physchem

MDPI

Article

On the Sufficiency of a Single Hidden Layer in Feed-Forward Neural Networks Used for Machine Learning of Materials Properties

Ye Min Thant [1], Sergei Manzhos [2,*], Manabu Ihara [2] and Methawee Nukunudompanich [1,*]

[1] Department of Industrial Engineering, School of Engineering, King's Mongkut Institute of Technology Ladkrabang (KMITL), 1 Chalong Krung, 1 Alley, Lat Krabang, Bangkok 10520, Thailand; 65011695@kmitl.ac.th
[2] School of Materials and Chemical Technology, Institute of Science Tokyo, Ookayama 2-12-1, Meguro-ku, Tokyo 152-8552, Japan; mihara@chemeng.titech.ac.jp
* Correspondence: manzhos.s.aa@m.titech.ac.jp (S.M.); methawee.nu@kmitl.ac.th (M.N.)

Abstract: Feed-forward neural networks (NNs) are widely used for the machine learning of properties of materials and molecules from descriptors of their composition and structure (materials informatics) as well as in other physics and chemistry applications. Often, multilayer (so-called "deep") NNs are used. Considering that universal approximator properties hold for single-hidden-layer NNs, we compare here the performance of single-hidden-layer NNs (SLNN) with that of multilayer NNs (MLNN), including those previously reported in different applications. We consider three representative cases: the prediction of the band gaps of two-dimensional materials, prediction of the reorganization energies of oligomers, and prediction of the formation energies of polyaromatic hydrocarbons. In all cases, results as good as or better than those obtained with an MLNN could be obtained with an SLNN, and with a much smaller number of neurons. As SLNNs offer a number of advantages (including ease of construction and use, more favorable scaling of the number of nonlinear parameters, and ease of the modulation of properties of the NN model by the choice of the neuron activation function), we hope that this work will entice researchers to have a closer look at when an MLNN is genuinely needed and when an SLNN could be sufficient.

Keywords: machine learning; materials informatics; multilayer neural network; single-hidden-layer neural network

Academic Editor: Ray Luo

Received: 15 December 2024
Revised: 8 January 2025
Accepted: 14 January 2025
Published: 16 January 2025

Citation: Thant, Y.M.; Manzhos, S.; Ihara, M.; Nukunudompanich, M. On the Sufficiency of a Single Hidden Layer in Feed-Forward Neural Networks Used for Machine Learning of Materials Properties. *Physchem* **2025**, *5*, 4. https://doi.org/10.3390/physchem5010004

1. Introduction

Prediction of properties of materials and molecules from descriptors of their chemical composition and structure (which forms a substantial part of materials informatics) is actively researched due to its promise of accelerated in silico design of novel materials, cutting experimental and human costs and lead times to materials development [1–3]. Various machine learning (ML) techniques are used for this purpose, from simple linear regression to regularized linear regressions [4] to more complex methods like decision trees [5,6], clustering [7], kernel methods [8–10], and neural networks of various kinds [11–15]. Neural networks (NNs) are most widely used for this purpose. Many materials informatics problems are multivariate regression-type problems or can be represented as such, and feed-forward NNs (FFNNs) are often used to solve them [14,16–18].

More often than not, multilayer NNs (MLNN), or so-called "deep" NNs, are used [18–22]. It is perceived that they have a higher expressive power than "shallow"

NNs such as single-hidden-layer NNs. However, the mathematical foundation for NNs' ability to represent any smooth multivariate function via a combination of univariate functions that are neuron activation functions [23] are universal approximator theorems [24–38] that were formulated for *single*-hidden-layer NNs. A single-hidden-layer NN (SLNN) is a universal approximator in the sense that for any smooth function $f(\boldsymbol{x})$, $\boldsymbol{x} \in R^D$ (such as a material property as a function of D descriptors), and any $\delta > 0$, there exists a finite number of neurons $N < \infty$ such that

$$\begin{gathered} \left|f^{NN}(\boldsymbol{x}) - f(\boldsymbol{x})\right| < \delta \\ f^{NN}(\boldsymbol{x}) = \sigma_{out}\left(\sum_{n=1}^{N} c_n \sigma_n(\boldsymbol{w}_n \boldsymbol{x} + b_n) + b_0\right), \end{gathered} \tag{1}$$

where $f^{NN}(\boldsymbol{x})$ is a single-hidden-layer NN approximation of $f(\boldsymbol{x})$ on some relevant domain spanned by $\boldsymbol{x}$. $\boldsymbol{w_i}$ and b_j are weights and biases. Neuron activation functions $\sigma_n(y)$ (where y is a scalar variable) can be any smooth nonlinear functions [38], although in practice, for algorithmic simplicity, they are often chosen to be the same for all n and of one of commonly used simple shapes such as sigmoid, $\sigma(y) = 2/\left(1 + e^{-2y}\right) - 1$, CELU (continuously differentiable exponential linear unit), $\sigma(y) = \max(0, y) + \min\left(0, \alpha\left(e^{x/\alpha} - 1\right)\right)$, etc.

By the universal approximator theorems, there should never be a need for an MLNN. The universal approximator property, however, refers to the expressive power of an SLNN representation and ignores the data aspect of the machine learning problem that may inhibit the construction of the best SLNN when the data are sparse (sampling density is low). It is necessarily so when D is sufficiently high [39–43]. As an example, in Ref. [44], when fitting kinetic energy densities (KEDs) [45] of different materials, SLNNs were sufficient to accurately reproduce Kohn–Sham KEDs of individual materials, but an SLNN was not able to reproduce KEDs simultaneously for several materials (i.e., to achieve model portability across materials). This was related to sampling issues and could be addressed with an MLNN [44]. This is but one example of a genuine need for an MLNN. An MLNN can also be useful to realize NN architectures (by the design of connections between layers) that reflect the hierarchy of the data (e.g., when data lie on sub-dimensional hyperplanes) or to realize certain properties of the approximation such as expansions over orders of coupling or many-body terms [40]. An MLNN, however, comes at the cost of a substantial number of nonlinear parameters, whereby each interlayer connection adds a number of weights, which is the product of the numbers of neurons in the two layers, as well as the corresponding biases. This increases the computational cost (of both fitting and using the NN) and complicates the search for an optimal architecture (optimal hyperparameters—numbers of layers and neurons in each layer that should ideally be optimized), ultimately increasing the danger of overfitting. For example, when building orders of coupling representations, we found it more efficient to use multiple simple all-connected NNs representing individual terms of the hierarchy rather than dealing with an MLNN realizing the hierarchy [40,43,46]; for those terms, we saw no advantage of using an MLNN when fitting molecular potential energy surfaces (PESs). In our experience, we also saw no advantage of using MLNNs when fitting PESs also in an all-connected (non-hierarchical) architecture [47–52], although MLNNs are much used in the PES literature [53–55]. The results are, however, expected to be application-dependent. Here, we concern ourselves with the machine learning of materials properties.

An SLNN has other advantages in addition to the simplicity of the architecture (and the associated lower CPU cost and relative ease of identifying optimal hyperparameters). With an SLNN, it is easier to achieve the desired properties of $f^{NN}(\boldsymbol{x})$ by the choices of the neuron activation function. For example, a sum-of-product form (that greatly facilitates multidimensional integration) is trivially achieved by using $\sigma(y) = e^y$ [56]. An SLNN

architecture allows relatively easily building NNs with optimal (for given data and N) and individual to each neuron activation function [57] and achieving orders-of-coupling representations [58] (that facilitate the construction of reliable ML models with sparse data [40]).

While researchers tend to use MLNNs, it is possible that in many applications, an SLNN would be able to do as good a job. Here, we therefore explore this possibility and consider several representative cases of machine learning of materials properties for which we compare SLNNs with MLNNs. We consider two cases where previously multilayer NNs were used: the prediction of the band gap of two-dimensional materials [59], prediction of the reorganization energies of oligomers [60], and prediction of a measure of formation energy (relative energy vs. the lowest energy isomer) of polyaromatic hydrocarbons (PAHs) [61]. In all cases, the results obtained with an SLNN were as good as or better than those obtained with an MLNN. We therefore hope that this work will entice researchers to have a closer look at when an MLNN is genuinely needed and when an SLNN could be sufficient when selecting an appropriate NN architecture.

2. Materials and Methods

The machine learning calculations were conducted using the Neural Net Fitting and Regression Learner toolboxes in MATLAB (R2024a).

2.1. Two-Dimensional Material Dataset for Band Gap Prediction

2.1.1. Data Preprocessing and Features

The dataset prepared by collecting band gaps of 2D materials from the Computational 2D Materials Database (C2DB) [62] was used to train an SLNN. The dataset contains 3129 datapoints calculated by density functional theory (DFT) [63] using the PBE [64] functional. The database was built using experimentally known crystal structure prototypes and chemical compositions using atoms chosen from a (chemically reasonable) subset of the periodic table; it included commonly explored structures such as transition metal chalcogenides, perovskites, etc. The features used in the model training included the density of states at the Fermi energy, heat of formation, number of atoms, sum of atomic masses in the unit cell, total energy, maximum force, maximum stress on a unit cell, volume of the unit cell, and band gap without spin–orbit coupling (SOC). These descriptors, most of which are DFT-calculated properties, are available from the C2DB database and were used in Ref. [59] to fit an MLNN. We used the same descriptors here because our purpose was to carry out comparisons with an SLNN; we do not necessarily agree that all of these are reasonable descriptors (for example, unit cell forces and stresses).

2.1.2. Model Training

The dataset was split into 2817 datapoints for the training set and 312 datapoints for the test set. Zhang et al. [59] applied an MLNN with 4 hidden layers to predict the band gap using this dataset. They used 262, 140, 139, and 180 neurons with the hyperbolic tangent activation function in each hidden layer. This resulted in about 84,420 nonlinear parameters. The Adam optimizer was used to train the MLNN. In our SLNN model, 9 neurons were used in the hidden layer, and each neuron used the hyperbolic tangent sigmoid function (tansig). The Levenberg–Marquardt optimization algorithm was used to train the SLNN model using the mean squared error (MSE) loss function. The model was trained using two descriptor sets: a 9-feature set including the band gap with spin–orbit coupling and an 8-feature set excluding it. The values of the MSE converge after 25 and 120 epochs, respectively.

2.2. Reorganization Energy Dataset and Model

2.2.1. Data Preprocessing and Features

The dataset used in this study was derived from 253 thiophene-based monomers with different functional groups at the 3 and 4 positions while connected to other monomers at the 2 and 5 positions; it is the same dataset as that used in Ref. [60] to predict reorganization energies with an MLNN. The initial combination yields 31,878 copolymers and 253 homopolymers, resulting in a total of 32,131 possible oligomer families. After preprocessing the data, the final dataset consisted of 7020 tetramers and 408 hexamers with reorganization energies (λ) computed with the DFT and B3LYP [65] functional, ranging about 0.01–2 eV. Geometrical data, including the oligomer length, neutral angle, cation angle, mean angle, neutral bond length, cation bond length, and mean bond length, were used as input features. The size of the π-system in the oligomer was also added to improve the results. In addition, extended circular fingerprints (ECFP4) [66] with a 2048-bit representation using RDKit [67] were also included as molecular features in the descriptors set. These descriptors are similar to those used in Ref. [60].

2.2.2. Model Training

A feed-forward neural network with a single hidden layer was trained on the dataset. The dataset was split into 85% and 15% for the training set and test set, respectively. MATLAB's Regression Learner was applied to find the optimal neurons, neural activation function, and learning parameters within the iteration limit of 200. The final optimized model was achieved with 30 neurons using the hyperbolic tangent function. The dataset was standardized to avoid large variation from centered feature values. Recently, Abarbanel et al. [60] used an MLNN with 2 hidden layers with 127 and 109 nodes in each hidden layer. They used a continuously differentiable exponential linear unit (CELU) as the activation function for the input and hidden layers and linear activation function in the output layer with a single node. They used the Adam optimizer for training the MLNN using the mean squared error (MSE) loss function. In our SLNN model, the L-BFGS algorithm and MSE loss function were used. Because Ref. [60] used regularization, we also used a regularized NN to facilitate comparisons. The optimal regularization parameter was about 0.01.

2.3. PAH Dataset and Model

2.3.1. Data Preprocessing and Features

A dataset of peri-condensed polybenzenoid hydrocarbons (Compas3) [61] was also used to compare the performance of the SLNN and MLPNN. This dataset contained optimized ground-state structures of approximately 39 thousand and 9 thousand molecules with molecular properties calculated at the GFN2-xTB [68] and DFT/CAM-B3LYP [69] levels of theory, respectively. Among available properties, including HOMO, LUMO, total energies, relative energy, dipole moment, etc., the DFT-calculated relative energy vs. lowest-energy isomer was chosen as the target to fit with both models. This is a more stringent test than fitting total energies or atomization energies. We used the eigenspectrum of the Coulomb matrix (ECM) [70] obtained from the structural parameters of 8844 hydrocarbons as input features (we did not consider $C_{16}H_{10}$ as it is the only molecule of this size in the dataset and is an outlier in the space of descriptors). As the dimension of the CM is molecule size-dependent, the dimensionality of the ECM-based feature vector was set to 62 based on the largest molecule in the dataset, and smaller molecules' ECM vectors were padded with zeros to have a length of 62.

2.3.2. Model Training

The dataset was split into 80% for the training set and 20% for the test set. The Levenberg–Marquardt optimization algorithm was applied to train both the SLNN and MLPNN models with the hyperbolic tangent sigmodal function as the neuron activation function. The MSE cost function was used to measure the performance. The number of neurons in the SLNN was searched by trying different number of neurons, and the minimum MSE on the test set was achieved with 10 neurons. To compare the potential of the SLNN with MLPNN, 3 hidden layers were used in the MLPNN. The number of neurons for each hidden layer was searched using grid search within 8–10 neurons, and then the third layer was adjusted with different trials of 1–10 neurons. The optimal MSE on the test set was achieved with [10, 10, 5] nodes for each. Convergence was achieved in less than 100 training epochs with all NNs.

3. Results

3.1. Prediction of Band Gap of 2D Materials

In Ref. [59], the band gap of 2D materials collected from the C2DB database was predicted using gradient boosting decision trees (GBDT), random forest (RF), support vector regression (SVR), and an MLNN. They used an MLNN with four hidden layers that resulted in a test set RMSE of 0.43 eV and a test set MAE of 0.24 eV with eight input features. The performance of the model was improved when the band gap without spin–orbit coupling was added as an input feature (the reference band gap values were computed using SOC corrections). This resulted in a test set RMSE of 0.12 eV and MAE of 0.06 eV.

With an SLNN, on the same data, we obtained RMSE values of $0.30 \pm 0.01/0.34 \pm 0.04$ eV and MAEs of $0.14 \pm 0.01/0.15 \pm 0.02$ eV (train/test) when using eight features, and the performance improved significantly when the gap without SOC was added as the ninth feature, resulting in RMSEs of $0.07 \pm 0.01/0.08 \pm 0.01$ eV and MAEs of $0.03 \pm 0.00/0.03 \pm 0.01$ eV, respectively. Here and elsewhere, "$\pm$" refers to the spread of values over 10 runs differing by random training–test splits. This spread indicates the extent of the reliability of the model with respect to test data choices as well as the extent of influence of different local minima during NN optimization (weights and biases are randomly initialized in each run). In the SLNN model, the best performance is achieved using only nine neurons in the hidden layer, and the value of the MSE stabilized after 120 epochs with the eight-feature set and after only four epochs with the nine-feature set. An example of the learning curve of the SLNN with the eight-feature input is shown in Figure 1. All learning curves in this and other sections were similarly taken to convergence and are not shown below. This shows that an SLNN model can provide comparable or better results than an MLNN using fewer neurons, when the hyperparameters are adjusted correctly. The correlation plots of the DFT-calculated band gaps and predicted band gaps on the test set using the SLNN and MLNN are compared for eight-feature and nine-feature inputs in Figures 2 and 3, respectively.

We also explored MLNNs with the same number of neurons as the best SLNN and did not see an advantage since the results are quite similar. For example, the best MLNN with [5 4] neurons resulted in an RSME of $0.07 \pm 0.01/0.08 \pm 0.02$ and MAE of $0.03 \pm 0.00/0.03 \pm 0.01$ (train/test), when using nine features.

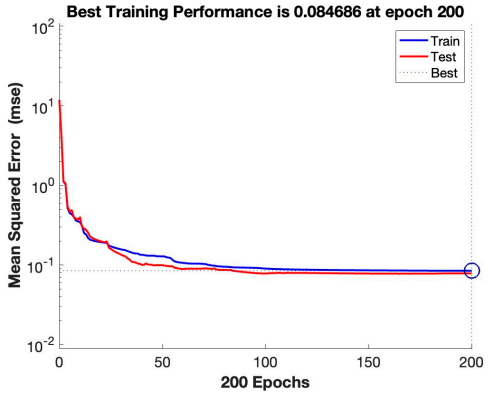
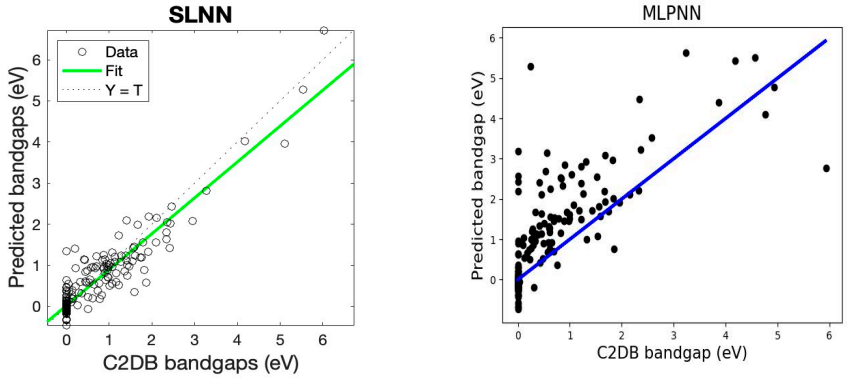
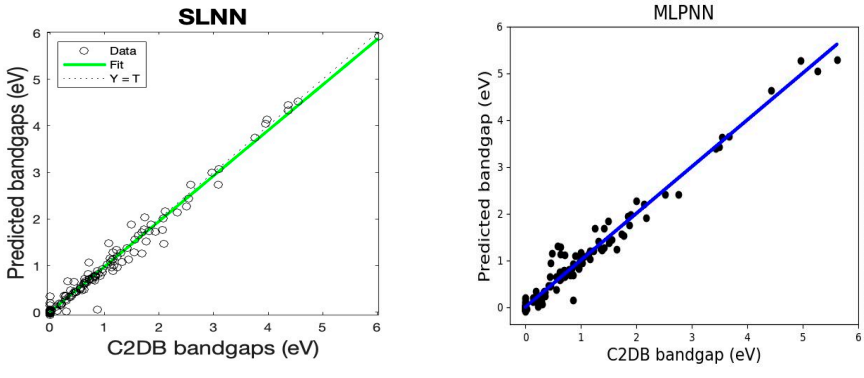


Figure 1. Example of learning curves of SLNN model on training with 8-feature dataset.



(**a**) (**b**)

Figure 2. Correlation plots of predicted vs. reference band gaps for the test set when using 8 input features (excluding the band gap computed without spin–orbit coupling): (**a**) single-layer neural network (SLNN); (**b**) multilayered perceptron neural network (MLNN). The MLNN graphic is reproduced from Ref. [59] under Creative Commons Attribution 4.0 International (CC BY) license. The symbols in each graph represent the datapoints and solid lines the fit. The ideal correlation ("Y = T") is shown in (**a**) as a dashed line.



(**a**) (**b**)

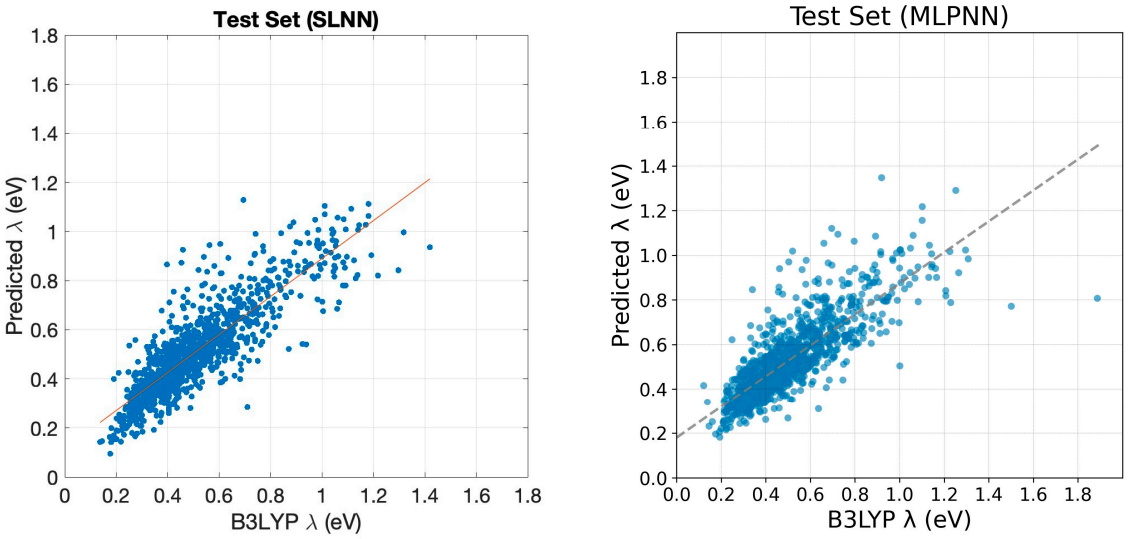
Figure 3. Correlation plots of predicted vs. reference band gaps for the test set when using 9 input features (including the band gap computed without spin–orbit coupling): (**a**) single-layer neural network (SLNN); (**b**) multilayered perceptron neural network (MLNN). The MLNN graphic is reproduced from Ref. [59] under Creative Commons Attribution 4.0 International (CC BY) license. The symbols in each graph represent the datapoints and solid lines the fit. The ideal correlation ("Y = T") is shown in (**a**) as a dashed line.

3.2. Prediction of Reorganization Energies of Thiophene-Based Oligomers

The MLNN used by Abarbanel et al. [60] resulted in a test set RMSE of 0.122 ± 0.001 eV and R^2 0.636 ± 0.009 eV with a three-fold cross validation set for the prediction of reorganization energies of thiophene-based oligomers, while the best result was achieved with an RF model with an RMSE of 0.107 eV and R^2 of 0.719 on the test set with a train/test ratio of 0.85/0.15. With the same ratio of train/test splitting, our SLNN model achieves an RMSE = 0.093 ± 0.006/0.102 ± 0.001 eV and R^2 = 0.789 ± 0.030/0.732 ± 0.008 on the test set, using far fewer hidden layer neurons (30 vs. 127 + 109).

We noted that among the features, monomer IDs that did not have physical meaning were included in Ref. [60]. Moreover, the RF model of Ref. [60] identified these as having substantial feature importance; these ID ranked fourth and sixth in importance among 10 features and had higher importance than some bonds and angle features [60]. This highlights that there is real danger of obtaining nonsensical results when trying to generate insight with black-box ML methods.

Therefore, from the original set of descriptors, the two monomer ID features were discarded when training the SLNN. Without these two features, an MLNN with the same hyperparameters as in Ref. [60] was also run. This resulted in better performance, with an RMSE of 0.116 eV and R^2 of 0.682, confirming that these two features do not possess predictive power. The correlation plots of the predicted λ and DFT-calculated λ are compared in Figure 4. The results show that an SLNN can produce accuracy comparable to that of an MLNN when the hyperparameters are appropriate and the training algorithm is sufficiently powerful. The SLNN accuracy is also competitive with the best results given by the RF model in this case. Parameters used in training the SLNN and MLNN are listed in Table 1.



(**a**) (**b**)

Figure 4. Correlation plots of predicted reorganization energy λ ("Predicted λ") and DFT-calculated λ ("B3LYP λ") on the test set for (**a**) single-layer neural network (SLNN) and (**b**) multilayered perceptron neural network (MLPNN).

We also explored MLNNs with the same number of neurons as the best SLNN and did not see an advantage. The best MLNN had [15 15] neurons in the hidden layers and resulted in an RSME of 0.096 ± 0.004/0.101 ± 0.001 eV and R^2 of 0.779 ± 0.020/0.736 ± 0.006 (train/test), i.e., no statistically significant advantage over an SLNN with 30 neurons.

Table 1. Parameters used during model training.

Parameters	SLNN	MLPNN
Number of neurons	30	127, 109
Number of epochs	547	151
Methods	Tanh	CELU in input and hidden layers, linear activation function at the output layer
Solver	LBFGS	Adam

3.3. Prediction of Relative Energies of Peri-Condensed Hydrocarbon Molecules

It has been found that using 62 features–eigenvalues of the ECM results in good prediction accuracy in both SLNN and MLPNN models. We tried to carry out the fit with a smaller subset of eigenvalues but there is a significant drop in the fit accuracy unless at least 40 values are used. This can be rationalized based on Figure 5 that shows the plot of the cumulative explained variance ratio versus the number of eigenvalues used.

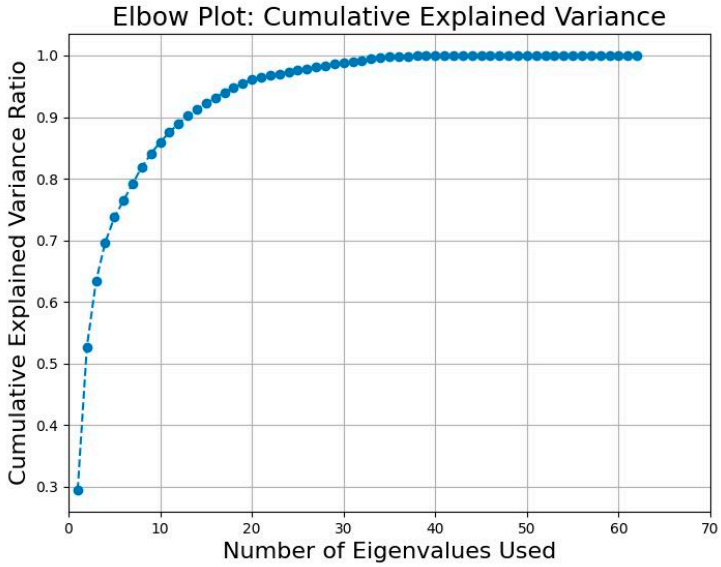


Figure 5. Cumulative explained variance ratio versus the number of eigenvalues of the Coulomb matrix used in the feature set.

We therefore used the full (62-dimensinal) ECM vector and compared the RMSE and R^2 values achieved with the SLNN and MLPNN. The SLNN model resulted in an RMSE of $0.041 \pm 0.002/0.063 \pm 0.021$ eV and R^2 of $0.98 \pm 0.001/0.95 \pm 0.04$ (train/test), while the MLPNN with three hidden layers achieved an RMSE of $0.037 \pm 0.002/0.053 \pm 0.007$ eV and R^2 values of $0.98 \pm 0.002/0.97 \pm 0.010$. The correlation plots from both models for the training and test sets are shown in Figure 6. The RMSE of the best-identified MLPNN is only slightly better than the results of the SLNN for the training set and is identical for the test set (which is the only error that matters as the training set error can always be driven to zero just by growing the NN). To check the reliability of the predicted values, the mean absolute percentage error (MAPE) wass also calculated for both models. The MAPE of the SLNN was $4.22 \pm 0.17/5.03 \pm 0.34\%$ and that of the MLPNN was $3.91 \pm 0.23/4.82 \pm 0.28\%$ on the training and test sets, respectively. Overall, no advantage of using a multilayer NN was found.

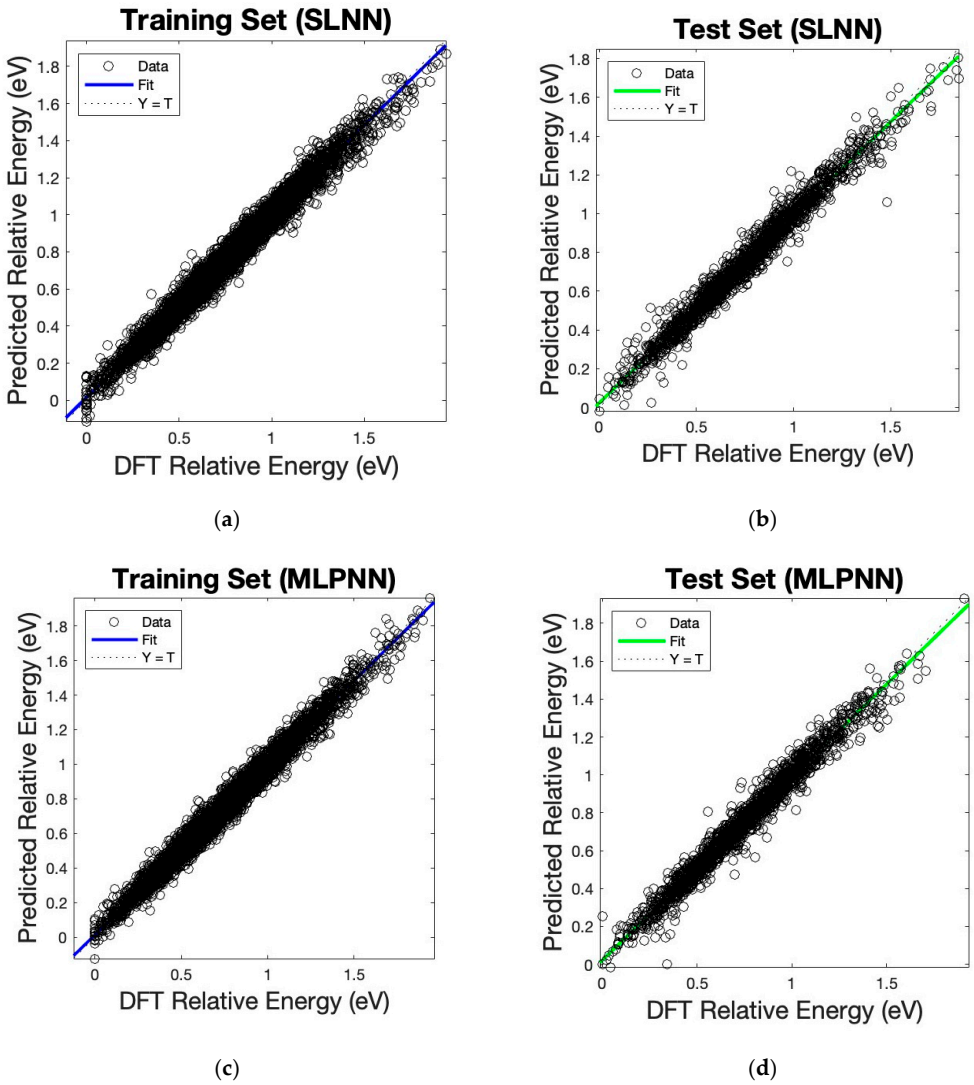


Figure 6. Correlation plots of B3LYP relative energy and predicted relative energy (vs. lowest-energy isomer) of peri-condensed polybenzenoid hydrocarbons: (**a**) training set with SLNN; (**b**) test set with SLNN; (**c**) training set with MLPNN; (**d**) test set with MLPNN.

We also explored an MLNN with the same number of neurons as the best SLNN. An MLNN with [7 3] neurons in the hidden layers resulted in an RMSE of 0.053 ± 0.003/ 0.063 ± 0.003 eV and R^2 of 0.97 ± 0.003/0.96 ± 0.004 (train/test), which is not statistically different from the SLNN with the same number of neurons.

4. Conclusions

We compared the performance of single-hidden-layer neural networks (SLNNs) to that of multilayer neural networks (MLNNs) for several representative problems of machine learning of materials properties often encountered in the literature: the learning of band gaps, reorganization energies, and formation energies. We used types of features commonly used in such studies. In the first two examples, we used the same data and features that were used with MLNNs in previous research, and therefore, a direct comparison with the results of those MLNNs could be carried out. The final example used in-house featurization (which was performed with a widely used Coulomb matrix-based method) and an MLNN. As we used common types of machine learning problems and features, the results were expected to have some generality.

In all examples, the SLNN was able to achieve as good an accuracy as a previously reported or in-house optimized MLNN. While MLNNs are widely used and appear to be preferred in the literature, their CPU cost is usually higher.. In some cases, the use of relatively large MLNNs instead of an SLNN that would provide sufficient accuracy may have to do with software choices. Many MLNN works used Python-based libraries that, for example, do not provide sufficiently powerful optimizers like the Levenberg–Marquardt algorithm. Among our examples, we, therefore, also carried out MLNN and SLNN comparisons using the same software (MATLAB) and the same efficient Levenberg–Marquardt optimizer. We still did not see any advantage of using more than one hidden layer.

Our results are not meant to imply that an SLNN is always a better choice; indeed, we cited an example where an MLNN provided substantial improvement [44]. They are meant to show that in various real-life applications where MLNNs are routinely used, SLNNs can provide an accuracy that is at least as good. An SLNN could be a less costly, easier-to-optimize option (whereby one optimizes one number of neurons rather than incurring the multidimensional optimization of the numbers of neurons in many layers) that also provides additional advantages, such as obtaining various properties (e.g., sum-of-products), through the simple choice of the neuron activation function. We hope that this work will serve to help researchers better select an appropriate NN architecture when machine learning materials properties, including the possibility of using an SLNN in cases where it could be sufficient, versus a costlier and more difficult-to-optimize MLNN.

Author Contributions: Conceptualization, S.M. and M.N.; data curation, Y.M.T.; formal analysis, Y.M.T.; investigation, Y.M.T.; methodology, S.M.; project administration, M.N.; resources, M.I. and M.N.; software, Y.M.T.; supervision, M.N.; validation, Y.M.T.; writing—original draft, Y.M.T., S.M. and M.N.; writing—review and editing, Y.M.T., S.M., M.I. and M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this work can be found in references [60–62].

Acknowledgments: We thank Tengxiang Li and Khant Nyi Lynn for assistance with some of the calculations.

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

References

1. Liu, Y.; Esan, O.C.; Pan, Z.; An, L. Machine learning for advanced energy materials. *Energy AI* **2021**, *3*, 100049. [CrossRef]
2. Lu, Z. Computational discovery of energy materials in the era of big data and machine learning: A critical review. *Mater. Rep. Energy* **2021**, *1*, 100047. [CrossRef]
3. Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine learning in materials informatics: Recent applications and prospects. *Npj Comput. Mater.* **2017**, *3*, 54. [CrossRef]
4. Bishop, C.M. *Pattern Recognition and Machine Learning*; Information science and statistics; Springer: New York, NY, USA, 2006; ISBN 978-0-387-31073-2.
5. Stoll, A.; Benner, P. Machine learning for material characterization with an application for predicting mechanical properties. *GAMM-Mitteilungen* **2021**, *44*, e202100003. [CrossRef]
6. Dabiri, H.; Farhangi, V.; Moradi, M.J.; Zadehmohamad, M.; Karakouzian, M. Applications of decision tree and random forest as tree-based machine learning techniques for analyzing the ultimate strain of spliced and non-spliced reinforcement bars. *Appl. Sci.* **2022**, *12*, 4851. [CrossRef]
7. Kaneko, H. Clustering method for the construction of machine learning model with high predictive ability. *Chemom. Intell. Lab. Syst.* **2024**, *246*, 105084. [CrossRef]
8. Allen, A.E.A.; Tkatchenko, A. Machine learning of material properties: Predictive and interpretable multilinear models. *Sci. Adv.* **2022**, *8*, eabm7185. [CrossRef]
9. Chen, K.; Kunkel, C.; Reuter, K.; Margraf, J.T. Reorganization energies of flexible organic molecules as a challenging target for machine learning enhanced virtual screening. *Digit. Discov.* **2022**, *1*, 147–157. [CrossRef]

10. Olsthoorn, B.; Geilhufe, R.M.; Borysov, S.S.; Balatsky, A.V. Band gap prediction for large organic crystal structures with machine learning. *Adv. Quantum Technol.* **2019**, *2*, 1900023. [CrossRef]
11. *Neural Networks: Tricks of the Trade*, 2nd ed.; 2012 edition; Montavon, G., Orr, G., Müller, K.-R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; ISBN 978-3-642-35288-1.
12. Hong, Y.; Hou, B.; Jiang, H.; Zhang, J. Machine learning and artificial neural network accelerated computational discoveries in materials science. *WIREs Comput. Mol. Sci.* **2020**, *10*, e1450. [CrossRef]
13. Bhadeshia, H.K.D.H. Neural networks in materials science. *ISIJ Int.* **1999**, *39*, 966–979. [CrossRef]
14. Scott, D.J.; Coveney, P.V.; Kilner, J.A.; Rossiny, J.C.H.; Alford, N.M.N. Prediction of the functional properties of ceramic materials from composition using artificial neural networks. *J. Eur. Ceram. Soc.* **2007**, *27*, 4425–4435. [CrossRef]
15. Zheng, X.; Zheng, P.; Zhang, R.-Z. Machine learning material properties from the periodic table using convolutional neural networks. *Chem. Sci.* **2018**, *9*, 8426–8432. [CrossRef]
16. Demirbay, B.; Kara, D.B.; Uğur, Ş. A bayesian regularized feed-forward neural network model for conductivity prediction of ps/mwcnt nanocomposite film coatings. *Appl. Soft Comput.* **2020**, *96*, 106632. [CrossRef]
17. Loh, G.C.; Lee, H.-C.; Tee, X.Y.; Chow, P.S.; Zheng, J.W. Viscosity prediction of lubricants by a general feed-forward neural network. *J. Chem. Inf. Model.* **2020**, *60*, 1224–1234. [CrossRef]
18. Çaylak, O.; Yaman, A.; Baumeier, B. Evolutionary approach to constructing a deep feedforward neural network for prediction of electronic coupling elements in molecular materials. *J. Chem. Theory Comput.* **2019**, *15*, 1777–1784. [CrossRef]
19. Tsubaki, M.; Mizoguchi, T. Fast and accurate molecular property prediction: Learning atomic interactions and potentials with neural networks. *J. Phys. Chem. Lett.* **2018**, *9*, 5733–5741. [CrossRef] [PubMed]
20. Ye, S.; Li, B.; Li, Q.; Zhao, H.-P.; Feng, X.-Q. Deep neural network method for predicting the mechanical properties of composites. *Appl. Phys. Lett.* **2019**, *115*, 161901. [CrossRef]
21. Ye, W.; Chen, C.; Wang, Z.; Chu, I.-H.; Ong, S.P. Deep neural networks for accurate predictions of crystal stability. *Nat. Commun.* **2018**, *9*, 3800. [CrossRef]
22. Fatriansyah, J.F.; Surip, S.N.; Hartoyo, F. Mechanical property prediction of poly(lactic acid) blends using deep neural network. *Evergreen* **2022**, *9*, 141–144. [CrossRef]
23. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*; Russian Academy of Sciences: Moscow, Russia, 1957; Volume 114, pp. 953–956.
24. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
25. Scarselli, F.; Chung Tsoi, A. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Netw.* **1998**, *11*, 15–37. [CrossRef] [PubMed]
26. Kůrková, V. Kolmogorov's theorem and multilayer neural networks. *Neural Netw.* **1992**, *5*, 501–506. [CrossRef]
27. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Syst.* **1989**, *2*, 303–314. [CrossRef]
28. Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Netw.* **1989**, *2*, 183–192. [CrossRef]
29. Nees, M. Approximative versions of kolmogorov's superposition theorem, proved constructively. *J. Comput. Appl. Math.* **1994**, *54*, 239–250. [CrossRef]
30. Hornik, K.; Stinchcombe, M.; White, H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw.* **1990**, *3*, 551–560. [CrossRef]
31. Katsuura, H.; Sprecher, D.A. Computational aspects of kolmogorov's superposition theorem. *Neural Netw.* **1994**, *7*, 455–461. [CrossRef]
32. Sprecher, D.A. A numerical implementation of kolmogorov's superpositions. *Neural Netw.* **1996**, *9*, 765–772. [CrossRef]
33. Sprecher, D.A. A numerical implementation of kolmogorov's superpositions ii. *Neural Netw.* **1997**, *10*, 447–457. [CrossRef]
34. Sprecher, D.A. An improvement in the superposition theorem of kolmogorov. *J. Math. Anal. Appl.* **1972**, *38*, 208–213. [CrossRef]
35. Sprecher, D.A. On the structure of continuous functions of several variables. *Am. Math. Soc.* **1965**, *115*, 340–355. [CrossRef]
36. Sprecher, D.A. A universal mapping for kolmogorov's superposition theorem. *Neural Netw.* **1993**, *6*, 1089–1094. [CrossRef]
37. Sprecher, D.A.; Draghici, S. Space-filling curves and kolmogorov superposition-based neural networks. *Neural Netw.* **2002**, *15*, 57–67. [CrossRef] [PubMed]
38. Gorban, A.N. Approximation of continuous functions of several variables by an arbitrary nonlinear continuous function of one variable, linear functions, and their superpositions. *Appl. Math. Lett.* **1998**, *11*, 45–49. [CrossRef]
39. Donoho, D.L. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Chall. Lect.* **2000**, *1*, 32.
40. Manzhos, S.; Carrington, T.; Ihara, M. Orders of coupling representations as a versatile framework for machine learning from sparse data in high-dimensional spaces. *Artif. Intell. Chem.* **2023**, *1*, 100008. [CrossRef]
41. Manzhos, S.; Ihara, M. Advanced machine learning methods for learning from sparse data in high-dimensional spaces: A perspective on uses in the upstream of development of novel energy technologies. *Physchem* **2022**, *2*, 72–95. [CrossRef]

42. Manzhos, S.; Yamashita, K.; Carrington, T. Extracting functional dependence from sparse data using dimensionality reduction: Application to potential energy surface construction. In *Proceedings of the Coping with Complexity: Model Reduction and Data Analysis*; Gorban, A.N., Roose, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 133–149.
43. Manzhos, S.; Yamashita, K.; Carrington, T. Fitting sparse multidimensional data with low-dimensional terms. *Comput. Phys. Commun.* **2009**, *180*, 2002–2012. [CrossRef]
44. Golub, P.; Manzhos, S. Kinetic energy densities based on the fourth order gradient expansion: Performance in different classes of materials and improvement via machine learning. *Phys. Chem. Chem. Phys.* **2019**, *21*, 378–395. [CrossRef]
45. Mi, W.; Luo, K.; Trickey, S.B.; Pavanello, M. Orbital-free density functional theory: An attractive electronic structure method for large-scale first-principles simulations. *Chem. Rev.* **2023**, *123*, 12039–12104. [CrossRef] [PubMed]
46. Manzhos, S.; Carrington, T., Jr. A random-sampling high dimensional model representation neural network for building potential energy surfaces. *J. Chem. Phys.* **2006**, *125*, 084109. [CrossRef]
47. Majumder, M.; Hegger, S.E.; Dawes, R.; Manzhos, S.; Wang, X.-G.; Tucker, C., Jr.; Li, J.; Guo, H. Explicitly correlated mrci-f12 potential energy surfaces for methane fit with several permutation invariant schemes and full-dimensional vibrational calculations. *Mol. Phys.* **2015**, *113*, 1823–1833. [CrossRef]
48. Castro, E.; Avila, G.; Manzhos, S.; Agarwal, J.; Schaefer, H.F.; Carrington, T., Jr. Applying a smolyak collocation method to cl2co. *Mol. Phys.* **2017**, *115*, 1775–1785. [CrossRef]
49. Kamath, A.; Vargas-Hernández, R.A.; Krems, R.V.; Carrington, T., Jr.; Manzhos, S. Neural networks vs gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy. *J. Chem. Phys.* **2018**, *148*, 241702. [CrossRef] [PubMed]
50. Manzhos, S.; Wang, X.; Dawes, R.; Carrington, T. A nested molecule-independent neural network approach for high-quality potential fits. *J. Phys. Chem. A* **2006**, *110*, 5295–5304. [CrossRef] [PubMed]
51. Manzhos, S.; Carrington, T., Jr. Neural network potential energy surfaces for small molecules and reactions. *Chem. Rev.* **2021**, *121*, 10187–10217. [CrossRef] [PubMed]
52. Manzhos, S.; Dawes, R.; Carrington, T. Neural network-based approaches for building high dimensional and quantum dynamics-friendly potential energy surfaces. *Int. J. Quantum Chem.* **2015**, *115*, 1012–1020. [CrossRef]
53. Jiang, B.; Guo, H. Permutation invariant polynomial neural network approach to fitting potential energy surfaces. *J. Chem. Phys.* **2013**, *139*, 054112. [CrossRef]
54. Nandi, A.; Qu, C.; Houston, P.L.; Conte, R.; Bowman, J.M. Δ-machine learning for potential energy surfaces: A pip approach to bring a dft-based pes to ccsd(t) level of theory. *J. Chem. Phys.* **2021**, *154*, 051102. [CrossRef]
55. Wang, Y.; Guan, Y.; Guo, H.; Yarkony, D.R. Enabling complete multichannel nonadiabatic dynamics: A global representation of the two-channel coupled, 1,21a and 13a states of nh3 using neural networks. *J. Chem. Phys.* **2021**, *154*, 094121. [CrossRef]
56. Manzhos, S.; Carrington, T., Jr. Using neural networks to represent potential surfaces as sums of products. *J. Chem. Phys.* **2006**, *125*, 194105. [CrossRef] [PubMed]
57. Manzhos, S.; Ihara, M. Neural network with optimal neuron activation functions based on additive gaussian process regression. *J. Phys. Chem. A* **2023**, *127*, 7823–7835. [CrossRef] [PubMed]
58. Manzhos, S.; Ihara, M. Orders-of-coupling representation achieved with a single neural network with optimal neuron activation functions and without nonlinear parameter optimization. *Artif. Intell. Chem.* **2023**, *1*, 100013. [CrossRef]
59. Zhang, Y.; Xu, W.; Liu, G.; Zhang, Z.; Zhu, J.; Li, M. Bandgap prediction of two-dimensional materials using machine learning. *PLoS ONE* **2021**, *16*, e0255637. [CrossRef]
60. Abarbanel, O.D.; Hutchison, G.R. Machine learning to accelerate screening for marcus reorganization energies. *J. Chem. Phys.* **2021**, *155*, 054106. [CrossRef]
61. Wahab, A.; Gershoni-Poranne, R. COMPAS-3: A dataset of peri -condensed polybenzenoid hydrocarbons. *Phys. Chem. Chem. Phys.* **2024**, *26*, 15344–15357. [CrossRef]
62. Haastrup, S.; Strange, M.; Pandey, M.; Deilmann, T.; Schmidt, P.S.; Hinsche, N.F.; Gjerding, M.N.; Torelli, D.; Larsen, P.M.; Riis-Jensen, A.C.; et al. The computational 2d materials database: High-throughput modeling and discovery of atomically thin crystals. *2D Mater.* **2018**, *5*, 042002. [CrossRef]
63. Kohn, W.; Sham, L.J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965**, *140*, A1133–A1138. [CrossRef]
64. Perdew, J.P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868. [CrossRef]
65. Becke, A.D. Density-functional thermochemistry. iii. the role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652. [CrossRef]
66. Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754. [CrossRef]
67. RDKit: Open-Source Cheminformatics Software | Bibsonomy. Available online: https://www.bibsonomy.org/bibtex/ee9a4ddeff3121aa622cf35709fa6e21 (accessed on 22 November 2024).

68. Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xtb—An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652–1671. [CrossRef]
69. Yanai, T.; Tew, D.P.; Handy, N.C. A new hybrid exchange–correlation functional using the coulomb-attenuating method (cam-b3lyp). *Chem. Phys. Lett.* **2004**, *393*, 51–57. [CrossRef]
70. Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O.A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **2012**, *108*, 058301. [CrossRef]