PES-Learn: An Open-Source Software Package for the Automated Generation of Machine Learning Models of Molecular Potential Energy Surfaces

Adam S. Abbott,[†] Justin M. Turney,[†] Boyi Zhang,[†] Daniel G. A. Smith,[‡] Doaa Altarawy,[‡,¶] and Henry F. Schaefer III[*,†]

[†]*Center for Computational Quantum Chemistry, The University of Georgia, Athens, Georgia 30602 USA*

[‡]*Molecular Sciences Software Institute, Virginia Tech, Blacksburg, Virginia 24061, USA*

[¶]*Computer and Systems Engineering Department, Alexandria University, Egypt*

E-mail: ccq@uga.edu

Abstract

We introduce a free and open-source software package (PES-Learn) which largely automates the process of producing high-quality machine learning models of molecular potential energy surfaces (PESs). PES-Learn incorporates a generalized framework for producing grid points across a PES that is compatible with most electronic structure theory software. The newly generated or externally supplied PES data can then be used to train and optimize neural network or Gaussian process models in a completely automated fashion. Robust hyperparameter optimization schemes designed specifically for molecular PES applications are implemented to ensure that the best possible model for the dataset is fit with high quality. The performance of PES-Learn toward fitting a few semi-global PESs from the literature is evaluated. We also demonstrate the use of PES-Learn machine learning models in carrying out high-level vibrational configuration interaction computations on water and formaldehyde.

1 Introduction

Central to the application of electronic structure theory (EST) to molecular systems is the concept of the Born–Oppenheimer potential energy surface (PES). A great wealth of valuable chemical information can be derived from a high-quality PES, including the structure of stationary points, reaction dynamics trajectories, thermochemical properties, spectroscopic quantities, kinetic parameters, and many other chemically relevant insights. Unfortunately, several PES applications require numerous evaluations of the electronic energy across the relevant molecular configuration space, so much so that it is often impractical to utilize highly accurate EST methods. Therefore, much effort has been dedicated towards developing methods for fitting mathematical representations of PESs which are less computationally expensive to evaluate compared to the EST method from which the electronic energies are derived. Generally, such a representation requires a grid of single-point energies across relevant geometrical configurations, which inform its construction. Naturally, the number of single-point energies required will determine the practicality of constructing the representation.

Numerous approaches for representing PESs have been introduced in the literature. Some archetypal examples include various interpolation methods[1–8] and fits to several polynomial functional forms[9–15] such as the highly successful permutation-invariant polynomial (PIP) approach of Bowman and coworkers.[11,13,14]

More recently, there has been a tremendous increase in the use of machine learning (ML) techniques to model molecular PESs.[16–29] At risk of oversimplifying, this approach can be divided into two broad classes, and it is important to distinguish between the two. The first class of methods can be regarded as using ML algorithms to directly fit Born–Oppenheimer

PESs of specific molecular systems. The idea here is analogous to the previously mentioned interpolation and polynomial fitting schemes, but the fitting is performed with modern and typically non-linear ML methods.

The second class of methods are attempts to fit general potentials which are intended to work for systems of arbitrary molecular size, so long as they have a specific atomic composition. These methods typically express the total energy of the system in terms of a sum of atomic energy contributions, which depend on the local atomic neighborhood within some cutoff radius.[16,19,30–36] These have been referred to as *atomistic* or *high-dimensional* ML methods. Examples of atomistic methods include the Behler–Parrinello approach which makes use of atom-centered symmetry functions (ASFs),[16,19] and modifications of this methodology such as the ANI-1 potential[37] and the weighted-ASF approach.[30] We also refer the reader to the deep neural network architectures of Schütt and coworkers[31,32] which likewise incorporate an atomistic framework. Such approaches are not the focus of the present work, and are already available in other software packages.[38–41] Generally speaking, the ML approaches for directly fitting PESs of specific molecular systems can model small molecular systems (≤ 6 atoms) with very high accuracy (< 5 cm^{-1}) with only a few hundred to a few thousand *ab initio* energy computations. However, these methods have an upper bound of practicality in regard to the size of the molecular system that can be treated. For atomistic methods, we find the reverse to be the case: substantial amounts of data are required for high accuracy, but larger system sizes are not a problem. This is not to say that the atomistic approaches are not useful for smaller systems. Indeed, when using very large training sets of tens of thousands of *ab initio* energies, it has been recently shown that the Behler–Parrinello ASF approach can give similar performance to direct fits of PESs of small reactive systems.[29]

It has been consistently observed that ML models are capable of accurately representing system-specific PESs with far fewer points compared to traditional fitting methods. For directly fitting PESs, neural network[21,22,28,29,42,43] (NN) and Gaussian process (GP) regression[18,24–28,35,36,44–47] techniques have emerged as popular ML algorithm choices, although success has been achieved with other methods such as kernel ridge regression.[23] Recently, Kamath and coworkers compared the performance of NNs and GPs in fitting a local PES of formaldehyde (H_2CO) up to 17,000 cm^{-1} above the global minimum.[28] Training the ML models with only 625 symmetry-unique *ab initio* computations, they obtained RMS fitting errors in the neighborhood of 5 cm^{-1}. The ML models were subsequently used as potentials in vibrational configuration interaction (VCI) computations. The vibrational energy levels obtained with the ML potentials reproduced those obtained with the reference PES with sub-wavenumber accuracy. In a comparative study of PIP least-squares fitting methods to GP regression, Bowman and coworkers found that while the GPs were more expensive to train and evaluate they generally provided lower PES fitting errors for several molecular systems, though the prediction of stationary point properties were similar between the two methods.[26] Dral and coworkers demonstrated the use of kernel ridge regression toward fitting a 44,819 point surface of methyl chloride (CH_3Cl).[23] By training the model on a subset of only 10% of the dataset, the fitting error and vibrational energy levels obtained using the ML model were competitive with the reference surface which was trained on the entire dataset with a standard linear least-squares method.[48]

These examples lead one to suggest that perhaps the usage of ML models should become the norm for producing *ab initio* PESs. Intuitively, the nonlinear regression capabilities offered by ML models are better suited than linear methods for fitting nonlinear, high-dimensional functions such as PESs. However, there is currently a lack of freely available software designed to facilitate the application of ML to system-specific theoretical chemistry investigations. As a result, if one wishes to apply these powerful methodologies in research, one is required to programmatically implement it themself. This can prove to be problematic and challenging, as the complexity of ML methods creates a rather steep learning curve toward applying them with best practices. Furthermore, the lack of a central software platform creates reproducibility issues. As models in the literature are produced by different parties on a case-by-case basis, they are not easily shareable, usable, or verifiable. This is of course a hindrance in the scientific process. To alleviate these difficulties and drive further innovation, we have developed an extensible, open-source, and user-friendly software package, PES-Learn, which we have developed for use by the computational chemisty community. PES-Learn is the first software package to date designed to both generate PES data and fit system-specific Born–Oppenheimer molecular PESs with machine learning methods.

Herein, we describe the capabilities of PES-Learn which aid in developing high-quality machine learning models of potential energy surfaces. We illustrate how the capabilities of PES-Learn make ML a highly accessible and useful tool for research. We show that the ML model representations of the PESs produced by the present software can be used to obtain very accurate *ab initio* results with low computational cost. We first compare the performance of PES-Learn models to previous fitting attempts for a few benchmark PESs from the literature. Next, we demonstrate how the ML models can be utilized for high-accuracy computational chemistry applications by carrying out VCI computations on water and formaldehyde. The software is open-source and freely available on GitHub (https://github.com/CCQC/PES-Learn).

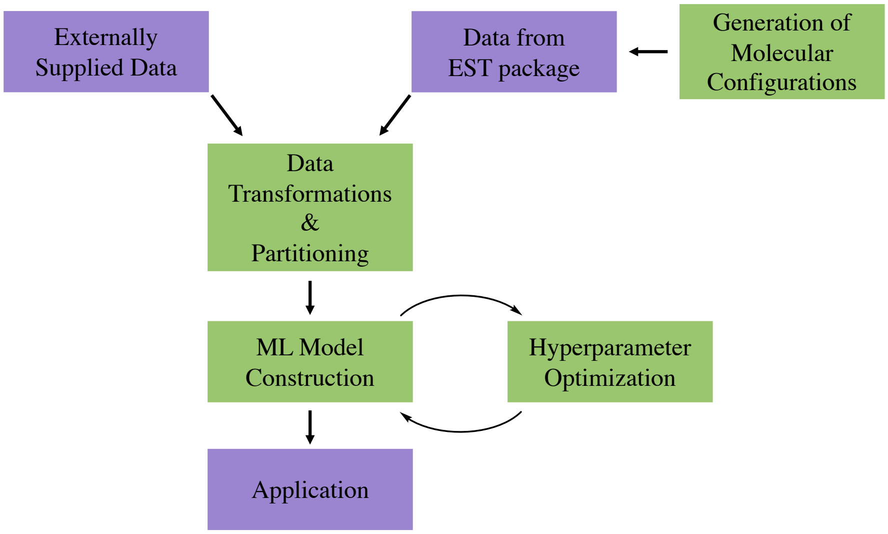
2 Capabilities

PES-Learn automates many of the various steps that are needed to go from a desired molecular system to an accurate ML model representation of the PES. These steps include:

- The generation of molecular geometry configurations and corresponding electronic energies
- Preprocessing and transforming the data in an optimal way
- Dividing the data into training and testing sets in an ideal manner
- Training a finely tuned ML model and exporting it in an easy-to-use form

Further details are discussed below. PES-Learn does not require programming knowledge to use, and features a familiar input file submission format through a command line interface. However, the modular structure gives rise to a natural application-programming interface (API) which enables more sophisticated workflows for users with minimal knowledge of the Python programming language. All of these features are accessible in either manner.

Figure 1: Workflow schematic for PES-Learn


Externally Supplied Data
Data from EST package
Generation of Molecular Configurations
Data Transformations & Partitioning
ML Model Construction
Hyperparameter Optimization
Application

2.1 Data Generation

In order to model a PES, one requires a grid of known geometry/energy pairs across the surface. This can be rather tedious to produce, but the present software makes it very easy to do. PES-Learn can automatically generate input files for an arbitrary electronic structure theory software package, as well as parse energies from the corresponding output files. The generation of input files is done by supplying a template input file for some EST package, named `template.dat`, which may contain anything, as long as the molecular coordinates are specified with Cartesian coordinates. Then, in the PES-Learn input file, standard "Z-matrix" internal coordinates are specified along with the corresponding range of values for each internal coordinate (e.g. r1 = [*start*, *stop*, *number of points*], etc.). Upon execution, the software will create a series of input files by replacing the Cartesian coordinates in template.dat with new Cartesian coordinates given by the internal coordinate grid. If desired, symmetrically redundant geometries resulting from the internal coordinate ranges, which are quite common, will be removed automatically, ensuring that only symmetry-unique points are computed. This greatly reduces the computational cost of generating the PES. For high-dimensional cases, the software also features an algorithm for filtering geometries from the configuration space based on the Euclidean distances between geometries in the dataset. If, for example, the internal coordinate grid contains tens of thousands of points, the user can specify to instead generate an evenly spaced sub-grid which only contains, say, 1000 points. This is done using a scheme similar to the "structure-based sampling" method described by Dral and coworkers,[23] and is especially useful for high-dimensional and low symmetry systems.

The parsing of energies from output files can be done with two different schemes. The first utilizes cclib,[49] an open-source Python package which can parse data from popular EST packages. For cases which cclib does not support, a completely general scheme using Python regular expressions is also implemented. The cclib interface is easier to use, while utilizing the regular expression scheme is more involved but carries the advantage of being completely general. To reiterate, PES-Learn supports input file writing and the parsing of energies from output files for any arbitrary EST package, so long as such a package supports Cartesian molecular coordinates in the input file for geometry specification. Therefore, in principle no barriers due to incompatibility with other software packages exist, leaving the user free to generate PES data in whatever way she chooses.

A simple example workflow of data generation in PES-Learn is given below. In this case, template.dat is a Psi4[50] input file:

```
molecule h2o {
O  0.000 0.000 0.000
H  0.000 0.000 1.000
H  0.000 1.000 0.000
}
set basis cc-pVTZ
energy(‘scf’)
```

In the corresponding PES-Learn input file, we choose to remove redundant geometries, and parse the energies with cclib:

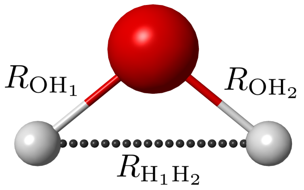
```
O
H 1 r1
H 1 r2 2 a1
r1 = [0.8, 1.5, 8]
r2 = [0.8, 1.5, 8]
a1 = [90, 180, 10]
remove_redundancy = true
energy = cclib
energy_cclib = scfenergies
```

Upon running PES-Learn, Cartesian coordinate input files corresponding to every unique combination of internal coordinates will be generated. The EST computations can then be run on the user's preferred computing resources. Once the EST computations are complete, the software can be run again to parse energies from the output files. The obtained dataset is then ready to be passed to the machine learning portion of the software. Though this data generation scheme is more than enough for many applications, not all use cases are intended to be satisfied by these capabilities. Fortunately, PES-Learn also supports externally supplied datasets in a variety of flexible formats.

2.2 Incorporation of Molecular Symmetry

The energy output of an ideal PES representation should be invariant with respect to geometry-preserving operations on a particular molecular configuration, such as translations, rotations, and the permutation of identical atoms. Therefore, such symmetries need to be built-in to a ML model of a PES. If desired, PES-Learn incorporates these symmetries into the ML models automatically.

Translational and rotational symmetry is taken care of by the use of internal coordinates, such as a standard "Z-matrix" or the set of interatomic distances. The remaining challenge is to incorporate permutational symmetry. Typically, this is done by using a set of permutation-invariant polynomials (PIPs) of the interatomic distances.[12–14,21,43] In the context of ML, the quantities in the input vector of the model are the evaluations of the PIPs instead of the raw molecular coordinates, and thus the permutation invariance is built-in. For small molecules, it is not difficult to devise a set of polynomials that satisfy the desired permutational symmetry. For example, the interatomic distances of water $R_{\mathrm{H_1H_2}}$, $R_{\mathrm{OH_1}}$, $R_{\mathrm{OH_2}}$ may be replaced with the polynomials $R_{\mathrm{H_1H_2}}$, $R_{\mathrm{OH_1}} \cdot R_{\mathrm{OH_2}}$, $R_{\mathrm{OH_1}} + R_{\mathrm{OH_2}}$, which are invariant under the permutation operation of the identical hydrogen atoms. In the context of ML, one would alter the input vector of the model in terms of the interatomic distances to instead be in terms of the PIPs:



$$\rightarrow \begin{bmatrix} R_{\mathrm{H_1H_2}} \\ R_{\mathrm{OH_1}} \\ R_{\mathrm{OH_2}} \end{bmatrix} \rightarrow \begin{bmatrix} R_{\mathrm{H_1H_2}} \\ R_{\mathrm{OH_1}} \cdot R_{\mathrm{OH_2}} \\ R_{\mathrm{OH_1}} + R_{\mathrm{OH_2}} \end{bmatrix}$$

More general approaches have been established for generating permutation-invariant geometry representations,[11,13] such as the monomial symmetrization method,[14,51] and methods motivated by theorems from invariant theory.[52] Such methods involve considering the effect of like-atom permutations on the set of interatomic distances, and then subsequently deriving the induced permutation operations on the interatomic distances. Then, armed with the set of interatomic distance variables and permutation operations on those variables, one can generally derive polynomials which are invariant with respect to those permutation operations. The derivation of invariant polynomials can be done automatically using computer algebra software such as MAGMA[53] or SINGULAR.[54] One example of such polynomials is the so-called *primary* and *secondary* invariant polynomials,[11–13] which together generate the invariant polynomial ring for the given set of interatomic distance variables and the permutation operations on those variables. More recently, the use of primary and secondary invariants have fallen out of favor compared to the *fundamental invariants* (FI), which can be derived with the same information but are the *minimal* generating set of the invariant polynomial ring.[22,29,55,56] For example, completely describing the permutational symmetry of an A_3B molecular system (e.g., ammonia) requires 14 primary and secondary invariants, but only 9 fundamental invariants. Since the cost of training and accuracy of ML models scale unfavorably with the size of the input vector, and the number of polynomials is equal to the size of the input vector, the FIs are clearly preferred. With larger molecules the advantage is even more apparent: an A_4B molecular system (e.g. methane) has 307 primary and secondary invariants, and only 31 fundamental invariants.

Therefore, PES-Learn utilizes the FIs as they are the most complete, general, and compact method presented thus far for incorporating permutational symmetry into PES models. PES-Learn features a library of fundamental invariants for various molecular systems which were computed with SINGULAR[54] using King's algorithm.[55] The SINGULAR input files are automatically generated with routines within PES-Learn, so additional molecular system types can easily be added as needed. However, we have found that the cost of computing the FIs for molecular systems with many like-atoms (> 4) is prohibitive. Not only that, but the number of FIs become too numerous for such systems, so much so that it is impractical to use them at all. In these cases it is more favorable to use a Behler–Parrinello style approach,[16] which makes use of atom-centered symmetry functions and represents the PES as a sum of atomic contributions.

When training machine learning models with PES-Learn, if the use of fundamental invariants is enabled, the geometries in the dataset are automatically transformed into the FI representation without any further input from the user. When a model is exported for later use in the form of a simple Python function accepting coordinate arguments (discussed later), the appropriate transformation of the coordinates into the PIP representation is performed on-the-fly.

2.3 Hyperparameter Optimization

The term *hyperparameter* has a rather loose definition, but here we use it to describe any attribute of the ML model or data transformation that is chosen *before* training a ML model. Some examples of hyperparameters include the number of layers in a NN, the choice of kernel for a GP, or the scaling of input and/or output data. Note the distinction of these hyperparameters compared to the model parameters, which are optimized during training.

The choice of hyperparameters can vary based on the application and type of ML model, but it is often difficult to know *a priori* whether a given hyperparameter is the proper choice. Indeed, model performance can vary wildly if just one hyperparameter is changed. A straightforward grid search of all possible hyperparameter settings can be done, but oftentimes the size of the hyperparameter space (the number of combinations of hyperparameters) is infeasibly large. Therefore, much effort has been dedicated towards creating hyperparameter optimization schemes based on algorithms which attempt to find correlations between hyperparameter settings. PES-Learn currently implements the very popular HyperOpt package for handling hyperparameter optimization,[57] with both the tree of parzen estimators[58] (TPE) and random search algorithms. Given a manageable hyperparameter space and enough iterations, HyperOpt reliably finds a combination of hyperparameters which minimize the prediction error of the ML model. The inclusion of this feature greatly reduces the need for user trial and error and enables users with limited knowledge of ML methods to obtain high quality models of PESs.

PES-Learn's Python API allows for setting custom hyperparameter configurations before running a hyperparameter optimization. This is useful if, for example, the user observes that a particular hyperparameter is well-suited for training a model on a particular dataset.

2.4 Dataset Sampling Algorithms

Given a dataset of points along a PES, it must be divided into a *training set* and a *test set* before passing it to a ML algorithm. The training set is the data that is seen by the model during training and used to optimize its parameters. The test set is used to verify the model's performance once it is finished training. Setting aside a test set is critical, as we desire the machine learning model to generalize to unseen datapoints. In order to truly gauge the model's performance, an ideal test set must be suitably large, diverse, and contain no datapoints which are identical or very similar to those in the training set. We therefore require methods of adequately sampling training and testing sets from a larger dataset of points. Random sampling is commonly used, but for modeling a PES this may lead to "holes" in the surface which are not well-described by the training set. To this end, in addition to random sampling, PES-Learn implements a variety of methods for splitting datasets into training and test sets:

- *smart random* sampling: selects a random seed which leads to a training set and test set energy distribution which most closely match that of the original dataset.

- *structure-based* sampling: selects a training set by analyzing the Euclidean distances between all of the geometries in the dataset, and choosing an evenly-spaced sub-grid of points for the training set, as described in reference 23.

2.5 Machine Learning Algorithms

Currently, PES-Learn supports the creation of Gaussian process and feed-forward neural network models. The theory and structure of these ML models is covered elsewhere,[59,60] and their application contextualized to molecular PESs is already detailed in the literature,[19,26] so we do not discuss the methodologies here. The Gaussian process models are created through an interface to the GPy open-source Python library.[61] The neural networks are built using PyTorch.[62]

The Gaussian process models perform well with a small number of training points and tend to not suffer from overfitting, but they scale rather unfavorably with the size of the training set, as noted previously.[26,28] Neural networks, on the other hand, scale well with increasing training set sizes, but they are more difficult to train, as they employ more parameters and hyperparameters, and often suffer from overfitting. Thus, the GP and NN models built by PES-Learn complement each other quite well with differing strengths and weaknesses, and should more than suffice for many use-cases. Future research may also pursue the implementation of atomistic neural network methods, discussed earlier. This will most likely be achieved through an interface to other open-source projects which have implemented these schemes.[38–41]

Once PES-Learn has finished training a model, it is saved as a file which can be readily loaded and utilized. PES-Learn also writes a simple Python file containing a convenience function for evaluating the PES ML model, which takes as arguments one or more sets of molecular coordinates and returns a corresponding number of electronic energies. The convenience function automatically handles all of the data transformations which that particular model was trained on, as well as the loading and execution of the ML model. Thus, the machine learning models created by PES-Learn can be easily used to predict energies for new molecular geometries with little to no programming experience.

3 Example Applications

In each of the following example applications, the molecular geometries were automatically transformed into a permutation-invariant representation using the present software's library of fundamental invariants (see section 2.2). Furthermore, each ML model for each example application presented here was optimized by the software in an entirely automated fashion. That is, no manual hand-tuning was performed for any of the models; each of them were constructed from the ground up by the same model optimization routine in PES-Learn.

3.1 Fitting Performance

We claim that the sophisticated nonlinear regression capabilities offered by modern machine learning techniques provide superior fits to molecular PESs compared to traditional methods, such as linear least-squares polynomial fitting. We also suggest that the creation of optimally-tuned ML models is well-automated by PES-Learn's hyperparameter optimization routines. Here we directly test these claims by fitting select potential energy surfaces from the literature and comparing the fitting error to previously published models. The diverse set of surfaces were featured in a recently published work[26] comparing GP regression to the linear least-squares permutation-invariant-polynomial (PIP-LS) approach.[11,13] The first surface is that of H_3O^+, originally published by Yu and coworkers.[63] The dataset as used in reference 26 contains 32,141 points, covers an energy range of ∼60 kcal/mol (∼21,000 cm^{-1}), and describes the $\mathrm{D_{3h}}$ saddle point separating the two equivalent $\mathrm{C_{3v}}$ global minima. The second surface is of the linear molecule OCHCO^+, and describes the proton-transfer between two equivalent $\mathrm{C_{\infty v}}$ minima ($\mathrm{OCH}\cdots\mathrm{CO}^+$, $\mathrm{OC}\cdots\mathrm{HCO}^+$) through a $\mathrm{D_{\infty h}}$ saddle point ($\mathrm{OC}\cdots\mathrm{H}\cdots\mathrm{CO}^+$).[64] This dataset contains 7,800 points spanning an energy range of ∼22,000 cm^{-1}.[26] The third surface describes the isomerization of formaldehyde (H_2CO) to *cis* and *trans* hydroxycarbene.[65] It contains three distinct minima and two 1^{st}-order saddle points connecting them. This H_2CO dataset as used previously[26] contains 34,750 points spanning ∼50,000 cm^{-1}.

Table 1 compares the previous attempts at fitting these PESs (the PIP-LS and GP methods[26]) to the optimized GP and NN models produced by PES-Learn. Each PES-Learn GP and NN was obtained by selecting the best-performing hyperparameter configuration after 20 hyperparameter optimization iterations. Fundamental invariant polynomials were used. Columns labeled RS (reference sample) use the exact same training and testing sets as described in reference 26 for each molecular species. As the fitting error for these columns corresponds to the exact same test set, these RMSE quantities directly compare the different fitting methods. The performance of training and test set sampling methods incorporated into PES-Learn are also included.

The most striking difference in RMSE can be seen for the case of H_3O^+ using 500 training points. The previously published PIP-LS and GP models yield errors of 116.99 and 268.74 cm^{-1}, respectively, while the PES-Learn optimized GP obtained an error of 19.97 cm^{-1} with the exact same training set. How could the same type of machine learning algorithm (Gaussian process regression) display such a large performance discrepancy? The robust hyperparameter optimization routines in PES-Learn search across not only different model configurations but also different transformations of the geometry parameters and energy. In the case of H_3O^+, PES-Learn found a hyperparameter configuration that performs exceptionally better than the GP model trained in the previous study.[26] With only 1,000 training points, the PES-Learn GP using structure-based (SB) sampling obtains an error of 6.09 cm^{-1}, beating the performance of the PIP-LS model trained on all 32,141 points (7.09 cm^{-1}). This constitutes a 97% reduction in the number of *ab initio* points required to obtain a given accuracy.

For $OCHCO^+$, the performance of the GP[26] and PES-Learn GP are similar, though the PES-Learn GP consistently has a slight edge. Generally, for smaller training set sizes of the H_3O^+ and $OCHCO^+$ surfaces, the PES-Learn neural networks perform slightly worse compared to the PES-Learn Gaussian process models, but the NNs become more competitive with increasing training set size. The PIP-LS models, on the other hand, fail to model the H_3O^+ and $OCHCO^+$ surfaces as effectively given the same number of points as the PES-Learn GP and NN models.

For the H_2CO surface, many more training points are required to describe the complex landscape of multiple minima and transition state structures. Here the PES-Learn neural networks outperform all other approaches by a wide margin. Using the reference sample training set of 5,104 points, both of the PES-Learn models achieve much better generalization across the surface than the PIP-LS model trained with 17,383 points.

There is a slight increase in fitting error between the PES-Learn NN models trained on the reference sample of 5,104 and 8,703 points (118.53 cm^{-1} and 121.07 cm^{-1}, respectively). We attribute this increase to the fact that the 5,104 training points are not contained within the 8,703 training point set. In fact, the 5,104 and 8,703 training sets only have 1,281 points in common. As such, a performance increase is not guaranteed, so the slight increase in fitting error is not suprising. The structure-based sampling algorithm samples the training set in a deterministic and systematic fashion, such that the smaller training set sizes are completely contained within the larger training set size. Because of this, we see a clear trend toward higher and higher accuracy as the training set size is increased.

In a very recent publication,[66] Brorsen fit these same $OCHCO^+$ and H_2CO surfaces with the SchNet architecture,[32] an atomistic NN approach. Training with 7,020 points from the $OCHCO^+$ surface, Brorsen obtained a test set RMSE of 23.55 cm^{-1}.[66] As observed from Table 1, PES-Learn obtains similar errors with just 520 training points with a variety of point sampling methods, and approaches sub-wavenumber accuracy with just 1,560 training points. For the H_2CO surface, Brorsen obtained a test set RMSE of 109.28 cm^{-1} with 31,275 training points using the SchNet architecture. PES-Learn, on the other hand, was able to construct better models (test set errors of 78.95, 82.96, 93.27 cm^{-1}) with a training set nearly half the size (17,383). For lack of a more comprehensive benchmark, this seems to suggest that PES-Learn models excel at fitting surfaces for small molecular systems and training set sizes. Meanwhile, atomistic methods such as the SchNet architecture can yield phenomenal performance for larger molecular systems, as demonstrated previously.[32,39]

We note that each result in Table 1 obtained in this work could be improved by increasing the number of hyperparameter optimization iterations. Since the results were obtained with just 20 hyperparameter optimization iterations, not all model configurations have been exhausted. We also note that though the results obtained by PES-Learn's automated model optimization routines are very good, an experienced machine learning practitioner would likely be able to achieve similar or even slightly better performance, given enough time. However, here we show that by using the present software, very impressive results can be obtained by a novice with minimal knowledge of ML methods.

Every result in Table 1 was obtained by running PES-Learn in parallel across 4 cores on an Intel Xeon E3-1270 v5 @ 3.60 GHz CPU. Each GP and NN result was obtained by running the software for less than 8 hours, with the exception of the 5104-point training sets used for the GP's, which required about one day. Such large training sets are really not appropriate for standard GP regression, as discussed earlier and noted previously.[26] The amount of time required to produce optimal models with smaller training set sizes (< 2000) required roughly 30 minutes or less. While these timings are much slower than linear fitting methods, the cost of training these ML models pales in comparison to the cost of generating the *ab initio* data in the first place. Given that the required number of *ab initio* computations for a given accuracy is considerably reduced, we contend that PES-Learn's ML model optimization is well worth the wait.

3.2 Vibrational Configuration Interaction Computations

Here we demonstrate the performance of the auto-generated machine learning models produced by PES-Learn through vibrational configuration interaction (VCI) computations.[67] As vibrational energy levels obtained through VCI are highly sensitive to the accuracy of energy predictions from the underlying PES, on the order of mere wavenumbers, VCI serves as an excellent test application. Rather than the brute-force computation of tens of thousands of *ab initio* points as is typically done in the literature, we leverage the flexible and robust fitting capabilities of machine learning models to minimize the number of *ab initio* computations needed. We attempt to obtain high-quality VCI results with a machine learning model trained from only a few hundred points for both H_2O and H_2CO. The capabilities of PES-Learn are utilized to greatly simplify the workflow of obtaining the present results.

Our VCI computations are performed with Changala's NITROGEN program.[68] We used NITROGEN's general curvilinear internal coordinate vibrational Hamiltonian.[69,70] The Hamiltonian was approximated with a many-body (or n-mode) expansion in normal coor-

Table 1: Test set fitting error (RMSE, cm^{-1}) of the H_3O^+, $OCHCO^+$, and H_2CO PESs using four different fitting methods: linear least-squares fitting of permutation invariant polynomials (PIP-LS), Gaussian process model (GP) reported previously, and PES-Learn auto-optimized Gaussian process (PES-Learn GP) and neural network (PES-Learn NN) models. Errors are given for different training and testing set sampling schemes. Reference sample (RS) is the exact same training set and test set sample used in reference 26. The performance of PES-Learn sampling algorithms, structure-based (SB) sampling and smart-random (SR) sampling, are also included for comparison.

		PIP-LS[a]	GP[a]	PES-Learn GP[b]			PES-Learn NN[b]		
		RS	RS	RS	SB	SR	RS	SB	SR
H_3O^+	N_{train}								
	500	116.99	238.74	19.97	15.24	22.59	22.53	15.27	34.76
	1000	39.20	125.76	10.71	6.09	10.22	13.99	11.77	27.78
	2000	28.13	36.33	5.88	2.94	6.36	11.35	6.99	15.31
	5000	11.81	11.17						
	10000	9.99	5.62						
	32141	7.09							
$OCHCO^+$	N_{train}								
	520	259.18	32.09	23.77	8.70	19.62	38.84	14.33	46.79
	780	293.53	26.62	23.03	4.06	17.87	29.60	5.92	34.96
	1560	128.65	15.78	10.58	1.03	8.03	11.99	2.33	10.30
	2600	97.40	15.55						
H_2CO	N_{train}								
	5104	432	238	191.15	402.84	375.27	118.53	124.09	189.14
	8703	322	229				121.07	91.36	146.29
	17383	239	143				78.95	82.96	93.27

[a] From Ref 26.

[b] This work.

dinates of the potential energy and kinetic coupling terms, which is a standard approximation.[69,71,72] We used the 3-mode expansion in each case. For water, this Hamiltonian is exact, while for formaldehyde, the 3-mode truncation of the Hamiltonian is an approximation to the exact 6-mode expansion. A discrete variable representation[73] (DVR) of the potential was constructed with 20 harmonic oscillator basis functions along each normal coordinate. A ground state vibrational self-consistent field (VSCF) reference wavefunction was obtained, and virtual configurations involving up to 15 total quanta of excitation were included in the VCI Hamiltonian matrix for water. For formaldahyde, due to prohibitive computational expense, configurations with up to 10 total quanta of excitation were included. Diagonalizing the VCI Hamiltonian matrix yields the $J = 0$ vibrational energy levels.

3.2.1 H_2O VCI

For water, we first computed the VCI results with density-fitted Møller-Plesset perturbation theory (DF-MP2) with a cc-pVDZ-RI basis set.[74] In the first case, the electronic energies required by the VCI procedure were analytically evaluated with the Psi4 quantum chemistry package.[50] Next, we performed VCI using energies obtained from a machine learning model generated with PES-Learn which was trained on DF-MP2/cc-pVDZ-RI energies. The ML model was constructed as follows. First, the data generation module of PES-Learn was used to construct a grid of internal coordinates spanning 11 points from 0.8 to 1.3 Å for each of the bonds and 15 points from 40° to 180 ° for the HOH angle. The equilibrium geometry was also included in the grid, and redundant geometries were removed. In total, 991 geometries were obtained, and the corresponding Psi4 input files were generated. The energies were computed with Psi4 and parsed by PES-Learn. The obtained dataset was used to train a Gaussian process model using 300 points in the training set which were chosen using the structure-based sampling algorithm. Twenty hyperparameter combinations were attempted, and the best model was selected. The hyperparameter optimization iterations in total took a few minutes on a modern desktop computer with a 4-core Intel Core i7-7700 CPU 3.60GHz.

For the best model, the root-mean-square error (RMSE) of the test set (the 691 points not used for training) was 10.8 cm^{-1}, and the energy range of the entire 991 point dataset was roughly 45,000 cm^{-1}. We performed a similar procedure at the CCSD(T)/aug-cc-pVTZ level of theory, comparing the VCI results with energies obtained both analytically and with a Gaussian process model trained on 300 points. Finally, using the exact same grid of points as the DF-MP2 and CCSD(T) cases (with the exception of the equilibrium geometry), we computed CCSDT(Q) energies at the complete basis set (CBS) limit with several additive corrections.

The CCSDT(Q)/CBS energies were obtained by a procedure analogous to the focal-point approach of Allen and coworkers[75–77] using the augmented correlation consistent basis sets of Dunning and coworkers (aug-cc-pVXZ).[78] The augmented functions were found to be necessary for adequately describing geometries heavily distorted from the equilibrium geometry. In addition to the energy extrapolation[79–81] we include a core correlation correction, a scalar relativistic correction (MVD1),[82] and a diagonal Born–Oppenheimer correction (DBOC).[83,84] Full details for how these were obtained are given in the supporting information. The resulting dataset consisted of a grid of geometries and their corresponding composite energies, each formed from a sum of the CCSDT(Q)/CBS electronic energy and the core, relativistic, and DBOC corrections. 300 points of this dataset were used to train a GP model of the PES. The RMSE of the test set was 9.26 cm^{-1}, and the RMSE of the full dataset was 7.81 cm^{-1}.

Table 2 compares the vibrational energy levels of water obtained from VCI using various methods. We denote the vibrational levels in terms of the normal modes of vibration: the symmetric stretch (ν_1), the bend (ν_2), and the asymmetric stretch (ν_3). The Gaussian process models of the PESs at DF-MP2/cc-pVDZ-RI and CCSD(T)/aug-cc-pVTZ reproduce each of the vibrational levels obtained with the corresponding analytic energies to well within 1 cm^{-1}. Likewise, a ML model of the composite CCSDT(Q)/CBS PES reproduces the experimental vibrational bands with exceptional agreement. The largest error in the first 15 vibrational levels with respect to experiment is only 1.06 cm^{-1}. The first 40 vibrational levels are presented in the supporting information, and the accuracy with respect to experiment does not degrade significantly for higher vibrational states. Of these 40 vibrational levels, the vast majority are within 2 cm^{-1} of the experimental values, with just a few exceptions; in particular, states with very high (> 6) quanta of excitation in the bending vibration ν_2 feature notably less experimental agreement. The mean-absolute error (MAE) of these 40 vibrational levels with respect to the experimental values is 2.16 cm^{-1}. Excluding the largest error, 34.21 cm^{-1} for the $8\nu_2$ level, the MAE of the remaining levels is just 1.34 cm^{-1}. It is likely that the vibrational levels with poor experimental agreement are negatively affected by the lack of coverage of the internal coordinate grid over the bending vibrational mode ν_2 normal coordinate. If our dataset were generated with better coverage over this coordinate, one would expect the vibrational states with highly excited bending modes to be better predicted. We also note here that ML models trained with 200 and 250 training points yielded slightly higher errors for some modes (on the order of 1–2 cm^{-1}), while models trained with 400 and 500 points did not improve over the 300 training point model significantly.

The accuracy of the vibrational levels are in good agreement with past high-level *ab initio* computations.[85–87] Admittedly, the usage of ML in this case does not significantly reduce the computational cost compared to linear fitting methods in regard to the number of *ab initio* computations required. For example, the CVRQD surface[85,86] consisted of a sum of seven surfaces, most of which were each generated with energies from roughly 300–400 geometries. This is improved here only slightly with the need for only 300 training points to reproduce results obtained from analytic energies. The advantage of ML over linear fitting methods is more clear for higher dimensional surfaces, as we will see in the next example. Nevertheless, this example application illustrates how the capabilities of PES-Learn facilitate the construction and application ML models of molecular PESs toward high-level *ab initio* results.

Table 2: Errors in VCI $J = 0$ vibrational energy levels of H_2O using an ML model of the PES.[a] The DF-MP2 and CCSD(T) deviations ($\Delta_{\text{Analytic–ML}}$) correspond to the difference between the vibrational energy levels obtained using exact *ab initio* energies versus those obtained with an ML potential. The final column represents the difference between experimental values[b] and values obtained from VCI using the CCSDT(Q) ML potential.[c] All ML models trained with 300 *ab initio* points.[d]

	DF-MP2	CCSD(T)	Experiment[b]	CCSDT(Q)	Diff.
State	$\Delta_{\text{Analytic–ML}}$	$\Delta_{\text{Analytic–ML}}$			
ν_2	0.03	0.14	1594.75	1594.62	0.13
$2\nu_2$	0.31	0.02	3151.63	3151.55	0.08
ν_1	0.06	−0.23	3657.05	3656.94	0.11
ν_3	−0.40	−0.19	3755.93	3756.53	−0.60
$3\nu_2$	0.33	−0.21	4666.79	4666.75	0.04
$\nu_1 + \nu_2$	−0.19	0.01	5234.98	5234.88	0.10
$\nu_2 + \nu_3$	−0.45	−0.12	5331.27	5331.65	−0.38
$4\nu_2$	0.11	−0.37	6134.02	6134.13	−0.11
$\nu_1 + 2\nu_2$	0.11	−0.06	6775.09	6775.09	0.00
$2\nu_2 + \nu_3$	−0.20	−0.23	6871.52	6871.90	−0.38
$2\nu_1$	−0.11	−0.32	7201.54	7201.80	−0.26
$\nu_1 + \nu_3$	−0.35	−0.36	7249.82	7250.85	−1.03
$2\nu_3$	−0.43	−0.25	7445.06	7446.12	−1.06
$5\nu_2$	−0.20	−0.39	7542.37	7543.40	−1.03
$\nu_1 + 3\nu_2$	0.02	−0.33	8273.98	8273.99	−0.01
				MAE first 40 levels	2.16

[a] Higher vibrational levels included in the supporting information.

[b] Experimental values obtained from the analysis in reference 88.

[c] CCSDT(Q)/CBS energies included additive core correlation, relativistic, and DBOC corrections. See text.

[d] The same 300 geometries were used for the training set for each ML model, with the exception of the equilibrium geometry.

3.2.2 H_2CO VCI

We perform VCI computations on formaldehyde using a similar procedure as discussed earlier for water. We first probed possible internal coordinate grids by comparing VCI results using analytic DF-MP2/cc-pVDZ-RI energies to VCI results obtained using energies produced by a ML model of the PES. As before, we quite easily found an internal coordinate grid which gave rise to a high quality ML model for obtaining sub-wavenumber agreement in the VCI vibrational energy levels compared to the levels obtained using analytic DF-MP2 energies (see supporting information). The final dataset was composed of points selected from two internal coordinate grids. The first internal coordinate grid consisted of 500 points selected using structure-based sampling from a larger grid of 8,126 symmetry-unique points. This grid spanned 5 increments across each of the following coordinates: the C–O bond distance from 1.16 to 1.27 Å, the H–C bond distances from 1.05 to 1.20 Å, the HCO angles from 105 to 135 degrees, and the dihedral angle from 160 to 180 degrees. A second grid spanning a more expansive configuration space was also generated by selecting 500 points using structure-based sampling from a larger grid of 23,977 symmetry-unique points. This grid spanned 6 increments across each of the following coordinates: the C–O bond distance from 1.05 to 1.45 Å, the H–C bond distances from 0.90 to 1.45 Å, the HCO angles from 95 to 145 degrees, and the dihedral angle from 110 to 180 degrees. The resulting dataset consisted of 1,000 distinct geometry configurations, and it was produced entirely by using PES-Learn data generation capabilities.

At each geometry, we computed a CCSD(T)/CBS energy as follows. A three-point scheme was used for extrapolating the Hartree–Fock energies (aug-cc-pVTZ, aug-cc-pVQZ, and aug-cc-pV5Z),[79,80] and the correlation energies at CCSD(T)/aug-cc-pVQZ and CCSD(T)/aug-cc-pV5Z were extrapolated using a two-point CBS extrapolation.[81] Each energy included relativistic effects using the spin-free exact two-component theory in its one electron variant[89] (SF-X2C-1e) as implemented in the CFOUR package.[90] The basis sets used were the SF-X2C-1e recontracted variants.[90] The frozen-core approximation was used.

Table 3 compares the first 25 vibrational energy levels of formaldehyde obtained using various methods. We denote the vibrational levels in terms of the following normal modes of vibration: symmetric C–H stretch (ν_1), C–O stretch (ν_2), symmetric OCH bend (ν_3), torsion (ν_4), antisymmetric C–H stretch (ν_5), and antisymmetric OCH bend (ν_6). The first column of deviations with respect to experimental results was obtained in the present work using an underlying GP PES trained using 800 *ab initio* points from the dataset of 1000 points computed at the CCSD(T)/CBS level of theory. The RMSE of the test set (the 200 points not used for training) was 1.76 cm^{-1}, while the RMSE of the full 1000-point dataset was just 1.16 cm^{-1}. The vast majority of the points lie within 44,000 cm^{-1} above the global minimum, so these errors indicate that our GP is an excellent model of not only the local PES around the equilibrium geometry but also for higher energy regions.

We first note that for the present CCSD(T)/CBS results, all of the fundamentals lie within 2 cm^{-1} of the experimental values, as do most of the overtones and combination bands. The largest error in the first 25 vibrational energy levels is $2\nu_2$ at -4.28 cm^{-1}. The excellent experimental agreement continues for the higher vibrational states, which are not included in Table 3, but are included in the supporting information. The mean absolute error (MAE) of the first 50 vibrational levels is 1.68 cm^{-1}, and the MAE for the first 125 vibrational levels is 2.52 cm^{-1}.

For the fundamentals, the error is directly correlated with the level of theory utilized. The previously reported CCSDT(Q)/CBS results perform the best for the few low-lying modes that were computed, as is expected.[91] We generally also see that the size of basis for the CCSD(T) results plays an important role in accurately describing the vibrational energy levels. As one might anticipate, the previously reported VCI results with triple-ζ quality basis sets[92,93] are outperformed by the present CBS limit result. The VCI CCSD(T)/CBS result also demonstrates significant improvement over values obtained by the TROVE CCSD(T)/aug-cc-pVQZ result,[94] though the TROVE approach is different than VCI so these are not as directly comparable.

It is important to put the 800 training points required to obtain the GP CCSD(T)/CBS VCI result into perspective. First, we point out that it is highly likely that more sophisticated sampling schemes would reduce the number of points needed, such as sampling along the normal coordinates. Nevertheless, in light of what is typically done for PESs intended for computing the energies of high-lying vibrational states, i.e., tens of thousands of *ab initio* computations, the amount of computational savings obtained by using ML for the PES is tremendous. The TROVE result[94] in Table 3, for example, was obtained using a PES obtained from a least-squares fit of 30,840 single point energies, while the present GP PES was fit with just 800 single point energies. This massive reduction in the grid size needed to fit the PES enabled us to compute our result at a very high level of theory [CCSD(T)/CBS] which would be otherwise infeasible.

The cost of obtaining the GP representation of the PES is even competitive with the cost of more approximate methods of obtaining anharmonic fundamental vibrational energies, which only consider the higher order derivatives of the equilibrium geometry. For instance, the number of *ab initio* energy evaluations needed for training the GP model for the present VCI result is considerably less than that needed for a full quartic force field (QFF) derived from finite differences in normal coordinates. For H_2CO this requires ∼1,500 single-point energies when exploiting its C_{2v} point group symmetry, though this can be reduced by other means, such as using a symmetry-adapted coordinate system.[91,92] However, QFF approaches are not precise for vibrational states far beyond the fundamentals, as they only describe the PES at near-equilibrium geometries. The present GP PES obtained with 800 single-point energies describes the geometrical configurations well above the equilibrium region, and as such is appropriate for a VCI treatment of high-lying vibrational states. Though the CCSD(T)/CBS high-lying vibrational energy levels are overall the best pure *ab initio* results presented in the literature thus far, the underlying potential lacks contributions which may improve the results even further. Here we are neglecting contributions from core electron correlation, as well as correlation arising from higher order excitations and corrections for the Born–Oppenheimer approximation.

Table 3: Deviation from experimental values of *ab initio* results for the $J = 0$ vibrational levels of S_0 formaldehyde. The CCSD(T)/CBS result obtained in this work utilized potential energies from a PES-Learn ML model trained on 800 *ab initio* computations. All values given in wavenumbers (cm^{-1}).

	Experiment	This work	Previous theory			
		VCI[a]	TROVE[b]	VCI[c]	VCI[d]	DVR[e]
		CCSD(T)	CCSD(T)	CCSD(T)	CCSD(T)-F12a	CCSDT(Q)
State		CBS	aug-cc-pVQZ	cc-pVTZ	aug-cc-pVTZ	CBS
ν_4	1167.2563[f]	−0.56	1.16	1.56	0.06	0.80
ν_6	1249.0948[f]	0.86	3.49	1.19	3.29	−0.04
ν_3	1500.1747[f]	1.53	1.07	−4.13	2.47	0.02
ν_2	1746.0093[g]	−1.89	1.40	−2.99	−1.29	0.32
$2\nu_4$	2327.5239[g]	−0.22	2.34	3.02		3.05
$\nu_4 + \nu_6$	2422.9701[g]	0.49	4.54	2.67		0.32
$2\nu_6$	2494.3543[g]	1.82	6.62	1.95		−0.52
$\nu_3 + \nu_4$	2667.0481[g]	1.32	2.41	−1.65		0.17
$\nu_3 + \nu_6$	2719.1550[g]	1.83	3.92	−5.64		−1.31
ν_1	2782.4575[g]	1.98	0.72	−6.74	−1.74	−1.18
ν_5	2843.3256[g]	−0.46	0.96	1.33	−6.07	−1.06
$\nu_2 + \nu_4$	2905.9685[g]	−2.69	2.62	−0.53		1.24
$2\nu_3$	2998.9873[g]	2.78	3.08	−1.71		
$\nu_2 + \nu_6$	3000.0659[g]	−1.07	2.83	−5.73		0.17
$\nu_2 + \nu_3$	3238.4548[g]	−1.07	2.34	−6.35		0.14
$2\nu_2$	3471.6 [h]	−4.28	2.21	−9.60		0.53
$3\nu_4$	3480.7 [h]	−0.38	2.66			6.28
$2\nu_4 + \nu_6$	3586.6 [h]	0.58	5.51			
$\nu_4 + 2\nu_6$	3673.5 [h]	−1.54	4.62			
$\nu_3 + 2\nu_4$	3825.3 [h]	1.27	2.87			
$\nu_3 + \nu_4 + \nu_6$	3886.5 [h]	−1.09	3.13			
$\nu_3 + 2\nu_6$	3937.4 [h]	3.80	7.54			
$\nu_1 + \nu_4$	3941.5295[i]	0.06	1.68	11.30		
$\nu_4 + \nu_5$	3996.5180[i]	−0.89	1.77	10.60		
$\nu_1 + \nu_6$	4021.080 66[j]	1.39	3.38	10.78		
MAE first 50 levels		1.68				
MAE first 125 levels		2.52				

[a] This research. Additional energy levels included in the supporting info. Includes relativistic correction. See text.

[b] Ref. 94.

[c] Ref. 92.

[d] Ref. 93.

[e] Ref. 91. Includes core correlation correction.

[f] Ref. 95 Tchana et al.

[g] Ref. 96 Perrin et al.

[h] Ref. 97 Bouwens et al.

[i] Ref. 98 Perez et al.

[j] Ref. 99 Flaud et al.

3.3 Additional Applications

The use of PES-Learn accelerates the acquisition of high quality PESs and opens the door to previously infeasible problems. There are numerous potential applications not explored in this work which will benefit from the capabilities of the present software. For example, in an upcoming study produced by our group regarding the reaction of 2-methylvinoxy radical with O_2, we utilize PES-Learn to model the 2-dimensional reaction coordinate and elucidate the nature of the entrance channel of the surface. We also note that in principle, the present software should be appropriate for modeling other geometry-dependent properties (e.g. dipole moment), though this has not been investigated. Finally, more sophisticated ML methodologies open up further avenues of research. Examples include using gradients to inform the construction of the PESs, or the implementation of multi-fidelity Gaussian processes,[46] which exploit a low level of theory PES to inform the construction of a higher level of theory PES.

4 Conclusions

The burgeoning field of applied machine learning is being established as a valuable tool for theoretical chemistry. Here we have illustrated how our software package, PES-Learn, can provide superior fits of complex PESs compared to traditional methods. We have also shown that the present software greatly simplifies the application of ML methods toward creating PES representations for small molecules. The sample workflow used here for the VCI computations of water and formaldehyde is expected to be appropriate for other applications as well. Initially probing several PES grids with a low level of theory, as done here, is recommended for ensuring that produced ML models are energetically relevant to the application in question. We note that the open-source nature and modular structure of PES-Learn allows for community contributions. For example, additional ML model types and training set data sampling methods can be easily added to the code. It is our hope that PES-Learn lowers the barrier-to-entry not only to utilizing machine learning in quantum chemistry research but also towards ordinarily prohibitive PES applications in general. We would like to further develop the code to include additional features which enable more complex use-cases such as larger system sizes, and we welcome suggestions and contributions from the community. The software is open-source and freely available on GitHub (https://github.com/CCQC/PES-Learn).

5 Acknowledgments

We gratefully acknowledge the Molecular Sciences Software Institute[100,101] (MolSSI) and the U.S. National Science Foundation (NSF) for funding this research. ASA and BZ are supported by a fellowship from the Molecular Sciences Software Institute (MolSSI) under NSF grant ACI-1547580. ASA, JMT, and HFS are supported by the NSF, grant CHE-1661604. We also thank Dr. Joel Bowman and coworkers for providing the datasets used to prepare Table 1.

Supporting Information Available

See SI for additional details regarding the *ab initio* computations and higher energy vibrational levels for H_2O and H_2CO.

References

(1) Ischtwan, J.; Collins, M. A. Molecular potential energy surfaces by interpolation. *J. Chem. Phys.* **1994**, *100*, 8080–8088.

(2) Jordan, M. J.; Thompson, K. C.; Collins, M. A. The utility of higher order derivatives in constructing molecular potential energy surfaces by interpolation. *J. Chem. Phys.* **1995**, *103*, 9669–9675.

(3) Jordan, M. J.; Thompson, K. C.; Collins, M. A. Convergence of molecular potential energy surfaces by interpolation: Application to the OH + H_2 → H_2O + H reaction. *J. Chem. Phys.* **1995**, *102*, 5647–5657.

(4) Thompson, K. C.; Jordan, M. J.; Collins, M. A. Polyatomic molecular potential energy surfaces by interpolation in local internal coordinates. *J. Chem. Phys.* **1998**, *108*, 8302–8316.

(5) Maisuradze, G. G.; Thompson, D. L.; Wagner, A. F.; Minkoff, M. Interpolating moving least-squares methods for fitting potential energy surfaces: Detailed analysis of one-dimensional applications. *J. Chem. Phys.* **2003**, *119*, 10002–10014.

(6) Guo, Y.; Harding, L. B.; Wagner, A. F.; Minkoff, M.; Thompson, D. L. Interpolating moving least-squares methods for fitting potential energy surfaces: An application to the H_2CN unimolecular reaction. *J. Chem. Phys.* **2007**, *126*, 104105.

(7) Dawes, R.; Thompson, D. L.; Guo, Y.; Wagner, A. F.; Minkoff, M. Interpolating moving least-squares methods for fitting potential energy surfaces: Computing high-density potential energy surface data from low-density ab initio data points. *J. Chem. Phys.* **2007**, *126*, 184108.

(8) Quintas-Sánchez, E.; Dawes, R. AUTOSURF: A Freely Available Program To Construct Potential Energy Surfaces. *J. Chem. Inf. Model.* **2018**, *59*, 262–271.

(9) Jäckle, A.; Meyer, H.-D. Product representation of potential energy surfaces. *J. Chem. Phys.* **1996**, *104*, 7974–7984.

(10) Jäckle, A.; Meyer, H.-D. Product representation of potential energy surfaces. II. *J. Chem. Phys.* **1998**, *109*, 3772–3779.

(11) Qu, C.; Yu, Q.; Bowman, J. M. Permutationally invariant potential energy surfaces. *Annu. Rev. Phys. Chem.* **2018**, *69*, 151–175.

(12) Huang, X.; Braams, B. J.; Bowman, J. M. Ab initio potential energy and dipole moment surfaces for $H_5O_2^+$. *J. Chem. Phys.* **2005**, *122*, 044308.

(13) Braams, B. J.; Bowman, J. M. Permutationally invariant potential energy surfaces in high dimensionality. *Int. Rev. Phys. Chem.* **2009**, *28*, 577–606.

(14) Xie, Z.; Bowman, J. M. Permutationally invariant polynomial basis for molecular energy surface fitting via monomial symmetrization. *J. Chem. Theory Comput.* **2009**, *6*, 26–34.

(15) Varandas, A. Intermolecular and intramolecular potentials. *Adv. Chem. Phys* **1988**, *74*, 255–338.

(16) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(17) Handley, C. M.; Popelier, P. L. Potential energy surfaces fitted by artificial neural networks. *J. Phys. Chem. A* **2010**, *114*, 3371–3383.

(18) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.

(19) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(20) Behler, J. Representing potential energy surfaces by high-dimensional neural network potentials. *J. Phys. Condens. Matter* **2014**, *26*, 183001.

(21) Jiang, B.; Li, J.; Guo, H. Potential energy surfaces from high fidelity fitting of ab initio points: the permutation invariant polynomial-neural network approach. *Int. Rev. Phys. Chem.* **2016**, *35*, 479–506.

(22) Shao, K.; Chen, J.; Zhao, Z.; Zhang, D. H. Communication: Fitting potential energy surfaces with fundamental invariant neural network. *J. Chem. Phys.* **2016**, *145*, 071101.

(23) Dral, P. O.; Owens, A.; Yurchenko, S. N.; Thiel, W. Structure-based sampling and self-correcting machine learning for accurate calculations of potential energy surfaces and vibrational levels. *J. Chem. Phys.* **2017**, *146*, 244108.

(24) Cui, J.; Krems, R. V. Efficient non-parametric fitting of potential energy surfaces for polyatomic molecules with Gaussian processes. *J. Phys. B* **2016**, *49*, 224001.

(25) Kolb, B.; Marshall, P.; Zhao, B.; Jiang, B.; Guo, H. Representing global reactive potential energy surfaces using Gaussian processes. *J. Phys. Chem. A* **2017**, *121*, 2552–2557.

(26) Qu, C.; Yu, Q.; Van Hoozen, B. L.; Bowman, J. M.; Vargas-Hernandez, R. A. Assessing Gaussian process regression and permutationally invariant polynomial approaches to represent high-dimensional potential energy surfaces. *J. Chem. Theory Comput.* **2018**,

(27) Guan, Y.; Yang, S.; Zhang, D. H. Construction of reactive potential energy surfaces with Gaussian process regression: active data selection. *Mol. Phys.* **2018**, *116*, 823–834.

(28) Kamath, A.; Vargas-Hernández, R. A.; Krems, R. V.; Carrington Jr, T.; Manzhos, S. Neural networks vs Gaussian process regression for representing potential energy sur-

faces: A comparative study of fit quality and vibrational spectrum accuracy. *J. Chem. Phys.* **2018**, *148*, 241702.

(29) Li, J.; Song, K.; Behler, J. A critical comparison of neural network potentials for molecular reaction dynamics with exact permutation symmetry. *Phys. Chem. Chem. Phys.* **2019**,

(30) Gastegger, M.; Schwiedrzik, L.; Bittermann, M.; Berzsenyi, F.; Marquetand, P. wACSF–Weighted atom-centered symmetry functions as descriptors in machine learning potentials. *J. Chem. Phys.* **2018**, *148*, 241709.

(31) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **2017**, *8*, 13890.

(32) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet–A deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722.

(33) Fletcher, T. L.; Popelier, P. L. Multipolar electrostatic energy prediction for all 20 natural amino acids using Kriging machine learning. *J. Chem. Theory Comput.* **2016**, *12*, 2742–2751.

(34) Thacker, J. C.; Wilson, A. L.; Hughes, Z. E.; Burn, M. J.; Maxwell, P. I.; Popelier, P. L. Towards the simulation of biomolecules: optimisation of peptide-capped glycine using FFLUX. *Mol. Simulat.* **2018**, *44*, 881–890.

(35) Popelier, P. L. QCTFF: on the construction of a novel protein force field. *Int. J. Quantum Chem.* **2015**, *115*, 1005–1011.

(36) Handley, C. M.; Hawe, G. I.; Kell, D. B.; Popelier, P. L. Optimal construction of a fast and accurate polarisable water potential based on multipole moments trained by machine learning. *Physical Chemistry Chemical Physics* **2009**, *11*, 6365–6376.

(37) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, *8*, 3192–3203.

(38) Kolb, B.; Lentz, L. C.; Kolpak, A. M. Discovering charge density functionals and structure-property relationships with PROPhet: A general framework for coupling machine learning and first-principles methods. *Sci. Rep.* **2017**, *7*, 1192.

(39) Schütt, K.; Kessel, P.; Gastegger, M.; Nicoli, K.; Tkatchenko, A.; Müller, K.-R. SchNetPack: A deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* **2018**, *15*, 448–455.

(40) Deepchem: Democratizing deep-learning for drug discovery, quantum chemistry, materials science and biology. `https://github.com/deepchem/deepchem`, 2016.

(41) Khorshidi, A.; Peterson, A. A. Amp: A modular approach to machine learning in atomistic simulations. *Comput. Phys. Commun.* **2016**, *207*, 310–324.

(42) Le, H. M.; Huynh, S.; Raff, L. M. Molecular dissociation of hydrogen peroxide (HOOH) on a neural network ab initio potential surface with a new configuration sampling method involving gradient fitting. *J. Chem. Phys.* **2009**, *131*, 014107.

(43) Jiang, B.; Guo, H. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. *J. Chem. Phys.* **2013**, *139*, 054112.

(44) Hughes, Z. E.; Thacker, J. C.; Wilson, A. L.; Popelier, P. L. Description of Potential Energy Surfaces of Molecules Using FFLUX Machine Learning Models. *J. Chem. Theory Comput.* **2018**, *15*, 116–126.

(45) Christianen, A.; Karman, T.; Vargas-Hernández, R. A.; Groenenboom, G. C.; Krems, R. V. Six-dimensional potential energy surface for NaK–NaK collisions: Gaussian process representation with correct asymptotic form. *J. Chem. Phys.* **2019**, *150*, 064106.

(46) Wiens, A. E.; Copan, A. V.; Schaefer, H. F. Multi-Fidelity Gaussian process modeling for chemical energy surfaces. *Chem. Phys. Lett.* **2019**, 100022.

(47) Bartók, A. P.; Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. of Quantum Chem.* **2015**, *115*, 1051–1057.

(48) Owens, A.; Yurchenko, S. N.; Yachmenev, A.; Tennyson, J.; Thiel, W. Accurate ab initio vibrational energies of methyl chloride. *J. Chem. Phys.* **2015**, *142*, 244306.

(49) O'boyle, N. M.; Tenderholt, A. L.; Langner, K. M. cclib: a library for package-independent computational chemistry algorithms. *J. Comput. Chem.* **2008**, *29*, 839–845.

(50) Parrish, R. M.; Burns, L. A.; Smith, D. G.; Simmonett, A. C.; DePrince, A. E.; Hohenstein, E. G.; Bozkaya, U.; Sokolov, A. Y.; Di Remigio, R.; Richard, R. M.; Gonthier, J. F.; James, A. M.; McAlexander, H. R.; Kumar, A.; Saitow, M.; Wang, X.; Pritchard, B. P.; Verma, P.; Schaefer, H. F.; Patkowski, K.; King, R. A.; Valeev, E. F.; Evangelista, F. A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability. *J. Chem. Theory Comput.* **2017**, *13*, 3185–3197.

(51) Bowman, J. M.; Braams, B. J.; Carter, S.; Chen, C.; Czakó, G.; Fu, B.; Huang, X.; Kamarchik, E.; Sharma, A. R.; Shepler, B. C.; Wang, Y.; Xie, Z. Ab-initio-based potential energy surfaces for complex molecules and molecular complexes. *J. Phys. Chem. Lett.* **2010**, *1*, 1866–1874.

(52) Derksen, H.; Kemper, G. *Computational Invariant Theory*; Encyclopaedia of Mathematical Sciences; Springer Berlin Heidelberg, 2002.

(53) Bosma, W.; Cannon, J.; Playoust, C. The Magma algebra system. I. The user language. *J. Symb. Comput.* **1997**, *24*, 235–265.

(54) Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H. SINGULAR 4-1-1 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de, 2018.

(55) King, S. A. Minimal generating sets of non-modular invariant rings of finite groups. *J. Symb. Comput.* **2013**, *48*, 101–109.

(56) Opalka, D.; Domcke, W. Interpolation of multi-sheeted multi-dimensional potential-energy surfaces via a linear optimization procedure. *J. Chem. Phys.* **2013**, *138*, 224103.

(57) Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D. D. Hyperopt: a python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **2015**, *8*, 014008.

(58) Bergstra, J. S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. Advances in neural information processing systems. 2011; pp 2546–2554.

(59) Williams, C. K.; Rasmussen, C. E. *Gaussian processes for machine learning*; MIT Press Cambridge, MA, 2006; Vol. 2.

(60) Reed, R.; MarksII, R. J. *Neural smithing: supervised learning in feedforward artificial neural networks*; Mit Press, 1999.

(61) GPy: A Gaussian process framework in python. `http://github.com/SheffieldML/GPy`, 2012.

(62) Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. NIPS-W. 2017.

(63) Yu, Q.; Bowman, J. M. Ab Initio Potential for $H_3O^+ \rightarrow H^+ + H_2O$: A Step to a Many-Body Representation of the Hydrated Proton? *J. Chem. Theory Comput.* **2016**, *12*, 5284–5292.

(64) Fortenberry, R. C.; Yu, Q.; Mancini, J. S.; Bowman, J. M.; Lee, T. J.; Crawford, T. D.; Klemperer, W. F.; Francisco, J. S. Communication: Spectroscopic consequences of proton delocalization in $OCHCO^+$. 2015.

(65) Wang, X.; Houston, P. L.; Bowman, J. M. A new (multi-reference configuration interaction) potential energy surface for H_2CO and preliminary studies of roaming. *Phil. Trans. R. Soc. A* **2017**, *375*, 20160194.

(66) Brorsen, K. R. Reproducing global potential energy surfaces with continuous-filter convolutional neural networks. *J. Chem. Phys.* **2019**, *150*, 204104.

(67) Christiansen, O. Vibrational structure theory: New vibrational wave function methods for calculation of anharmonic vibrational energies and vibrational contributions to molecular properties. *Phys. Chem. Chem. Phys.* **2007**, *9*, 2942–2953.

(68) NITROGEN, Numerical and Iterative Techniques for Rovibronic Energies with General Internal Coordinates, a program by P. B. Changala, http://www.colorado.edu/nitrogen.

(69) Strobusch, D.; Scheurer, C. Hierarchical expansion of the kinetic energy operator in curvilinear coordinates for the vibrational self-consistent field method. *J. Chem. Phys.* **2011**, *135*, 124102.

(70) Strobusch, D.; Scheurer, C. The hierarchical expansion of the kinetic energy operator in curvilinear coordinates extended to the vibrational configuration interaction method. *J. Chem. Phys.* **2011**, *135*, 144101.

(71) Carter, S.; Culik, S. J.; Bowman, J. M. 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.* **1997**, *107*, 10458–10469.

(72) Bowman, J. M.; Carrington, T.; Meyer, H.-D. Variational quantum approaches for computing vibrational energies of polyatomic molecules. *Mol. Phys.* **2008**, *106*, 2145–2182.

(73) Light, J. C.; Carrington Jr, T. Discrete-variable representations and their utilization. *Adv. Chem. Phys.* **2000**, *114*, 263–310.

(74) Weigend, F.; Köhn, A.; Hättig, C. Efficient use of the correlation consistent basis sets in resolution of the identity MP2 calculations. *J. Chem. Phys.* **2002**, *116*, 3175–3183.

(75) East, A. L. L.; Allen, W. D. *J. Chem. Phys.* **1993**, *99*, 4638–4650.

(76) Császár, A. G.; Allen, W. D.; Schaefer, H. F. *J. Chem. Phys.* **1998**, *108*, 9751–9764.

(77) Schuurman, M. S.; Muir, S. R.; Allen, W. D.; Schaefer, H. F. *J. Chem. Phys.* **2004**, *120*, 11586–11599.

(78) Kendall, R. A.; Dunning Jr, T. H.; Harrison, R. J. Electron affinities of the first-row atoms revisited. Systematic basis sets and wave functions. *J. Chem. Phys.* **1992**, *96*, 6796–6806.

(79) Feller, D. Application of systematic sequences of wave functions to the water dimer. *J. Chem. Phys.* **1992**, *96*, 6104–6114.

(80) Feller, D. The use of systematic sequences of wave functions for estimating the complete basis set, full configuration interaction limit in water. *J. Chem. Phys.* **1993**, *98*, 7059–7071.

(81) Helgaker, T.; Klopper, W.; Koch, H.; Noga, J. Basis-set convergence of correlated calculations on water. *J. Chem. Phys.* **1997**, *106*, 9639–9646.

(82) Cowan, R. D.; Griffin, D. C. Approximate relativistic corrections to atomic radial wave functions. *J. Opt. Soc. Am.* **1976**, *66*, 1010–1014.

(83) Sellers, H.; Pulay, P. The adiabatic correction to molecular potential surfaces in the SCF approximation. *Chem. Phys. Lett.* **1984**, *103*, 463–465.

(84) Handy, N. C.; Yamaguchi, Y.; Schaefer III, H. F. The diagonal correction to the Born–Oppenheimer approximation: its effect on the singlet–triplet splitting of CH_2 and other molecular effects. *J. Chem. Phys.* **1986**, *84*, 4481–4484.

(85) Polyansky, O. L.; Császár, A. G.; Shirin, S. V.; Zobov, N. F.; Barletta, P.; Tennyson, J.; Schwenke, D. W.; Knowles, P. J. High-accuracy ab initio rotation-vibration transitions for water. *Science* **2003**, *299*, 539–542.

(86) Barletta, P.; Shirin, S. V.; Zobov, N. F.; Polyansky, O. L.; Tennyson, J.; Valeev, E. F.; Császár, A. G. CVRQD ab initio ground-state adiabatic potential energy surfaces for the water molecule. *J. Chem. Phys.* **2006**, *125*, 204307.

(87) Császár, A. G.; Mátyus, E.; Szidarovszky, T.; Lodi, L.; Zobov, N. F.; Shirin, S. V.; Polyansky, O. L.; Tennyson, J. First-principles prediction and partial characterization of the vibrational states of water up to dissociation. *J. Quant. Spectrosc. Radiat. Transf.* **2010**, *111*, 1043–1064.

(88) Tennyson, J.; Bernath, P. F.; Brown, L. R.; Campargue, A.; Császár, A. G.; Daumont, L.; Gamache, R. R.; Hodges, J. T.; Naumenko, O. V.; Polyansky, O. L.; Rothman, L. S.; Vandaele, A. C.; Zobov, N. F.; Al Derzi, A. R.; Fábri, C.; Fazliev, A. Z.; Furtenbacher, T.; Gordon, I. E.; Lodi, L.; Mizus, I. I. IUPAC critical evaluation of the rotational–vibrational spectra of water vapor, Part III: Energy levels and transition wavenumbers for $H_2^{16}O$. *J. Quant. Spectrosc. Radiat. Transf.* **2013**, *117*, 29–58.

(89) Cheng, L.; Gauss, J. Analytic energy gradients for the spin-free exact two-component theory using an exact block diagonalization for the one-electron Dirac Hamiltonian. *J. Chem. Phys.* **2011**, *135*, 084114.

(90) CFOUR, a quantum chemical program package written by J.F. Stanton, J. Gauss, L. Cheng, M.E. Harding, D.A. Matthews, and P.G. Szalay with contributions from A.A. Auer, R.J. Bartlett, U. Benedikt, C. Berger, D.E. Bernholdt, Y.J. Bomble, O. Christiansen, F. Engel, R. Faber, M. Heckert, O. Heun, C. Huber, T.-C. Jagau, D. Jonsson, J. Jusélius, K. Klein, W.J. Lauderdale, F. Lipparini, T. Metzroth, L.A. Mück, D.P. ONeill, D.R. Price, E. Prochnow, C. Puzzarini, K. Ruud, F. Schiffmann, W. Schwalbach, C. Simmons, S. Stopkowicz, A. Tajti, J. Vázquez, F. Wang, J.D. Watts and the integral packages MOLECULE (J. Almlöf and P.R. Taylor), PROPS (P.R. Taylor), ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, and J. Olsen), and ECP routines by A. V. Mitin and C. van Wüllen. For the current version, see http://www.cfour.de.

(91) Morgan, W. J.; Matthews, D. A.; Ringholm, M.; Agarwal, J.; Gong, J. Z.; Ruud, K.; Allen, W. D.; Stanton, J. F.; Schaefer III, H. F. Geometric Energy Derivatives at the Complete Basis Set Limit: Application to the Equilibrium Structure and Molecular Force Field of Formaldehyde. *J. Chem. Theory Comput.* **2018**, *14*, 1333–1350.

(92) Martin, J.; Lee, T. J.; Taylor, P. An accurate ab initio quartic force field for formaldehyde and its isotopomers. *J. Mol. Spectrosc.* **1993**, *160*, 105–116.

(93) Rauhut, G.; Knizia, G.; Werner, H.-J. Accurate calculation of vibrational frequencies using explicitly correlated coupled-cluster theory. *J. Chem. Phys.* **2009**, *130*, 054105.

(94) Yachmenev, A.; Yurchenko, S. N.; Jensen, P.; Thiel, W. A new spectroscopic potential energy surface for formaldehyde in its ground electronic state. *J. Chem. Phys.* **2011**, *134*, 244307.

(95) Tchana, F. K.; Perrin, A.; Lacome, N. New analysis of the ν_2 band of formaldehyde ($H_2^{12}C^{16}O$): Line positions for the ν_2, ν_3, ν_4 and ν_6 interacting bands. *J. Mol. Spectrosc.* **2007**, *245*, 141–144.

(96) Perrin, A.; Valentin, A.; Daumont, L. New analysis of the $2\nu_4$, $\nu_4 + \nu_6$, $2\nu_6$, $\nu_3 + \nu_4$, ν_3i + ν_6, ν_1, ν_5, $\nu_2 + \nu_4$, $2\nu_3$, $\nu_2 + \nu_6$ and $\nu_2 + \nu_3$ bands of formaldehyde $H_2^{12}C^{16}O$: Line positions and intensities in the 3.5 μm spectral region. *J. Mol. Struct.* **2006**, *780*, 28–44.

(97) Bouwens, R. J.; Hammerschmidt, J. A.; Grzeskowiak, M. M.; Stegink, T. A.; Yorba, P. M.; Polik, W. F. Pure vibrational spectroscopy of S_0 formaldehyde by dispersed fluorescence. *J. Chem. Phys.* **1996**, *104*, 460–479.

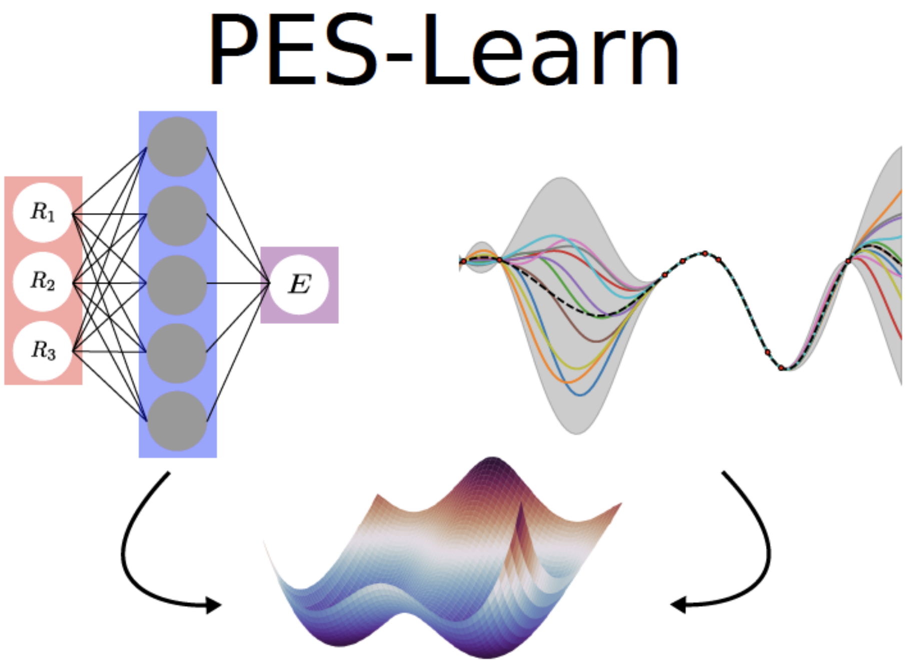
(98) Perez, R.; Brown, J. M.; Utkin, Y.; Han, J.; Curl, R. F. Observation of hot bands in the infrared spectrum of H_2CO. *J. Mol. Spectrosc.* **2006**, *236*, 151–157.

(99) Flaud, J.-M.; Lafferty, W. J.; Sams, R. L.; Sharpe, S. W. High resolution spectroscopy of $H_2^{12}C^{16}O$ in the 1.9 to 2.56 μm spectral range. *Mol. Phys.* **2006**, *104*, 1891–1903.

(100) Wilkins-Diehr, N.; Crawford, T. D. NSFs inaugural software institutes: The science gateways community institute and the molecular sciences software institute. *Comput. Sci. Eng.* **2018**, *20*, 26–38.

(101) Krylov, A.; Windus, T. L.; Barnes, T.; Marin-Rimoldi, E.; Nash, J. A.; Pritchard, B.; Smith, D. G.; Altarawy, D.; Saxe, P.; Clementi, C.; Crawford, T. D.; Harrison, R. J.; Jha, S.; Pande, V. S.; Head-Gordan, T. Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science. *J. Chem. Phys.* **2018**, *149*, 180901.

For Table of Contents use only


PES-Learn
R_1
R_2
R_3
E