5$ optimises the combined loss [36–38]. In the many cases in which a kernel ridge regression (KRR) model out-performs LR, one can improve the model by using its kernelised counterpart, KPCovR [36], where $\tilde{\mathbf{K}} = \alpha\mathbf{K} + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T$, with $\mathbf{K}$ being the kernel matrix and $\hat{\mathbf{Y}}$ the predicted properties obtained through KRR.

2.3. Selection methods

Selecting samples or features amounts to picking rows and columns of $\mathbf{X}$ that provide model performances comparable to that of the full feature matrix. Of the many proposed selection strategies, we will expand upon two methods: FPS and CUR decomposition.

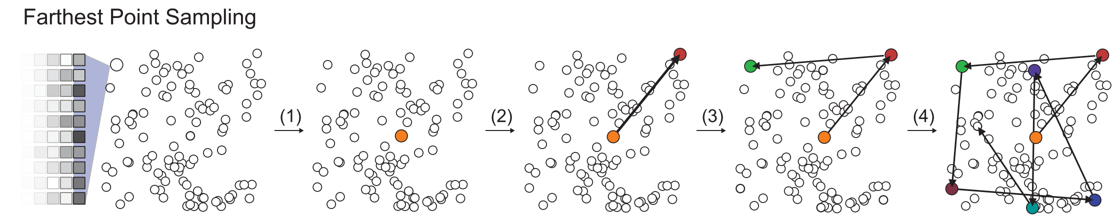


Figure 2. FPS selection. Each point corresponds to a row (for sample selection) or column (for feature selection) of the matrix. Procedure: (1) Select an initial item. (2) Select the item that is farthest from (1). (3) Select the item which is farthest from those already selected. (4) Repeat until the target number of items is selected.

2.3.1. Farthest point sampling

FPS employs a distance metric to maximise the diversity of the selection [39]. FPS is a greedy selection scheme (meaning that points are selected incrementally) and is deterministic apart from choosing the first point that is typically picked at random. Each subsequent choice is made to maximise the Haussdorf distance, i.e. the minimum distance to all previous selections

$$*_{m+1} = \mathrm{argmax}_j \left\{ \min_{i \in *_m} [\mathrm{d}(i,j)] \right\} \tag{4}$$

where $*_m$ contains the previous selections, $*_{m+1}$ is the next selected sample or feature, and $\mathrm{d}(i,j)$ indicates the distance between the ith and jth column or row. A schematic of this process is depicted in figure 2.

Even though $\mathrm{d}(i,j)$ may be defined by any metric, one often uses a Euclidean distance [40]. For sample selection, traditional FPS employs a row-wise Euclidean distance, which we give here in terms of the Gram matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$

$$\mathrm{d}_r(i,j) = K_{ii} - 2\,K_{ij} + K_{jj}. \tag{5}$$

Equation (5) simplifies the incorporation of a kernel formulation of distances rather than through an explicit set of features—by setting $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, one can perform sample-space FPS using the same procedure discussed here, in a way that is consistent with the kernel-induced metric.

The formulation of a PCovR-inspired version of FPS for sample selection is rather straightforward, as it simply involves replacing the Euclidean distance d in FPS with an augmented distance

$$\tilde{\mathrm{d}}_r(i,j) = \alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2 + (1-\alpha)\|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\|^2.$$

With this definition, the method linearly interpolates between Euclidean FPS at $\alpha = 1.0$ and one which maximises the diversity of $\mathbf{Y}$ at $\alpha = 0.0$.

By writing out explicitly the distances in terms of scalar products, one can see that this definition is equivalent to equation (5) replacing $\mathbf{K}$ with $\tilde{\mathbf{K}}$ from equation (2)

$$\tilde{\mathrm{d}}_r(i,j) = \tilde{K}_{ii} - 2\,\tilde{K}_{ij} + \tilde{K}_{jj}. \tag{6}$$

The extension to KPCov-FPS, which is warranted when the kernel is highly non-linear, is trivial and accomplished using the PCov extension of a kernel matrix $\mathbf{K}$, as discussed in Helfrecht *et al* [36].

For feature selection, the corresponding column-wise Euclidean distance can be expressed in terms of the covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$

$$\mathrm{d}_c(i,j) = C_{ii} - 2\,C_{ij} + C_{jj}, \tag{7}$$

and a feature-space version of PCov-FPS can be obtained by using a feature distance analogous to equation (7), computed in terms of $\tilde{\mathbf{C}}$, resulting in the metric $\tilde{\mathrm{d}}_c(i,j) = \tilde{C}_{ii} - 2\,\tilde{C}_{ij} + \tilde{C}_{jj}$.

2.3.2. Increasing efficiency with Voronoi FPS

Selecting m rows/columns from an $n_{\mathrm{samples}} \times n_{\mathrm{features}}$ feature matrix using FPS requires $\mathcal{O}(m \times n_{\mathrm{samples}})$ distance evaluations. We can reduce the number of distance calculations by exploiting the fact that (1) the FPS scheme implicitly partitions the rows/columns into Voronoi cells centred on the current selections and (2) the triangle inequality quickly identifies the subset of distances that needs to be re-calculated with each new selection.

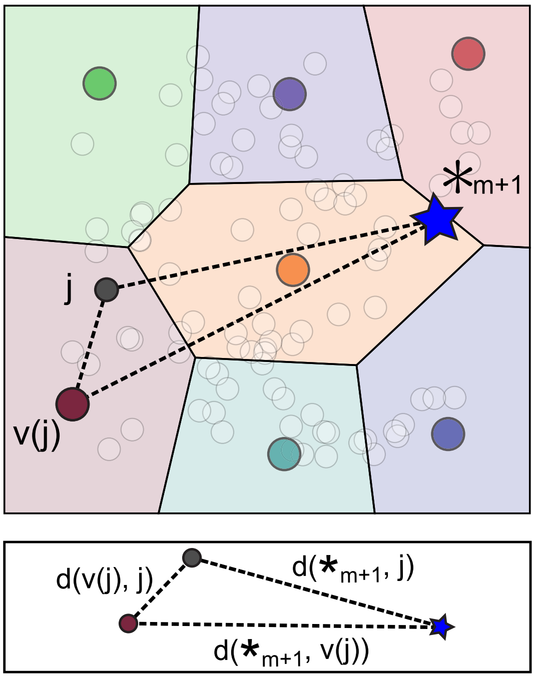


Figure 3. Schematic of Voronoi FPS. Each selected point defines an implicit Voronoi cell; this partitioning can be used to determine the points that need to be updated after each additional selection based on equation (9).

With the first selection $*_1$, we compute the distance to all remaining items, such that the Haussdorf distance of the *j*th item is assigned $h(j) = d(*_1, j)$. Also, we assign $v(j) = *_1$ to signify that each remaining item resides in the Voronoi cell of this first selection. With each subsequent selection $*_{m+1} = \text{argmax}_j h(j)$, we need only to compute the distances and update the entries of h and v for items which lie closer to the most recent selection than their previous Voronoi centre, i.e. where $\mathrm{d}(*_{m+1}, j) < h(j)$. At this point, we note that

$$\mathrm{d}(*_{m+1}, j) \geq \left|\mathrm{d}(*_{m+1}, v(j)) - \mathrm{d}(v(j), j)\right|. \tag{8}$$

It follows that the points of interest are those for which

$$h(j) > \frac{1}{2}\mathrm{d}(*_{m+1}, v(j)), \tag{9}$$

noting that $h(j) = \mathrm{d}(v(j), j)$. This application of the triangular inequality is visualised in figure 3, where we show it is unnecessary to compute $\mathrm{d}(*_{m+1} j)$ because $\mathrm{d}(v(j), j) < \frac{1}{2}\mathrm{d}(*_{m+1}, v(j))$.

The efficiency of this approach depends on several considerations. For the first few selections, the triangle-inequality bound will not reduce the number of distance calculations substantially, and as more selections are made, the number of distance calculations needed will likewise increase. Hence, a sweet spot for applying this technique involves scenarios where the total number of items is vast, and one wants to select a small fraction. Furthermore, with a Euclidean metric, it is often possible to perform fast dense operations on $\mathbf{X}$, and the random access needed to make use of the Voronoi scheme might be detrimental unless it avoids a substantial number of distance evaluations. On can automatically determine the switching point to fall back on full distance evaluation. In SI section S1.1 (available online at stacks.iop.org/MLST/2/035038/mmedia), we report the benchmarks in terms of the number of assessments of d^*—which reflects the scaling when employing a computationally-intensive distance metric that cannot be computed efficiently in terms of dense matrix operations.

2.3.3. CUR decomposition

CUR decomposition [41] aims to approximate a matrix $\mathbf{X}$ using a subset of columns and rows, such that

$$\hat{\mathbf{X}} \approx \mathbf{X_c}\left(\mathbf{X_c^-}\mathbf{X}\mathbf{X_r^-}\right)\mathbf{X_r}. \tag{10}$$

Typically in CUR, $\mathbf{X_c}$, $(\mathbf{X_c^-}\mathbf{X}\mathbf{X_r^-})$, and $\mathbf{X_r}$ are denoted $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{R}$, however we have changed notation to avoid confusion with the covariance matrix $\mathbf{C}$ and eigenvectors $\mathbf{U}$. For a given choice of rows and columns, equation (10) gives the best approximation of the original feature matrix in terms of $\mathbf{X_c}$ and $\mathbf{X_r}$. Various

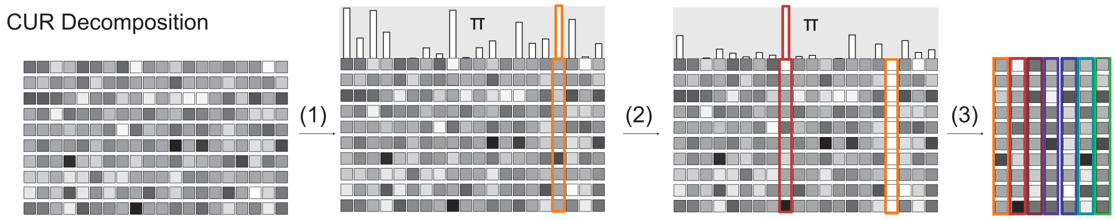


Figure 4. CUR selection. The figure demonstrates CUR *feature* selection. Sample selection proceeds identically with the matrix rows used in place of the columns. (1) Compute the importance score π for each column; select the feature that maximises π. (2) Orthogonalise the matrix with respect to selected feature; recompute π; select the feature which maximises π. (3) Repeat step (2) until the target number of items is obtained.

implementations of CUR mainly differ by the strategy for selecting **c** and **r**. Many flavours of CUR, including that of Mahoney and Drineas [41], incorporate an element of randomness in the selection—mostly to improve performance in the limit of large data sets. The subsets of rows and columns are usually determined incrementally, computing at each stage a *leverage score* π, representative of the relative importance of each column or row,

$$*_{m+1} = \operatorname{argmax}_j \left\{\pi_j\right\}. \tag{11}$$

After having selected the entry j for which π_j is highest, we orthogonalise the remaining columns or rows with respect to this selection, a wrapping procedure first introduced in Imbalzano *et al* [40]. This deterministic approach generally out-performs the more traditional approach, wherein one selects all features in a single iteration, as further demonstrated in figure S3. A schematic of feature selection using CUR is shown in figure 4.

In the most common form of CUR, the leverage score is computed from the singular value decomposition [42, 43] of the feature matrix, $\mathbf{X} = \mathbf{U_K}\boldsymbol{\Lambda}^{1/2}\mathbf{U_C}^T$, where the eigenvector subscripts denote the Gram and covariance matrices $\mathbf{K}$ and $\mathbf{C}$. For selecting samples, π is the sum over the squares of the first k components of the left singular vectors

$$\pi_i = \sum_j^k \left(\mathbf{U_K}\right)_{ij}^2, \tag{12}$$

and for feature selection the right singular vectors,

$$\pi_i = \sum_j^k \left(\mathbf{U_C}\right)_{ij}^2. \tag{13}$$

To incorporate PCovR into CUR-based selection, we propose computing the leverage scores using $\mathbf{U}_{\tilde{\mathbf{K}}}$ and $\mathbf{U}_{\tilde{\mathbf{C}}}$ in place of the left and right singular vectors. This is motivated by the fact that $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{K}}$ share the same relationship as $\mathbf{C}$ and $\mathbf{K}$, and that one could define PCovR-style features $\tilde{\mathbf{X}}$ whose singular value decomposition (SVD) yields $\mathbf{U}_{\tilde{\mathbf{K}}}$ and $\mathbf{U}_{\tilde{\mathbf{C}}}$ as left and right singular vectors (this is briefly discussed in appendix A, with the full derivation in S1.2). The number k of singular vectors included in computing the leverage score should usually be small; we obtain the best results using $k \leq n_{\text{properties}}$ eigenvectors, as demonstrated in figure S4.

After each iteration we orthogonalise the remaining samples with respect to the most recently selected row $\mathbf{x}_r$

$$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{X}\left(\frac{\mathbf{x}_r^T\mathbf{x}_r}{||\mathbf{x}_r||^2}\right), \tag{14}$$

or in feature selection relative to the most recently selected column $\mathbf{X_c}$

$$\mathbf{X} \leftarrow \mathbf{X} - \left(\frac{\mathbf{X_c}\mathbf{X_c}^T}{||\mathbf{X_c}||^2}\right)\mathbf{X}. \tag{15}$$

For PCovR-inspired CUR selection, the only additional change involves eliminating at each step the components of the property matrix that the selected features or samples can describe. For sample selection, we subtract from the property matrix the result of a regression trained on the selected samples

$$\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} - \mathbf{X}\mathbf{X}_{\mathbf{r}}^{-}\hat{\mathbf{Y}}_{\mathbf{r}}. \tag{16}$$

And for feature selection, one should perform the update

$$\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{Y}} - \mathbf{X}_{\mathbf{c}}\mathbf{X}_{\mathbf{c}}^{-}\hat{\mathbf{Y}}, \tag{17}$$

so that the next iteration of the CUR selection identifies the features that are best suited to describe the residual error in the predicted properties.

2.3.4. Improving the efficiency of CUR

The deterministic CUR scheme is particularly demanding, since it involves in each iteration reconstructing $\tilde{\mathbf{K}}$ (equation (2)) or $\tilde{\mathbf{C}}$ (equation (3)) to only use the first k eigenvectors to compute the importance score π. In unsupervised CUR one can use an iterative algorithm to determine the top eigenvectors, that—for large numbers of features—can be much faster than full eigendecomposition. Furthermore, the orthogonalisation steps outlined in equations (14) and (15) constitute *rank-one downdates* to $\mathbf{K}$ and $\mathbf{C}$ and their updated eigendecompositions can be computed using the method outlined in Bunch, Nielsen, and Sorensen [44] or Gu and Eisenstat [45]. Similar low-rank updates could be used to compute the eigenvectors of $\tilde{\mathbf{K}}$ given those of $\mathbf{K}$ and update the inverse square root $\mathbf{C}^{-1/2}$ that enters the calculation of $\tilde{\mathbf{C}}$. A further discussion of these considerations, and preliminary benchmarks of their implementation, are provided in SI section S1.3.

3. Results

In this paper we assess the performance of PCov-inspired FPS and CUR covering several scenarios that are common in supervised learning. In particular, we benchmark sample selection as a strategy to reduce the training set size (3.1.2), and active set selection for sparse KRR methods (3.1.3). We benchmark feature selection in the context of linear (3.2.2), kernel (3.2.3), and complex non-LR problems (3.3). Even though we focus on the impact of sample and feature selection on the regression performance, we also discuss the implications for unsupervised tasks in sections 3.1.1 and 3.2.1. For every model and task, we compare an entirely random selection with PCov-CUR and PCov-FPS across different values of α, again noting that $\alpha = 1$ corresponds to standard FPS and CUR. Due to the random initialisation of (PCov-)FPS, we perform multiple PCov-FPS selections and report average errors for each α. Unless otherwise stated, for all supervised models, the hyperparameters are optimised by two-fold cross-validation.

While our approach is completely general, we focus here on a benchmark system that is relevant for applications of ML to atomistic simulations, chemistry, and materials science—a field in which feature and sample selection, particularly using FPS and CUR, has lately become an increasingly common practice. In sections 3.1 and 3.2, we focus on the CSD-1000r [46] dataset that contains C, H, N, O atomic environments taken from 1000 crystal structures of molecular compounds, and their NMR chemical shieldings as target properties. We describe the atom-centred environments in terms of the smooth overlap of atomic positions (SOAP) power spectrum [47] computed with *librascal* [48], which have been previously employed for similar ML tasks [32, 36, 49]. This is a particularly relevant application because the number of SOAP features can be increased systematically, and the high number of resulting features is the main reason for the comparably high computational cost of the resulting regression models [40, 50, 51]. Models were trained on identical full training sets of 11 854 environments, and we report errors for a separate test set of 1317 environments.

In section 3.3, we also provide a distinct example of the use of PCov-CUR feature selection for force field construction, training a feed-forward neural network (NN) based on Behler–Parrinello atom-centred symmetry functions (SFs) [33, 52]. We predict energy and forces for a data set containing 10 000 benzene configurations, including four different benzene polymorphs, and using 90% of the structures for training and 5% each for validation and testing, respectively.

3.1. Sample selection

3.1.1. Preserving the covariance for unsupervised tasks

We begin by considering how selecting a subset of the structures, with and without incorporating a PCov component, impacts performance in unsupervised tasks, and how to reduce such impact. The reduced training set size potentially leads to a reduced rank of the covariance matrix computed from $\mathbf{X}_{\mathbf{r}}$, $\mathbf{C}_{\mathbf{r}} = \mathbf{X}_{\mathbf{r}}^T\mathbf{X}_{\mathbf{r}}$ but also to a skewed weighting of the importance of different features in the covariance built on the training set. The latter can be mitigated by introducing a correction based on the matrix decomposition in equation (10) to obtain a modified subset $\tilde{\mathbf{X}}_{\mathbf{r}}$ to better preserves the sample covariance, where

$$\tilde{\mathbf{C}}_{\mathbf{r}} = \tilde{\mathbf{X}}_{\mathbf{r}}^T\tilde{\mathbf{X}}_{\mathbf{r}} = \mathbf{X}_{\mathbf{r}}^T(\mathbf{X}_{\mathbf{r}}^{-})^T\mathbf{X}^T\mathbf{X}\mathbf{X}_{\mathbf{r}}^{-}\mathbf{X}_{\mathbf{r}}, \tag{18}$$

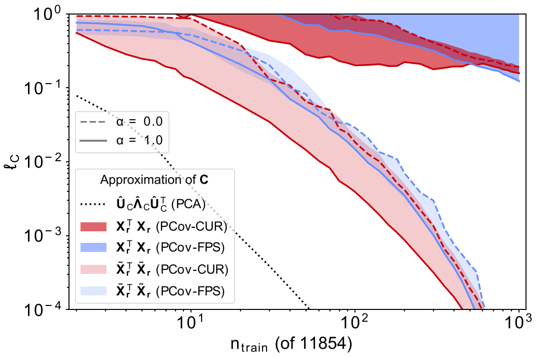


Figure 5. Approximation of covariance matrix $\mathbf{C}=\mathbf{X}^T\mathbf{X}$ using $\mathbf{X}_\mathbf{r}$ and $\tilde{\mathbf{X}}_\mathbf{r}$ for PCov-CUR and PCov-FPS on CSD-1000R. Shaded regions bound the minimum and maximum error across all α for each PCovR-inspired method. The dotted line corresponds to the PCA-type approximation of **C** from the n_{train} eigenpairs corresponding to the largest eigenvalues.

which can then be diagonalised to compute a modified PCA projection matrix. If necessary, one could also evaluate the corrected feature matrix explicitly,

$$\tilde{\mathbf{X}}_\mathbf{r}=\left[(\mathbf{X}_\mathbf{r}^-)^T\mathbf{X}^T\mathbf{X}\mathbf{X}_\mathbf{r}^-\right]^{1/2}\mathbf{X}_\mathbf{r},\tag{19}$$

which can be useful if one wants to perform feature selection based on a reduced train set.

Figure 5 shows the error in reproducing the covariance matrix with reduced sample sets. For each method, the shading indicates the range spanned as α goes from 1 (indicated with full lines, equivalent to the standard unsupervised selection) to 0 (indicated with dashed lines, giving full weight to the supervised component). We can compare these results to the approximation of **C** by its eigendecomposition $\hat{\mathbf{C}}=\hat{\mathbf{U}}_C\hat{\mathbf{\Lambda}}_C\hat{\mathbf{U}}^T$, computed with the top n_{samples} eigenvalues and their corresponding eigenvectors. Computing the covariance using the sample set $\mathbf{X}_\mathbf{r}$ results in large error ($\ell_C\approx 0.3$ with 1000 samples), with convergence only as n_{samples} approaches n_{features} (2520 in this case). Computing the covariance with $\tilde{\mathbf{X}}_\mathbf{r}$ reduces the loss considerably, with a covariance loss of $\ell_C=0.01$ possible with as few as 60 of the 11 854 of the training points. PCov-CUR typically out-performs PCov-FPS, and for both, the approximation degrades as $\alpha\rightarrow 0$.

3.1.2. Training set selection

When choosing the most important training points out of a large pool of candidates, a fully-unsupervised scheme offers the clear advantage that one does not need to compute or measure the properties in advance, which is usually the time-consuming step. However, one can often obtain inexpensively an approximate estimate of **Y**, which can be used with PCov-based selection methods to reduce the number of accurate reference evaluations. Similarly, one may want to select samples from an existing training set to use them for more demanding data analytics, e.g. picking the most relevant snapshots from a molecular dynamics trajectory to use for feature selection or non-linear dimensionality reduction.

In figure 6, we show the property regression error on a fixed randomly-selected test set of environments for models trained on different subselections of the complete train set. For each subselection, we train a linear ridge regression model

$$\hat{\mathbf{Y}}=\mathbf{X}\mathbf{X}_\mathbf{r}^-\mathbf{Y}_\mathbf{r}.\tag{20}$$

The curves in the figure compare the convergence with train set size for a random selection (the usual construction of a learning curve) with that obtained by different PCov-inspired methods.

Interestingly, when employing PCov-FPS, the unsupervised selection ($\alpha=1$) performs best, usually being the only values that (marginally) improves upon a random selection, with errors increasing with decreasing α. For PCov-CUR, the supervised and unsupervised methods perform comparably in terms of mean error, with a slight decrease in error possible as $\alpha\rightarrow 0$. In order to fully rationalise the performance of the different methods, it is necessary to consider the error distribution rather than just the mean error. As shown in figure 7 the supervised limit of PCov-CUR reduces significantly the tails of the error distribution, indicating they provide more robust models that are less susceptible to the presence of outliers in the test set. The PCov-FPS($\alpha=0$) selection results in a flatter distribution of errors, which has the smallest maximum error, but a more significant mean error, as seen clearly in figure 6. In this example, one must consider that the (full) training and test set are obtained by randomly selecting environments from the same dataset. Thus, since a random subselection of the train set has the same makeup as the test target, methods such as FPS and

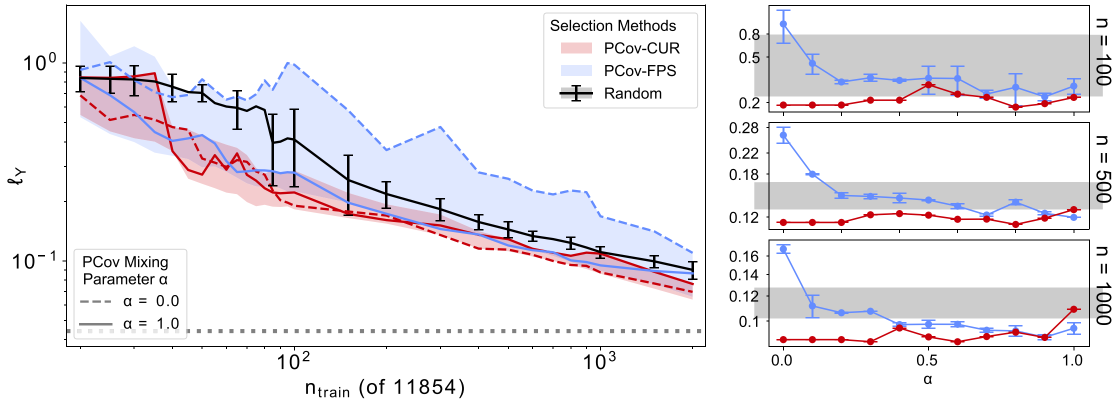


Figure 6. Mean relative error using CSD-1000R training subsets. Metrics shown are PCov-CUR (red), PCov-FPS with random r_1 (blue), and random feature selection (black, solid). Shaded regions bound the minimum and maximum error across all α for each PCovR-inspired method. In the left panel the grey dashed line denotes the error using the full training set. The right side panels show performance across α at $n_{\text{samples}} = 100, 500, 1000$ samples.

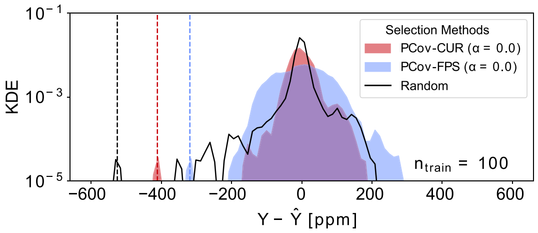


Figure 7. Kernel density estimate (KDE) of errors for random (black), PCov-CUR ($\alpha = 0$, red) and PCov-FPS ($\alpha = 0$, blue) selected training subsets, for $n_{\text{train}} = 100$ (of 11 854). The dotted lines represent the maximum absolute error for each respective training subset. Results for further α and n_{train} are in figure S7.

CUR are not guaranteed to match or improve upon the mean error compared to random selection, consistent with the observations in [53] for a KRR model of the atomisation energy of small organic molecules.

3.1.3. Active set selection for sparse kernels

Another context in which one wants to perform sample selection is when building a sparse kernel model, using the projected-process approximation [54]. In sparse KRR, one defines an ansatz for the properties of a sample as

$$\mathbf{y}(\mathbf{x}) \approx \sum_{i \in M} \mathbf{w}_i k(\mathbf{x}, \mathbf{x}_i), \tag{21}$$

where M indicates a set of 'reference samples' that effectively constitute a basis to expand $\mathbf{y}(\mathbf{x})$. The weights $\mathbf{w}_i$ are optimised to minimise the regression error on the train set. In the projected-process approximation, the reference samples $\mathbf{X_r}$ (also known as 'active points') correspond to a sample-selected subset of the training set. If we denote $\mathbf{K}$ as the kernel matrix between a dataset, described by the feature matrix $\mathbf{X}$, and itself; $\mathbf{K_r}$ as the kernel between $\mathbf{X}$ and $\mathbf{X_r}$; $\mathbf{K_{rr}}$ as the kernel computed between $\mathbf{X_r}$ and itself, the values of the predicted properties for the train set are given by

$$\hat{\mathbf{Y}} = \mathbf{K_r}(\mathbf{K_r}^T \mathbf{\Lambda}^{-1} \mathbf{K_r} + \mathbf{K_{rr}})^{-1} \mathbf{K_r}^T \mathbf{\Lambda}^{-1} \mathbf{Y}, \tag{22}$$

where $\mathbf{\Lambda}$ is a regularisation matrix, typically taken to be a scalar, in which case (22) reduces to that reported in [36]. In this example, we use the radial basis function (RBF) kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$, and we choose $\gamma = 10^{-4}$, which optimises the combined regression and reconstruction loss for the full feature vectors (see S8b).

Figure 8 compares the performance of a sparse KRR as a function of the number of reference points selected by different techniques. All data-driven selection methods out-perform by up to a factor of two the random baseline, particularly in the small active-set limit. CUR marginally out-performs FPS at all but the smallest active set sizes and PCov-inspired methods generally perform better than their unsupervised counterparts, with an error decreasing further as $\alpha \to 0$.

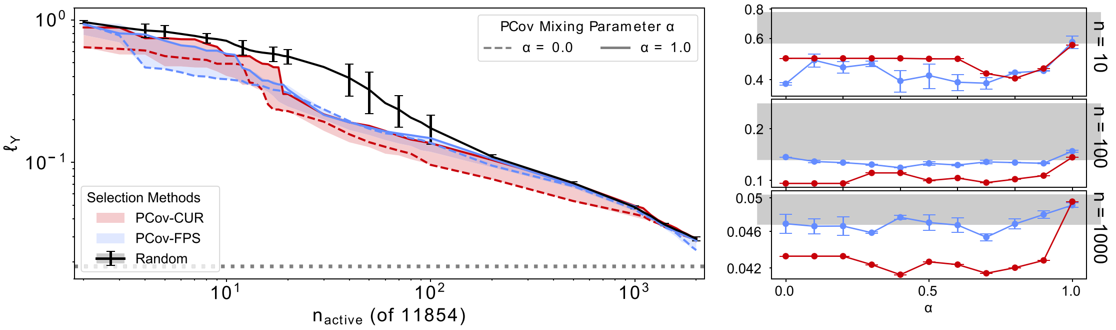


Figure 8. Regression loss of CSD-1000R using sparse KRR with a sample-selected active set. Metrics shown are PCov-CUR (red), PCov-FPS with random r_1 (blue), and random feature selection (black, solid). Shaded regions bound the minimum and maximum error across all α for each PCovR-inspired method. In the left panel, the grey dashed line denotes the error using the full kernel. The right side panels show the performance at $n_{\mathrm{samples}} = 10, 100, 1000$ (of 11 854).

The sample selection that underlies figure 8 is the same as for the train set selection in the previous section, i.e. based on a linear PCovR framework based on the raw $\mathbf{X}$ features. That the representative samples chosen with a linear framework are effective for a non-linear kernel model is vital, as in most cases, one wants to use a fixed subselection while tuning the model, e.g. by optimising hyperparameters or testing different kernels, and underscores the robustness of the selection criteria. If one wanted to select an active set consistent with the kernel-induced metric, it would suffice to substitute the kernel matrix to the matrix of scalar products in equation (5) or the leverage scores equation (12).

3.2. Feature selection

The techniques employed in section 3.1 to select the most representative samples can also be used to identify the features that provide the most information about the training data set and—with a PCov component—about structure-property relations. Selecting the most relevant features is beneficial because the cost of evaluating $\mathbf{x}$ usually scales with n_{features}, and the cost of evaluating a model built on $\mathbf{x}$ similarly increases with the size of the feature vector.

3.2.1. Assessing information richness and preserving distances in feature space

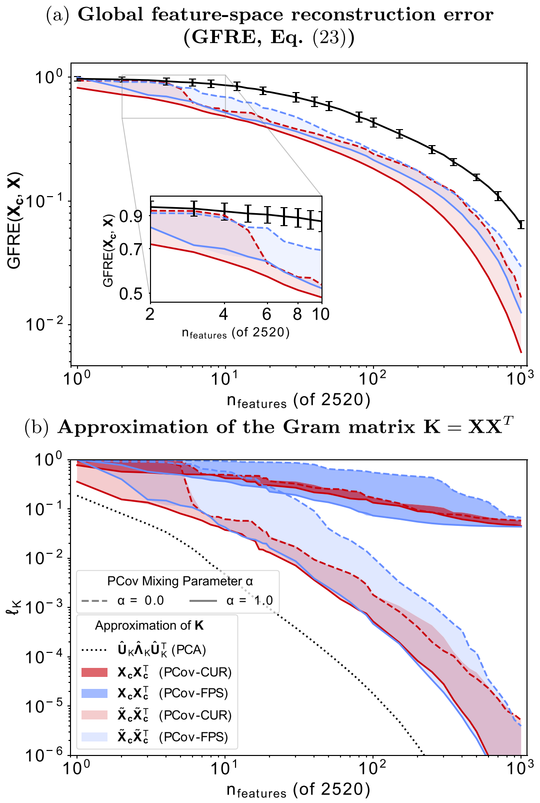
We begin our assessment of the performance of feature selection schemes by verifying whether the chosen subset of features contains comparable amounts of information to the full feature vector. A quantitative, albeit unsupervised, measure of the relative information content of two sets of features is given by the recently-introduced *global feature-space reconstruction error* (GFRE) [55]

$$\mathrm{GFRE}(\mathbf{A},\mathbf{B}) = \sqrt{\left\|\mathbf{B} - \mathbf{A}\mathbf{P}_{AB}\right\|^2 / n_{\mathrm{samples}}}, \tag{23}$$

where $\mathbf{A}$ and $\mathbf{B}$ are different, standardised featurisations representing the same dataset. In this exercise, $\mathbf{A}$ is the feature subset $\mathbf{X_c}$ and $\mathbf{B}$ is the full set of features $\mathbf{X}$. We construct $\mathbf{P}_{AB}$ using the representations of the training set and evaluate $\mathrm{GFRE}(\mathbf{X_c}, \mathbf{X})$ for the testing set.

Figure 9(a) shows that the GFRE decreases rather slowly with the number of selected features: this is to be expected as SOAP power spectrum features are linearly independent, which in combination with a diverse dataset containing many different types of chemical environments leads to high intrinsic dimensionality of the feature space. In both the small and large n_{features} limit, using a PCov-augmented scheme with $\alpha < 1$ leads to a degradation of the GFRE, while for intermediate n_{features}, the effect of α is small. This is unsurprising because the GFRE reflects only the feature vectors' information content and not the regression accuracy: the $\alpha = 1$ case is the most compatible with minimising the error in reconstructing the full feature vectors. Despite this fact, all data-driven selection schemes systematically reduce the GFRE compared to a random selection, including the $\alpha \to 0$ limit. CUR generally out-performs FPS, although it is more sensitive to an increase in the supervised component's weight.

When one wants to use $\mathbf{X_c}$ in the context of unsupervised-learning algorithms that depend on preserving the value of the scalar products—and hence the distances—between feature vectors, it necessary to incorporate a linear transformation analogous to that discussed for the case of sample selection. Imagine the case in which two features are identical—dropping one would entail no information loss but would distort distances in feature space. Scaling the retained feature by $\sqrt{2}$ would restore exactly the original metric. Regardless of the feature selection method, one can use the matrix decomposition in equation (10) to obtain a modified subset $\tilde{\mathbf{X}}_{\mathbf{c}}$ which better preserves the Euclidean distances in feature space.


(a) **Global feature-space reconstruction error (GFRE, Eq.** (23))

(b) **Approximation of the Gram matrix** $\mathbf{K} = \mathbf{X}\mathbf{X}^T$

Figure 9. Performance of feature subsets of CSD-1000R in unsupervised tasks. Metrics shown are PCov-CUR (red), PCov-FPS with random c₁ (blue), and random feature selection (black, solid). Shaded regions bound the minimum and maximum error across all α for each PCovR-inspired method.

We again start with the approximation of $\mathbf{X}$ in equation (10) and assume $\mathbf{X_r} = \mathbf{X}$ to construct $\tilde{\mathbf{X}}_\mathbf{c}$ such that $\tilde{\mathbf{X}}_\mathbf{c}\tilde{\mathbf{X}}_\mathbf{c}^T \approx \mathbf{X}\mathbf{X}^T$,

$$\tilde{\mathbf{X}}_\mathbf{c} = \mathbf{X}_\mathbf{c}\left[\mathbf{X}_\mathbf{c}^-\mathbf{X}\mathbf{X}^T(\mathbf{X}_\mathbf{c}^-)^T\right]^{1/2}. \tag{24}$$

The $n_\mathrm{features} \times n_\mathrm{features}$ matrix $\left[\mathbf{X}_\mathbf{c}^-\mathbf{X}\mathbf{X}^T(\mathbf{X}_\mathbf{c}^-)^T\right]^{1/2}$ can be computed once and re-used every time one needs to compute $\tilde{\mathbf{X}}_\mathbf{c}$, even with out-of-sample data. Figure 9(b) shows the error in reproducing the Gram matrix $\mathbf{X}\mathbf{X}^T$ with a feature-vector of reduced dimensionality as a measure of the distortion in feature-space metrics. The error one incurs when truncating a PCA latent space is, by construction, the minimal that one can achieve with a linear n_features-dimensional projection of $\mathbf{X}$. Feature selection leads to a large distortion of the underlying metric, with $\ell_K \approx 0.1$ even with more almost 50% of the features included in $\mathbf{X_c}$. The use of a correction to the selected feature matrix, as in equation (24), improves the accuracy dramatically in preserving the feature-space metric, even though asymptotically, a PCA projection out-performs column-selected features by up to a factor of ten.

3.2.2. Linear ridge regression

The advantages of using PCov-augmented feature selection are most apparent when considering their application to regression tasks. To assess the performance of a given selection scheme, we consider the error in linearly approximating a target $\mathbf{Y}$ given a feature subset $\mathbf{X_c}$

$$\hat{\mathbf{Y}} = \mathbf{X}_\mathbf{c}\mathbf{X}_\mathbf{c}^-\mathbf{Y}. \tag{25}$$

The results in figure 10 demonstrate that both PCov-CUR and PCov-FPS improve the regression performance when compared to random selection, often with comparable losses achieved with ten times fewer features. As shown in the side panels, both CUR and FPS selections improve as $\alpha \to 0$ and reach full-features accuracy at $n_\mathrm{features} \approx 1000$. Typically, PCov-CUR will out-perform PCov-FPS for corresponding α.

For reference, we also include the results obtained from principal components regression, where the latent space projection from a PCA with $n_\mathrm{latent} = n_\mathrm{features}$ is used in place of $\mathbf{X_c}$ to predict the properties of the

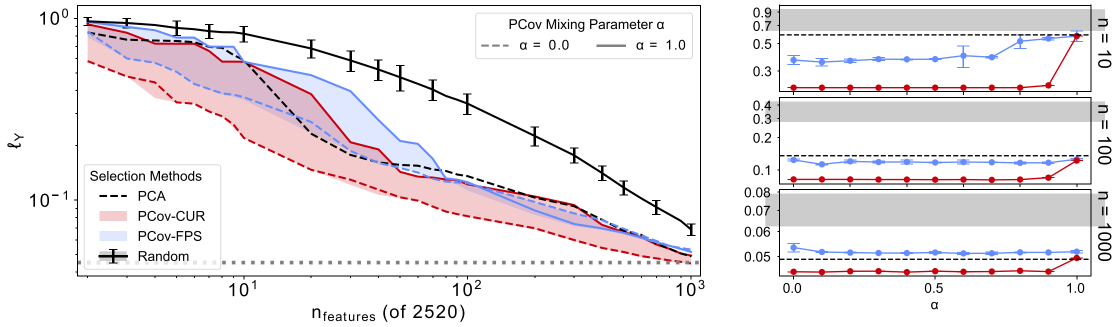


Figure 10. Ridge regression losses for different feature selection metrics for CSD-1000r Metrics shown are PCov-CUR (red), PCov-FPS with random c_1 (blue), and random feature selection (black, solid). Shaded regions bound the minimum and maximum error across all α for each PCovR-inspired method. In the left panel, the grey dashed line denotes the error using the full feature vectors. Side panels depict losses for various α values used to select 10, 100, and 1000 features. In all panels, the black dotted line denotes the regression loss for an analogous PCR.

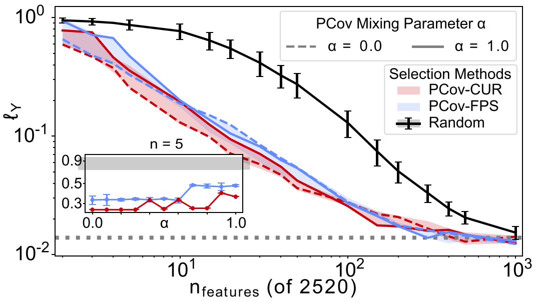


Figure 11. Regression results using low-rank kernels built on selected features. The inset shows the KRR loss for selections of five features at different α.

materials. As PCA provides the optimal unsupervised approximation of the feature vector, it serves as a baseline to assess the improvements gained by incorporating a supervised component to feature selection. Indeed, principal components regression usually performs better than unsupervised FPS and comparably to unsupervised CUR. The PCov-inspired methods consistently out-perform PCA—which supports that retaining the largest variance components does not necessarily yield features that are predictive for the properties of interest [56]. This finding is also relevant for the methods that rely on feature (co)variance to construct a hierarchy of increasingly complex representations of the atomic structure [57].

3.2.3. Kernel ridge regression

The feature-selected $\mathbf{X_c}$ can also be used to compute an approximation $\hat{\mathbf{K}}$ of a kernel matrix, by simply using the compressed feature vectors to evaluate the (non-linear) kernel function $\hat{K}_{ij} = k((\mathbf{X_c})_i, (\mathbf{X_c})_j)$. As in section 3.1.3, we use an RBF kernel with $\gamma = 10^{-4}$.

We assess the performance of the approximate kernel by fitting a KRR model $\hat{\mathbf{Y}} \approx \hat{\mathbf{K}}(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1}\mathbf{Y}$. The non-linearity in the definition of $\mathbf{K}$ indicates that regression is performed in a different feature space than $\mathbf{X}$, improving upon the regression loss of the linear model based on the full $\mathbf{X}$ by a factor of four.

Nevertheless, as shown in figure 11, kernels built on a subset of the features chosen by a PCov-FPS or PCov-CUR method out-perform those based on a random selection of equivalent size by up to an order of magnitude and match a kernel built on the full $\mathbf{X}$ with just $n_{\text{features}} \approx 400$. With an increasing number of features, the value of α has a smaller effect than in the case of LR.

3.3. NN models

Thus far, we have demonstrated the effects of hybrid supervised/unsupervised feature and sample selection for simple ML models, with a deterministic relationship between features, training set, and test error. More complex models, such as those based on artificial NNs, can reproduce an arbitrary, non-linear dependence of the target properties $\mathbf{Y}$ on the input features $\mathbf{X}$. NNs are 'trained' by iterative minimisation of the (L^2) loss between the NN output $\mathbf{Y}_{\text{NN}}$ and reference values $\mathbf{Y}$ to determine the free parameters in the network, usually called NN weights. Behler and Parrinello introduced a now commonplace NN to fit interatomic potentials [33, 52, 58] based on the decomposition of atomic configurations and total energies into local, atom-centred environments and associated energy contributions. Environments are described using two-body and three-body SFs, which correspond to a projection of two and three-body correlations between the neighbours of the target atomic centre on a bespoke non-orthogonal basis. These constitute the input layer of features $\mathbf{X}$, which is connected to narrower, fully connected 'hidden layers', and finally combined to predict an atom-centred decomposition of the potential energy of the system. Nodes are linked via non-linear activation functions f_a and each node i in layer $k-1$ is connected to each node j in layer k with a tunable weight w_{ij}^k. For a single hidden layer with N nodes, a Behler–Parrinello NN to fit a single property y can be expressed as

$$y_{\mathrm{NN}}(\mathbf{x}) = f_a^2 \left[u^2 + \sum_{n=1}^{N} w_{n1}^1 f_a^1 \left(v_n^1 + \sum_{i=1}^{f} w_{in}^1 x_i\right)\right] \tag{26}$$

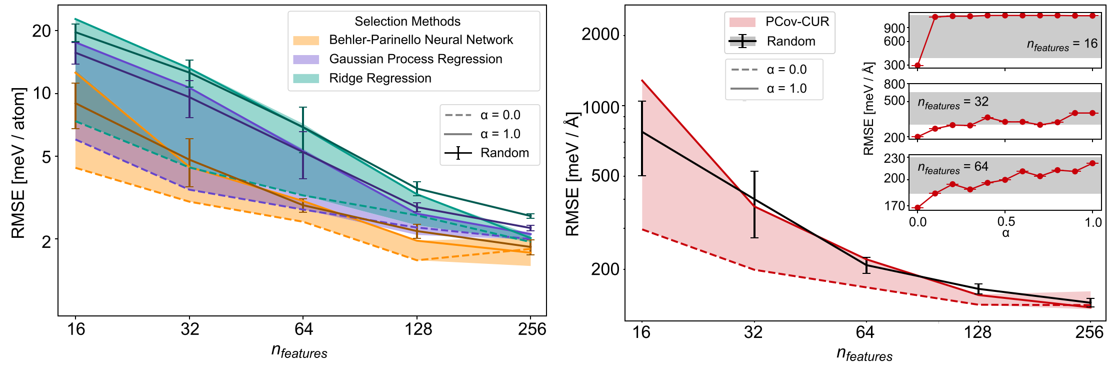
where u^k and v_j^k are adjustable offsets. Besides allowing us to test the effectiveness of our approaches in the presence of a more complicated functional relationship between features and properties, BP NN also give us an opportunity to verify the effects of including derivatives of the target (the forces) as part of the regression.

The evaluation of SFs is computationally demanding, and the number of possible two and three-body SFs increases quadratically or cubically with the number of species. Furthermore, Behler-Parrinello SFs are not orthogonal and are often highly redundant, which, together with considerations of computational efficiency, strongly motivates the selection of a small number of SFs. Traditionally, SF have been selected 'by hand' using a combination of chemical intuition and trial-and-error, but more recently, Imbalzano *et al* [40] showed that CUR and FPS provide a viable strategy to choose a small set of SFs out of a large pool of candidates. To investigate how a PCov augmentation impacts the selection of features in the context of a NN potential, we construct and test such a potential for crystalline benzene. We use the first 10 000 most structurally diverse configurations within a larger set of 55 000 FPS-ordered, thermally-distorted configurations of forms I and II of benzene, as well as the hypothetical high-pressure I_{hp} and V' polymorphs. The dataset was generated to train a potential to assess the relative (free) energy of different crystal polymorphs, and the details of its makeup and reference energetics are reported in the corresponding publication [59], as well as in the public data record [60], that also contains complete configurational energy, atomic forces, and cell stress for each configuration, as determined by semi-local, dispersion-corrected density functional theory calculations. The set of benzene molecular crystals considered here exhibit energies with a standard deviation of 63 meV atom^{-1} and an average atomic force component of 1.374 eV Å^{-1}. The minimum and maximum energies differ by as much as 540 meV atom^{-1}, while the maximum force component reaches 21.633 eV Å^{-1}. We encode the structure data in a feature matrix that consist of 452 two and three-body SF, by varying the function parameters on a regular grid, following the protocol in [40]. We focus on PCov-CUR, which has been shown to yield consistently better performance in feature selection than PCov-FPS.

We begin by discussing how the NN's non-linear nature affects the performance of the feature selection schemes. We compare LR, KRR using an RBF kernel, and the Behler–Parrinello NN, using as inputs subsets of the SFs, determined by PCov-CUR and random selection. For this purpose the 10 000 benzene configurations were randomly divided into 90:5:5 train:validation:test. All models were subsequently trained on the same training set, using only energy information, and tested on the same independent test set to determine the errors reported in figure 12. All hyperparameters involved in constructing the linear and kernel ridge models, i.e. the regularisation and the characteristic length scale of the RBF kernel, were individually optimised for each feature selection by minimising the energy RMSE for the validation set. While we might have similarly optimised the architecture of the NN potentials, simply adopting the parameters of established Behler–Parrinello type NN potentials [61–64] more than suffices to assess the impact of incorporating information regarding the target property in the feature selection. We train four NN potentials for feature selection and report the best out of the four results.

Figure 12 demonstrates the interplay between the nature of the features, the selection protocol, and the regression model. Models that incorporate increasing levels of non-linearity lead to better regression performance, with KRR usually out-performing LR by 25%, and NN reducing the energy RMSE by an additional 40%. A side-effect of using highly-correlated features is that the performance of a random selection is not particularly poor in comparison with an unsupervised CUR selection. A PCov-CUR($\alpha = 0$) selection that incorporates target information allows one to identify the most relevant features for constructing the potential, resulting in a very substantial reduction of the energy RMSE, particularly for small n_{features} values. For all values of α and number of features, PCov-CUR selections of SFs consistently out-perform (and never perform worse than) the average random selection. Furthermore, PCov-CUR selections at $\alpha = 0$, consistently out-perform unsupervised CUR selections across all regression schemes.

This improvement is particularly remarkable given the gap between the linear, energy-based supervised framework that underlies the PCov augmentation and the non-linear predictions of atomic forces. Thus, the



(a) **RMSE in energies** using Ridge Regression (green), Gaussian Process Regression (purple), and BP NN (gold).

(b) **RMSE in forces** for BP NN. Insets denote the dependence on α for 16, 32, or 64 features.

Figure 12. RMSE for energies (left) and forces (right) from various models and feature selections. The dotted and solid lines denote results for $\alpha = 0$ and $\alpha = 1$, respectively. Darker error bars and/or grey regions represent the mean and standard deviation of errors for random feature selection.

methods we propose are robust and can be applied to improve feature selection beyond linear or kernel methods. Overall, the best PCov-CUR selection reduces by 50% the number of SF while retaining roughly the same force RMSE, indicating direct computational savings when using the NN potential and a simpler, less memory-intensive task when training the model.

4. Conclusions

Selecting from a large pool of candidates the samples and/or features most relevant for an ML task can be very advantageous from a computational perspective and reveal the most important or insightful descriptors. This is particularly useful in cases in which features are constructed systematically, leading potentially to large and redundant input representations. Unsupervised methods, based on a low-rank approximation of the feature matrix or maximising diversity, provide an effective approach to prune a training set or a collection of descriptors with little loss in performance.

Whenever a featurisation is used in a supervised model, it is appealing to incorporate the regression target into the feature or sample selection by combining two methods (FPS and CUR) with a hybrid supervised/unsupervised linear scheme, PCovR. For a variety of different problems, ranging from reducing the size of a training set to active point selection to linear and non-linear model fitting, we find that such PCov-augmented selections out-perform almost universally their unsupervised counterparts, which makes it possible to obtain (with much-reduced effort) models that achieve comparable prediction accuracy to the model based on full features and training set. The simplicity of PCov-FPS and PCov-CUR, the ease by which they can be extended to kernelised schemes, and the empirical evidence showing that they can also improve the accuracy of non-linear NN models make them applicable to virtually any regression task. These results, together with the availability of an open-source implementation [65], give these methods the potential to become a standard tool in the application of data-driven methods to different fields of science.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files). The analysis of NMR chemical shieldings and benzene configurations were conducted using data contained in Refs. [49] and [60], respectively, as noted in the text.

Acknowledgment

M C, R K C, B A H acknowledge the ERC Horizon 2020 Grant No. 677013-HBMAP. E A E acknowledges support from Trinity College, Cambridge and compute time from the Swiss National Computing Centre (projects s960, s1000).

Code availability

Software for PCov-CUR or PCov-FPS can be found at www.github.com/cosmo-epfl/scikit-cosmo with documentation at scikit-cosmo.readthedocs.io.

Appendix. PCov feature space

Given the singular value decomposition of the feature matrix $\mathbf{X} = \mathbf{U}_{\mathbf{K}}\mathbf{\Lambda}^{1/2}\mathbf{U}_{\mathbf{C}}^{T}$ it is possible to define a PCov feature matrix $\tilde{\mathbf{X}} = \mathbf{U}_{\mathbf{K}}\tilde{\mathbf{L}}^{1/2}\mathbf{U}_{\mathbf{C}}^{T}$, with

$$\tilde{\mathbf{L}} = \alpha\mathbf{\Lambda} + (1-\alpha)\mathbf{U}_{\mathbf{K}}^{T}\mathbf{Y}\mathbf{Y}^{T}\mathbf{U}_{\mathbf{K}}. \tag{A1}$$

It is easy–albeit tedious (see S1.2)—to check that $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{T} = \tilde{\mathbf{K}}$ and $\tilde{\mathbf{X}}^{T}\tilde{\mathbf{X}} = \tilde{\mathbf{C}}$. In general, for $\alpha \neq 1$, the matrix $\tilde{\mathbf{L}}$ is not diagonal, and so the singular vectors of $\tilde{\mathbf{X}}$ are not given by $\mathbf{U}_{\mathbf{K}}$ and $\mathbf{U}_{\mathbf{C}}^{T}$, but can be obtained by diagonalising $\tilde{\mathbf{K}}$ or $\tilde{\mathbf{C}}$. Equation (A1) also makes it possible to diagonalise $\tilde{\mathbf{L}} = \mathbf{U}_{\tilde{\mathbf{L}}}\tilde{\mathbf{\Lambda}}\mathbf{U}_{\tilde{\mathbf{L}}}^{T}$ and compute $\mathbf{U}_{\tilde{\mathbf{K}}} = \mathbf{U}_{\mathbf{K}}\mathbf{U}_{\tilde{\mathbf{L}}}$ and $\mathbf{U}_{\tilde{\mathbf{C}}} = \mathbf{U}_{\mathbf{C}}\mathbf{U}_{\tilde{\mathbf{L}}}$,

ORCID iDs

Rose K Cersonsky https://orcid.org/0000-0003-4515-3441
Benjamin A Helfrecht https://orcid.org/0000-0002-2260-7183
Edgar A Engel https://orcid.org/0000-0003-2944-9445
Sergei Kliavinek https://orcid.org/0000-0001-8326-325X
Michele Ceriotti https://orcid.org/0000-0003-2571-2832

References

[1] Bolton R and Hand D 2002 Statistical fraud detection: a review *Stat. Sci.* **17** 235–49

[2] Fischer T and Krauss C 2018 Deep learning with long short-term memory networks for financial market predictions *Eur. J. Oper. Res.* **270** 654–69

[3] Huang Z, Chen H, Hsu C, Chen W and Wu S 2004 Credit rating analysis with support vector machines and neural networks: a market comparative study *Decis. Support Syst.* **37** 543–58

[4] Tsai C-F and Wu J-W 2008 Using neural network ensembles for bankruptcy prediction and credit scoring *Expert Syst. Appl.* **34** 2639–49

[5] Guyon I, Weston J and Barnhill S 2002 Gene selection for cancer classification using support vector machines *Mach. Learn.* **46** 389–422

[6] Peng X, Lin P, Zhang T and Wang J 2013 Extreme learning machine-based classification of ADHD using brain structural MRI data *PLoS One* **8** 11

[7] Rajkomar A *et al* 2018 Scalable and accurate deep learning with electronic health records *NPJ Digital Med.* **1** 18

[8] Wolf F A, Angerer P and Theis F J 2018 SCANPY: large-scale single-cell gene expression data analysis *Genome Biol.* **19** 15

[9] Belgiu M and Dragut L 2016 Random forest in remote sensing: a review of applications and future directions *ISPRS J. Photogramm. Remote Sens.* **114** 24–31

[10] Gramfort A, Luessi M, Larson E, Engemann D A, Strohmeier D, Brodbeck C, Parkkonen L and Haemaelaeinen M S 2014 MNE software for processing MEG and EEG data *Neuroimage* **86** 446–60

[11] Mountrakis G, Im J and Ogole C 2011 Support vector machines in remote sensing: a review *ISPRS J. Photogramm. Remote Sens.* **66** 247–59

[12] Berrueta L A, Alonso-Salces R M and Heberger K 2007 Supervised pattern recognition in food analysis *J. Chromatogr.* A **1158** 196–214

[13] Daina A, Michielin O and Zoete V 2017 SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules *Sci. Rep.* **7** 42717

[14] McGibbon R T *et al* 2015 MDTraj: a modern open library for the analysis of molecular dynamics trajectories *Biophys. J.* **109** 1528–32

[15] Blum A L and Langley P 1997 Selection of relevant features and examples in machine learning *Artif. Intell.* **97** 245–71

[16] Li J, Cheng K, Wang S, Morstatter F, Trevino R P, Tang J and Liu H 2018 Feature selection: a data perspective *ACM Comput. Surv.* **50** 94

[17] Xu X, Liang T, Zhu J, Zheng D and Sun T 2019 Review of classical dimensionality reduction and sample selection methods for large-scale data processing *Neurocomputing* **328** 5–15

[18] Du Q, Faber V and Gunzburger M 1999 Centroidal Voronoi tessellations: applications and algorithms *SIAM Rev.* **41** 637–76

[19] García-Osorio C, de Haro-García A and García-Pedrajas N 2010 Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts *Artif. Intell.* **174** 410–41

[20] Akdemir D, Sanchez J I and Jannink J-L 2015 Optimization of genomic selection training populations with a genetic algorithm *Genet. Selection Evol.* **47** 38

[21] Wang X-Z, Dong L-C and Yan J-H 2012 Maximum ambiguity-based sample selection in fuzzy decision tree induction *IEEE Trans. Knowl. Data Eng.* **24** 1491–505

[22] Widrow B and Hoff M E 1960 *Adaptive switching circuits* ;1553-1Office of Naval Research

[23] Zeng X and Yeung D S 2001 Sensitivity analysis of multilayer perceptron to input and weight perturbations *IEEE Trans. Neural Netw.* **12** 1358–66

[24] Ng W W, Yeung D S and Cloete I 2003 Input sample selection for RBF neural network classification problems using sensitivity measure *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics* vol 3 (IEEE) pp 2593–8

[25] Hart P E 1968 The condensed nearest neighbor rule *IEEE Trans. Inf. Theory* **14** 515–6

[26] Balakrishnan K and Honavar V 1996 On sensor evolution in robotics *Genetic Programming 1996* (*Stanford University 28–31 July 1996*) vol 98 pp 455–60

[27] Ding C and Peng H 2005 Minimum redundancy feature selection from microarray gene expression data *J. Bioinform. Computat. Biol.* **03** 185–205

[28] Fan Y-J and Chaovalitwongse W A 2010 Optimizing feature selection to improve medical diagnosis *Ann. Oper. Res.* **174** 169–83

[29] Chuang L Y, Chang H W, Tu C J and Yang C H 2008 Improved binary PSO for feature selection using gene expression data *Computat. Biol. Chem.* **32** 29–38

[30] de Jong S and Kiers H A L 1992 Principal covariates regression: part I. Theory *Chemometrics and Intelligent Laboratory Systems Series Proc. 2nd Symp. on Chemometrics* vol 14 pp 155–64

[31] Cuny J, Xie Y, Pickard C J and Hassanali A A 2016 Ab Initio quality NMR parameters in solid-state materials using a high-dimensional neural-network representation *J. Chem. Theory Comput.* **12** 765–73

[32] Paruzzo F M, Hofstetter A, Musil F, De S, Ceriotti M and Emsley L 2018 Chemical shifts in molecular solids by machine learning *Nat. Commun.* **9** 4501

[33] Behler J and Parrinello M 2007 Generalized neural-network representation of high-dimensional potential-energy surfaces *Phys. Rev. Lett.* **98** 146401

[34] Bartók A P, Payne M C, Kondor R and Csányi G 2010 Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons *Phys. Rev. Lett.* **104** 136403

[35] Rupp M, Tkatchenko A, Müller K-R and von Lilienfeld O A 2012 Fast and accurate modeling of molecular atomization energies with machine learning *Phys. Rev. Lett.* **108** 058301

[36] Helfrecht B A, Cersonsky R K, Fraux G and Ceriotti M 2020 Structure-property maps with Kernel principal covariates regression *Mach. Learn.: Sci. Technol.* **1** 045021

[37] Vervloet M, Van Deun K, Van den Noortgate W and Ceulemans E 2013 On the selection of the weighting parameter value in principal covariates regression *Chemometr. Intell. Lab. Syst.* **123** 36–43

[38] Vervloet M, Kiers H A L, Noortgate W V d and Ceulemans E 2015 PCovR: an R Package for principal covariates regression *J. Stat. Software* **65** 1–14

[39] Eldar Y, Lindenbaum M, Porat M and Zeevi Y 1997 The farthest point strategy for progressive image sampling *IEEE Trans. Image Process.* **6** 1305–15

[40] Imbalzano G, Anelli A, Giofré D, Klees S, Behler J and Ceriotti M 2018 Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials *J. Chem. Phys.* **148** 241730

[41] Mahoney M W and Drineas P 2009 CUR matrix decompositions for improved data analysis *Proc. Natl Acad. Sci. USA* **106** 697–702

[42] Golub G H and Reinsch C 1970 Singular value decomposition and least squares solutions *Numer. Math.* **14** 403–20

[43] Klema V and Laub A 1980 The singular value decomposition: its computation and some applications *IEEE Trans. Autom. Control* **25** 164–76

[44] Bunch J R, Nielsen C P and Sorensen D C 1978 Rank-one modification of the symmetric eigenproblem *Numer. Math.* **31** 31–48

[45] Gu M and Eisenstat S C 1994 A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem *SIAM J. Matrix Anal. Appl.* **15** 1266–76

[46] Musil F, Willatt M J, Langovoy M A and Ceriotti M 2019 Fast and accurate uncertainty estimation in chemical machine learning *J. Chem. Theory Comput.* **15** 906–15

[47] Bartók A P, Kondor R and Csányi G 2013 On representing chemical environments *Phys. Rev.* B **87** 184115

[48] Musil F, Veit M, Goscinski A, Fraux G, Willatt M J, Stricker M, Junge T and Ceriotti M 2021 Efficient implementation of atom-density representations *J. Chem. Phys.* **154** 114109

[49] Engel E A, Anelli A, Hofstetter A, Paruzzo F, Emsley L and Ceriotti M 2019 A Bayesian approach to NMR crystal structure determination *Phys. Chem. Chem. Phys.* **21** 23385–400

[50] Onat B, Ortner C and Kermode J R 2020 Sensitivity and dimensionality of atomic environment representations used for machine learning interatomic potentials *J. Chem. Phys.* **153** 144106

[51] Zuo Y *et al* 2020 Performance and cost assessment of machine learning interatomic potentials *J. Phys. Chem.* A **124** 9b08723

[52] Behler J 2011a Atom-centered symmetry functions for constructing high-dimensional neural network potentials *J. Chem. Phys.* **134** 074106

[53] Bartók A P, De S, Poelking C, Bernstein N, Kermode J R, Csányi G and Ceriotti M 2017 Machine learning unifies the modeling of materials and molecules *Sci. Adv.* **3** e1701816

[54] Rasmussen C E 2005 *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning Series)* (Cambridge, MA: MIT Press)

[55] Goscinski A, Fraux G and Ceriotti M 2020 The role of feature space in atomistic learning *Mach. Learn.: Sci. Technol.* **2** 2

[56] Jolliffe I T 1982 A note on the use of principal components in regression *J. R. Stat. Soc. Ser.* C **31** 300–3

[57] Nigam J, Pozdnyakov S and Ceriotti M 2020 Recursive evaluation and iterative contraction of *N*-body equivariant features *J. Chem. Phys.* **153** 121101

[58] Behler J 2011b Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations *Phys. Chem. Chem. Phys. PCCP* **13** 17930–55

[59] Kapil V and Engel E A 2021 A complete description of thermodynamic stabilities of molecular crystals (arxiv:2102.13598)

[60] Engel E A and Kapil V 2021 Semi-local and hybrid functional DFT data for thermalised snapshots of polymorphs of benzene, succinic acid and glycine *Mater. Cloud Arch.* **2021.51**

[61] Eshet H, Khaliullin R Z, Kühne T D, Behler J and Parrinello M 2010 Ab initio quality neural-network potential for sodium *Phys. Rev.* B **81** 184107

[62] Khaliullin R Z, Eshet H, Kühne T D, Behler J and Parrinello M 2010 Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface *Phys. Rev.* B **81** 100103

[63] Khaliullin R Z, Eshet H, Kühne T D, Behler J and Parrinello M 2011 Nucleation mechanism for the direct graphite-to-diamond phase transition *Nat. Mater.* **10** 693

[64] Cheng B, Engel E A, Behler J, Dellago C and Ceriotti M 2019 Ab initio thermodynamics of liquid and solid water *Proc. Natl Acad. Sci. USA* **116** 1110–5

[65] Cersonsky R K, Fraux G, Kliavinek S, Goscinski A, Helfrecht B A and Ceriotti M 2021 *scikit-cosmo* (available at: https://github.com/cosmo-epfl/scikit-cosmo) (https://doi.org/10.5281/zenodo.4752370)