Transfer Learning for Molecular Property Predictions from Small Data Sets

Thorren Kirschbaum[†] and Annika Bande[*,†,‡]

[†]*Theory of Electron Dynamics and Spectroscopy, Helmholtz-Zentrum Berlin für Materialien und Energie GmbH, Hahn-Meitner-Platz 1, 14109 Berlin, Germany*

[‡]*Institute of Inorganic Chemistry, Leibniz University Hannover, Callinstr. 9, 30167 Hannover, Germany*

E-mail: annika.bande@helmholtz-berlin.de

Abstract

Machine learning has emerged as a new tool in chemistry to bypass expensive experiments or quantum-chemical calculations, for example, in high-throughput screening applications. However, many machine learning studies rely on small data sets, making it difficult to efficiently implement powerful deep learning architectures such as message passing neural networks. In this study, we benchmark common machine learning models for the prediction of molecular properties on small data sets, for which the best results are obtained with the message passing neural network PaiNN, as well as SOAP molecular descriptors concatenated to a set of simple molecular descriptors tailored to gradient boosting with regression trees. To further improve the predictive capabilities of PaiNN, we present a transfer learning strategy that uses large data sets to pre-train the respective models and allows to obtain more accurate models after fine-tuning on the original data sets. The pre-training labels are obtained from computationally cheap ab initio or semi-empirical models and corrected by simple linear regression on the target data set to obtain labels that are close to those of the original data. This strategy is tested on the Harvard Oxford Photovoltaics data set (HOPV, HOMO-LUMO-gaps), for which excellent results are obtained, and on the Freesolv data set (solvation energies), where this method is unsuccessful due to a complex underlying learning task and the dissimilar methods used to obtain pre-training and fine-tuning labels. Finally, we find that the final training results do not improve monotonically with the size of the pre-training data set, but pre-training with fewer data points can lead to more biased pre-trained models and higher accuracy after fine-tuning.

Introduction

In the realm of molecular sciences, predicting the properties of molecules stands as a pivotal pursuit across numerous domains, including pharmaceuticals, materials science, and environmental studies. Traditionally, this endeavor has been tackled by quantum chemistry, relying on sophisticated theoretical frameworks and computational simulations to unravel the intricate behaviors of molecules. However, the advent of machine learning (ML) has ushered in a paradigm shift, offering an alternative approach that leverages data-driven methodologies efficiently compute molecular properties.[1–5]

One notable application of ML in this domain is its integration into high throughput screening workflows, where it replaces conventional quantum chemistry methods.[3] Through the judicious application of ML algorithms, researchers can expedite the process of identifying promising molecules for drug discovery or materials development, thereby accelerating the pace of scientific innovation. However, the powerful deep neural networks (NNs) often used in such approaches usually need large amounts of training data to finally make accurate predictions.

Machine learning algorithms, particularly deep NNs, have exhibited remarkable success in discerning intricate patterns and relationships within data sets of substantial size. In recent developments, the message passing neural network PaiNN has demonstrated highest performance in predicting molecular properties, achieving state-of-the-art results across various datasets.[6,7] Another prevalent approach in ML for molecular sciences entails the utilization of tailored molecular descriptors, such as SOAP,[8,9] in conjunction with traditional ML models like kernel ridge regression (KRR). In general, as a rule of thumb, deep learning architectures are to be preferred when $> 10^3$–10^4 training points are available, and different approaches, such as KRR, should be considered otherwise.[10] Yet, in the context of molecular property prediction, data sets often fall short regarding the typically required large amounts of data for deep NNs. This mismatch often presents a formidable obstacle, impeding the efficacy of ML in scenarios characterized by small data sets, which are common in practical applications.[11–13] Bridging this divide necessitates innovative strategies and methodologies tailored to the unique challenges posed by small data sets. In this paper, we present a transfer learning strategy to address this pressing need.

In transfer learning, the knowledge learned in one setting is used to improve performance on a related target task.[14] In a first step, known as pre-training, an NN is trained on a large set of data that is closely related to the target task, yielding an NN with pre-optimized parameters. In the second step, termed fine-tuning, the same model is re-trained using data of the target task. Now, since the model is already close to the optimal set of parameters after the pre-training, much less data is needed during fine-tuning to obtain an accurate model for the target task. This strategy has proven useful in various fields, including image classification[15] and captioning,[16] gaming strategies,[17] and social network analysis.[18] In the physical sciences, some previous studies have used transfer learning to improve property predictions of molecules and materials.[19,20] Fur such tasks, Sun *et al.* found that supervised pre-training generally helps, if the pre-training labels are closely aligned with the downstream tasks, but the success of efficient transfer learning strongly depends on the employed data sets, ML models, and their hyperparameters.[21] Moreover, a range of studies recently investigated the use of transfer learning to improve atomic force predictions for molecular dynamics simulations.[22–29]

In this study, we start by undertaking a rigorous benchmarking, evaluating standard ML techniques in the context of small molecular data sets. In particular, we compare the use of SOAP descriptors tailored to a standard ML algorithm (KRR, NN, or gradient boosting) to the message passing neural network PaiNN for regression. The comparison is done both on the Harvard Organic Photovoltaics (HOPV) data set,[30] containing photovoltaic properties of 350 organic molecules, and on the Freesolv data set,[31] which contains solvation energies of 643 small organic molecules.

Secondly, we propose a transfer learning strategy as a means of augmenting predictive modeling with small data sets. It involves leveraging computations from computationally cheap ab initio or semi-empirical models coupled with simple linear regression, to generate pre-training labels for PaiNN.

Methods

In this study, we use two small common physical chemistry data sets, the Harvard Organic Photovoltaics (HOPV) data set[30] and the Freesolv data set.[31] The HOPV data set contains 350 molecules and their photovoltaic data obtained from both experiments and quantum-chemical calculations. In this study, we focus on the HOMO-LUMO gaps obtained from PBE0/def2-SVP density functional theory, which is a common *ab initio* method used for standard chemistry data sets.[7,32–37] Note that the version of the HOPV data set used by us (HOPV_15_revised_2) contains a single molecule with an Se atom, which we regarded as a structural outlier and therefore removed from the data set prior to training. The Freesolv data set contains experimental and calculated solvation energies of 643 small molecules,[31] from which we use the values obtained from "alchemical" calculations based on molecular dynamics simulations by the Mobley group.[38]

For benchmarking standard ML methods with HOPV and Freesolv, we employed the PaiNN implementation of the *schnetpack* Python library,[39] as well as the SOAP implementation of the *dscribe* library[40] in junction with either *sklearn*'s GradientBoostingRegressor (GBoost), kernel ridge regression, or a simple feed-forward NN. The GBoost algorithm profited from an additional set of simple molecular descriptors concatenated to the SOAP descriptor. For NN learning, to limit the size of the NNs, the sparse SOAP descriptors were compressed by principal component analysis, retaining 99.999 % of the variance.[7,9] The hyperparameters of PaiNN were optimized using an iterative grid search, and the hyperparameters of the SOAP-ML models were optimized using Bayesian optimization. For PaiNN, a learning rate decay and early stopping were employed, and results were averaged over five runs. The final model hyperparameters are listed in the appendix.

In the second part of this study, we present a method to improve the predictions of the message passing neural network PaiNN,[6] which was also identified as the best ML model for the data sets at hand (see Results and Discussion), via transfer learning. For this, we need (i) a (small) molecular data set with high-accuracy data of the target property (y_{true}), (ii) a large data set of molecular structures to be used for pre-training, and (iii) a computationally inexpensive method from the field of quantum chemistry to compute the target property for the pre-training data. Cheap ab initio methods, like density functional theory in the linear density approximation (LDA-DFT), or semi-empirical methods typically yield results which are qualitatively correct, but quantitatively far off.[41] To correct the results obtained from these methods, we use the respective quantum-chemical (QC) method to compute the target property on the original high-quality data set (i), y_{QC}, and apply simple linear regression to obtain the coefficients a and b that optimize the equation $(a \cdot y_{\mathrm{QC}} + b) - y_{\mathrm{true}} = 0$ for this data set. Subsequently, we use the same QC method to compute the target property for the structures of the pre-training data set (ii), again y_{QC}, and finally obtain the pre-training properties as $y_{\mathrm{pt}} = a \cdot y_{\mathrm{QC}} + b$, thought to be similar to that of the methods used in (i).

In this study, we use the molecular structures of the OE62 data set as pre-training structures for the HOPV data set, and the molecular structures of the QM9 data set as pre-training structures for Freesolv (ii). The OE62 data set contains crystal structures of 62 k medium-sized organic molecules,[37] and the QM9 data set contains DFT-optimized structures of 134 k small molecules.[42] To obtain pre-training labels for these data sets (iii), HOMO-LUMO gaps and solvation energies were calculated with LDA-DFT using the ORCA software,[43] and tight-binding density functional theory (DFTB) using the GFN2-xTB method[44] implemented in the XTB software.[45] Solvation energies were obtained as the difference in single point energy of the molecule in vacuum and the molecule with an implicit solvent model for water (conductor-like polarizable continuum model, CPCM in ORCA,[46] and generalized Born model with surface area contributions, GBSA in XTB,[47–49] respectively). In Figure 1, we display the scatter plots of the original HOPV and Freesolv data, respectively, against the data obtained from LDA and XTB, respectively, corrected by linear regression. These plots clearly show that the obtained HOMO-LUMO gaps are very close to the original HOPV data, but the solvation energies are quite dissimilar to the original Freesolv data. Furthermore, for both data sets, the properties obtained from LDA are closer to the original data those obtained from XTB. The mean absolute errors of the fits are: 0.05 eV (HOPV LDA), 0.12 eV (HOPV XTB), 1.1 kcal/mol (Freesolv LDA), 1.7 kcal/mol (Freesolv XTB).

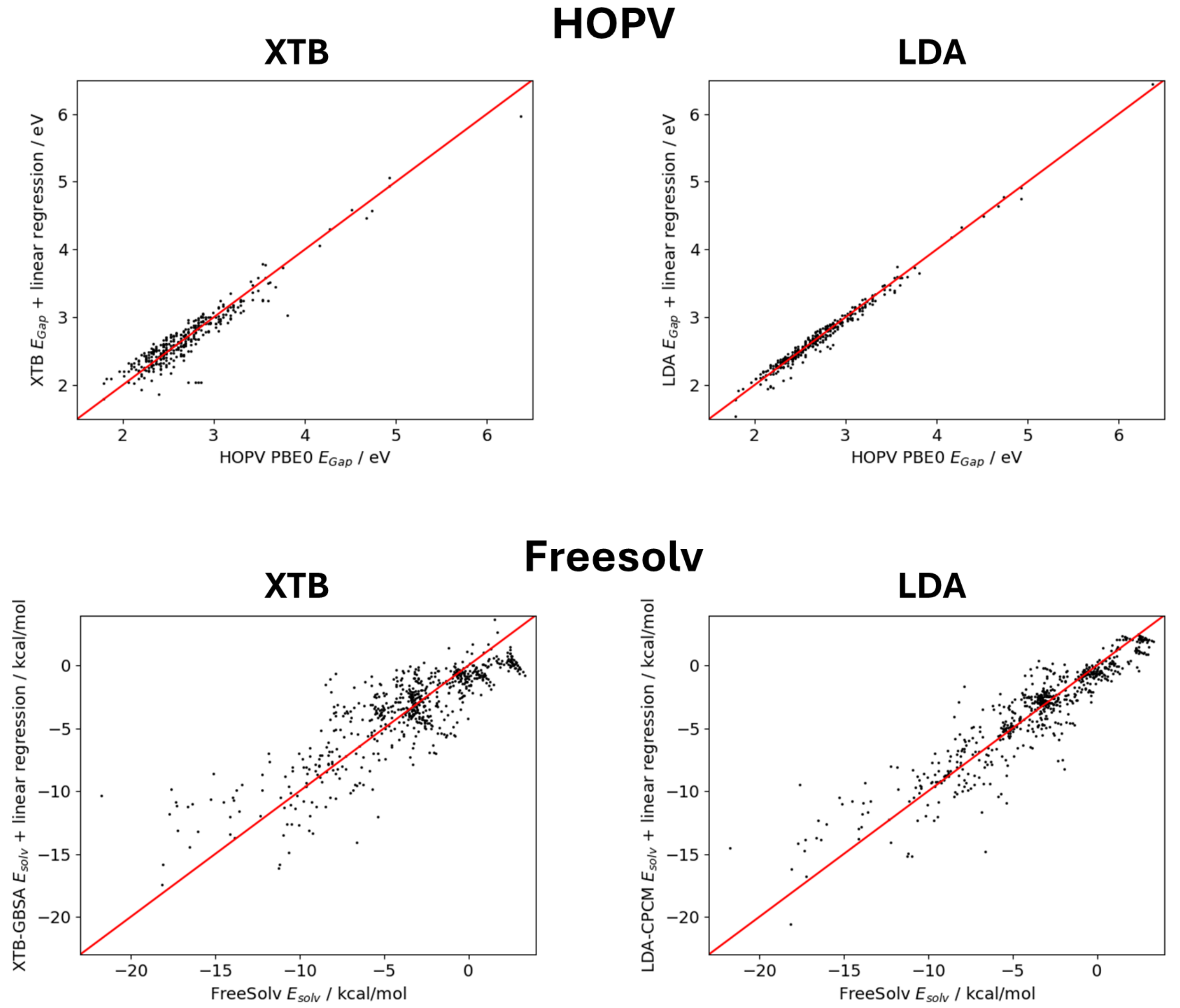


Figure 1: Scatter plots of the original data plotted against data obtained from XTB (left) or LDA (right) plus linear regression for the data sets HOPV (HOMO-LUMO-gaps, top) and Freesolv (solvation energies, bottom).

Furthermore, we also GBoost trained on HOPV and Freesolv, respectively, to compute HOMO-LUMO gaps for OE62 structures and solvation energies for QM9 structures, respectively, and used these for pre-training PaiNN. The results are then compared to the pre-training described above, in which the pre-training labels are obtained from cheap quantum-chemical methods.

On average, HOMO-LUMO gap computations took approximately 106 s (LDA-DFT), 0.22 s (XTB), and < 0.1 s (GBoost) per data point in OE62, respectively. Solvation energy computations took approximately 70 s (LDA-DFT), 0.15 s (XTB), and < 0.1 s (GBoost) per data point in QM9, respectively, and 1.33 s (XTB) per data point in OE62.

Pre-training on QM9 data was done for 200 epochs (100 k training, 10 k validation, 20 k test data points). Pre-training on OE62 data was done for 250 epochs for LDA and XTB data (50 k training, 5 k validation, 6 k test data points), and for 300 epochs for GBoost data (35 k training, 5 k validation, 5537 testing data points). For pre-training with GBoost data, the pre-training data containing elements which did not occur in the fine-tuning data set were omitted to keep the employed SOAP descriptors reasonably small and therefore minimize memory requirements. Finally, pre-training was done using 35 k training data points (SOAP+SD+GBoost/OE62 for HOPV, and SOAP+SD+GBoost/QM9 for Freesolv) or 30 k data points (SOAP+SD+GBoost/OE62 for Freesolv), respectively. For each method, three models were trained with different random seeds and the best model as evaluated on the test subset was selected for fine-tuning.

Results and Discussion

ML benchmarks

We start by benchmarking standard chemistry ML models for regression on the two small, but high-quality data sets under investigation, HOPV[30] and Freesolv,[31] listed under (i) in the Methods section. The results are summarized in Table 1 as mean absolute errors (MAEs, root mean squared errors (RMSE) are given in parenthesis). For both data sets, very low errors are obtained with PaiNN, yielding MAEs of 0.20 eV for HOPV and 0.56 kcal/mol for Freesolv, respectively. For HOPV, the SOAP+SD+GBoost model yields even slightly better results with an MAE of 0.19 eV, while 0.86 kcal/mol are obtained for regression on Freesolv. The results obtained with these algorithms are qualitatively similar to those obtained in previous ML studies on HOPV[50–56] and Freesolv.[56–67]

Table 1: Mean absolute errors for HOPV HOMO-LUMO gaps (in eV) and Freesolv solvation energies (in kcal/mol) from four different models. The first three models use SOAP molecular descriptors connected to either KRR, a dense NN, or GBoost. The GBoost model uses an additional set of simple molecular descriptors (SD) concatenated with the SOAP descriptors. The final model is the message passing NN PaiNN. Uncertainties are measured in RMSE (in parentheses). Best MAE results are bold.

Model	**HOPV** **MAE (RMSE) / eV**	**Freesolv** **MAE (RMSE) / kcal/mol**
SOAP+KRR	0.31 (0.46)	1.7 (2.2)
SOAP+NN	0.25 (0.33)	2.1 (3.2)
SOAP+SD+GBoost	**0.19 (0.30)**	0.86 (1.30)
PaiNN	**0.20 (0.31)**	**0.56 (0.86)**

The remaining models SOAP+NN and SOAP+KRR perform considerably worse on both data sets, in direct comparison SOAP+NN yielding better results for HOPV while SOAP+KRR yields better results for Freesolv. We also note that the MAEs from the different models are much closer for regression on HOPV, where the worst model (SOAP+KRR) has a 63 % higher MAE than the best model (SOAP+SD+GBoost). In contrast, for Freesolv, the worst model (SOAP+NN) has a 375 % higher MAE than the best model (PaiNN), indicating that the learning task is much more complex than for HOPV (cf. also Figure 1.

For the following transfer learning with PaiNN, the best remaining model was selected to generate the respective pre-training labels, which is SOAP+SD+GBoost for both data sets.

Transfer Learning

We use both cheap models from quantum chemistry (LDA-DFT and XTB-DFTB) corrected by simple linear regression, and an ML regressor (SOAP+SD+GBoost, from here on named GBoost) to obtain labels for pre-training the PaiNN models (details in the Methods section). For HOPV, consisting of medium-sized organic molecules, we use the structures of the OE62 data set for pre-training, which contains ca. 62 k diverse structures of medium-sized organic molecules. For Freesolv, which on the other hand consists of small organic molecules, we use the structures of the QM9 data set for pre-training, which contains ca. 132 k organic molecules with up to nine heavy (non-H) atoms. Tables 2 and 3 summarize the results obtained for regression on the HOPV and Freesolv data set, respectively, after PaiNN training from scratch (cf. Table 1), or after pre-training on data labeled with either GBoost, XTB, or LDA, respectively. The learning curves displaying the MAE vs. the number of fine-tuning training points are shown in Figure 2 (here, pre-training is done on the full respective pre-training data set), and the learning curves displaying the MAE vs. the number of pre-training points are shown in Figure 3 (here, fine-tuning is done on the full respective fine-tuning data set), each for the HOPV set on top and the Freesolv set in the bottom.

Table 2: MAEs for HOPV HOMO-LUMO gaps (in eV) from PaiNN obtained after training from scratch, or after pre-training on OE62 data with labels obtained from GBoost, XTB or LDA, respectively. Uncertainties are measured in RMSE (in parentheses). Best MAE results are bold.

Pre-training	**HOPV** **MAE (RMSE) / eV**
None	0.20 (0.31)
OE62+GBoost	0.18 (0.25)
OE62+XTB	**0.13 (0.17)**
OE62+LDA	**0.13 (0.19)**

Table 3: MAEs for Freesolv solvation energies (in kcal/mol) from PaiNN obtained after training from scratch, or after pre-training on OE62 data with labels obtained from GBoost, XTB or LDA, respectively. Uncertainties are measured in RMSE (in parentheses). Best MAE results are bold.

Pre-training	**Freesolv** **MAE (RMSE) / kcal/mol**
None	**0.56 (0.86)**
QM9+GBoost	0.74 (1.19)
QM9+XTB	0.60 (0.99)
QM9+LDA	0.64 (1.00)

For HOPV (Table 2), pre-training results in an improved learning performance. The MAE is reduced from 0.20 eV (training from scratch) to 0.18 eV after pre-training on OE62+GBoost, and further to 0.13 eV after pre-training on OE62+XTB or OE62+LDA. The pre-training with GBoost is less effective, probably due to the fact that for this method, some training points were removed from OE62 to avoid creating overly large SOAP descriptors for the ML model. However, the GBoost model might also produce less accurate labels than the QC models corrected by linear regression, since the GBoost MAE (0.19 eV, Table 1) is higher than the MAEs of the LDA or XTB plus linear regression fit on HOPV (MAEs of 0.05 eV for LDA, and 0.12 eV for XTB, respectively).

The fine-tuning learning curves for HOPV (Figure 2, top) show that this improvement of ca. 0.07 eV over training from scratch (black curve) after pre-training on OE62+XTB (light blue) or OE62+LDA (dark blue) is relatively stable, independent of the number of HOPV fine-tuning training points. Only, after pre-training on OE62+GBoost, the improvement obtained from pre-training is much higher when fine-tuning on increasingly many HOPV data points (red curve).

The HOPV pre-training learning curves (Figure 3, top) confirm that for all pre-training data set sizes tested, the labels obtained from the QC models lead to a higher accuracy as compared to the ML-generated labels. Interestingly, for GBoost and LDA labels, increasing the size of the pre-training data set does not monotonically improve the final training results. Instead, there appears to be an optimal training set size at around 20 k pre-training points

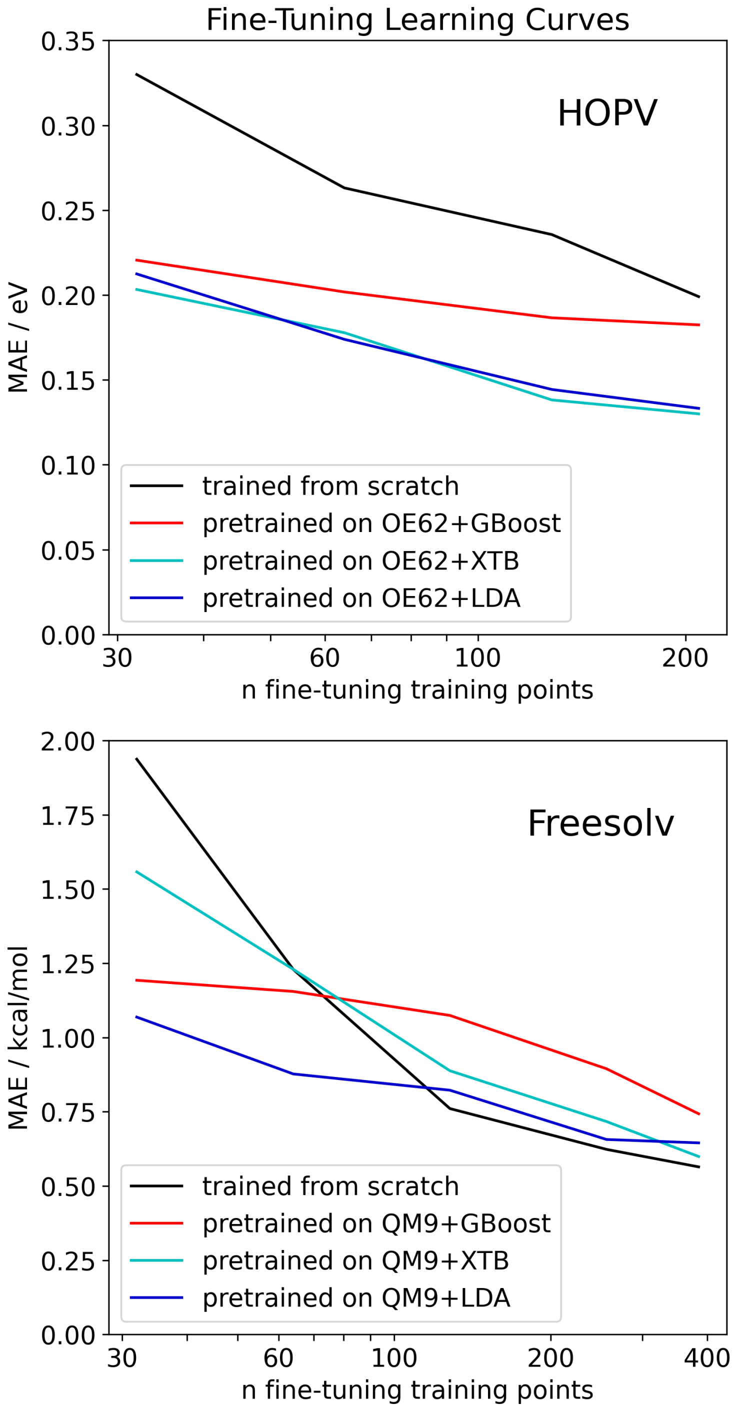


Figure 2: Fine-tuning learning curves for PaiNN training on HOPV (top) and Freesolv (bottom), after training from scratch (black) or after pre-training on OE62 or QM9 data, respectively, with labels obtained from GBoost (red), XTB (light blue) or LDA-DFT (dark blue). The MAE is plotted against the number of training examples used for fine-tuning (log scale).

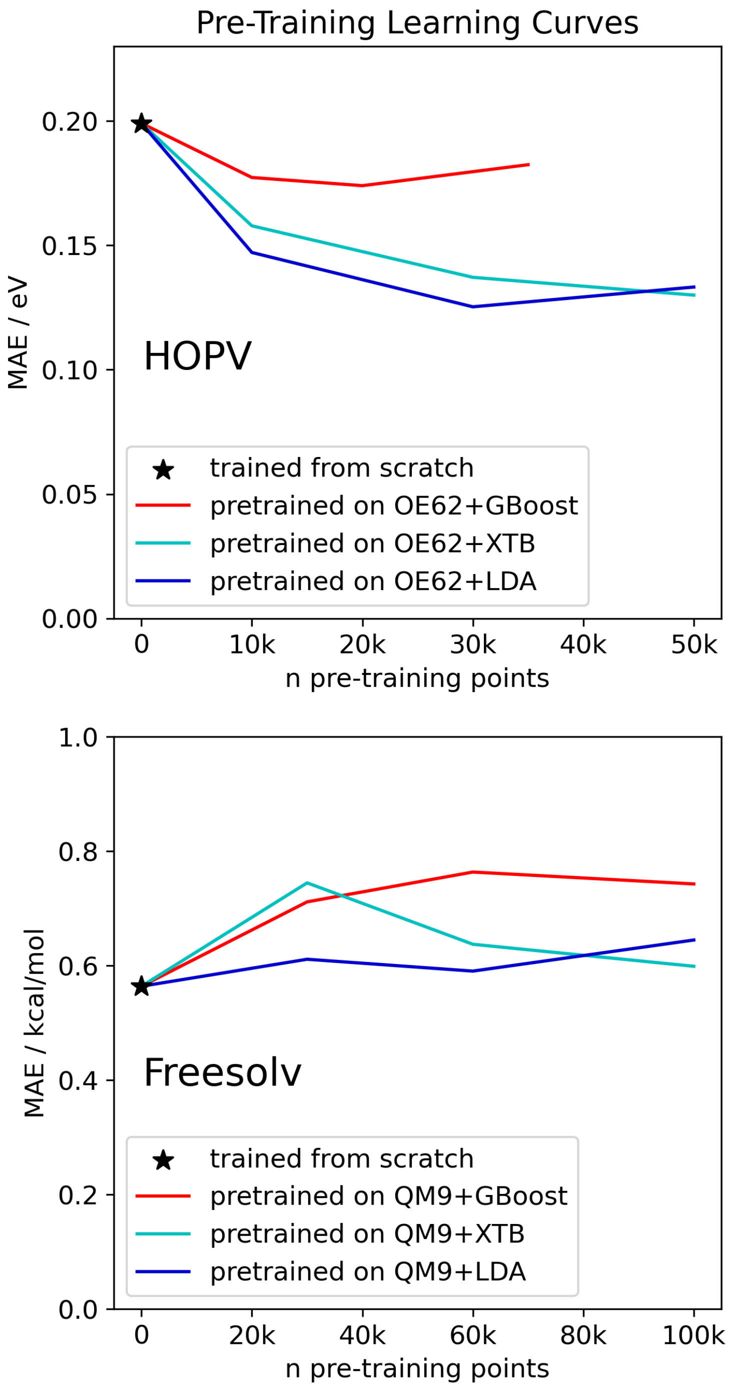


Figure 3: Pre-training learning curves for PaiNN training on HOPV (top) and Freesolv (bottom), after training from scratch (black star) or after pre-training on OE62 or QM9 data, respectively, with labels obtained from GBoost (red), XTB (light blue) or LDA-DFT (dark blue). The mean absolute error (MAE) is plotted against the number of training examples used for pre-training.

for GBoost-labeled data and 30 k pre-training points for LDA-labeled data, respectively, after which further improving the pre-training data set size results in higher MAEs after fine-tuning. For XTB-labeled data, a similar optimal pre-training data set size may exist outside the range of the data available here. This indicates that a pre-trained model that is too strongly fitted on the pre-training task may be worse for transfer learning than a more biased model that only learned to grasp high-level concepts from a limited amount of training data.

Regression learning on Freesolv, on the other hand, is not improved by pre-training the PaiNN models (Table 3). Instead, the final training MAEs are increased from 0.56 kcal/mol (training from scratch) to 0.74 kcal/mol (GBoost), 0.60 kcal/mol (XTB), and 0.64 kcal/mol (LDA). The fine-tuning learning curves (Figure 2, bottom) show that an improvement is achieved when fine-tuning is done with very few ($\leq$ ca. 64) data points, QM9+LDA fine-tuning resulting in the largest improvement. However, for larger fine-tuning data set sizes, the results obtained after QM9+XTB or QM9+LDA pre-training are slightly worse as compared to training from scratch, and pre-training with QM9+GBoost results in the worst performance. Furthermore, the pre-training learning curves (Figure 3, bottom) indicate that the loss of performance is approximately independent from the size of the pre-training data set. Moreover, for pre-training data obtained from GBoost and LDA, the final models tend to even decrease in quality after being pre-trained on larger data sets.

As discussed before, the regression task on Freesolv might be more complex as compared to regression on HOPV. As a consequence, the pre-training labels generated by GBoost are less reliable for Freesolv solvation energies than those generated for HOPV HOMO-LUMO gaps (cf. Table 1). Additionally, the methods used for producing the respective pre-training labels (QC methods) are quite similar to the method used for generating HOPV data (standard DFT),[30] but quite dissimilar to the method used for generating Freesolv data (”alchemical” calculations based on molecular dynamics simulations, see also the scatter plots in Figure 1).[38] These factors may be the reason for why the employed transfer learning strategy does work very well for use with the HOPV data set, but not for Freesolv.

Another hypothesis for why transfer learning might fail in certain scenarios is that the underlying regression task is too easy, and therefore no additional pre-training is needed.[21] However, previous results for regression on molecular energies on the QM9 data set (free energy G, internal energy U, enthalpy H) using PaiNN indicated that MAEs as low as ca. 0.14 kcal/mol can be obtained on a sufficiently large data set, which is considerably lower than the MAEs obtained here.[6] Also, these quantities (G, U, H) can easily be calculated from only the respective molecular structure, but solvation energy computations additionally need to take into account the shape and intermolecular interactions of the respective solvation shell, thus adding another level of complexity. Therefore, we anticipate that the regression task on Freesolv is not too easy for transfer learning.

In an attempt to obtain pre-trained models that are more robust and therefore allegedly more suitable for transfer learning, we added additional data points to the respective pre-training data sets. For this, we used XTB to generate additional labels for QM9 (HOMO-LUMO gaps) and OE62 (solvation energies). However, we found that pre-training on a joint OE62 and QM9 data set did not improve the final fine-tuning results as compared to pre-training on OE62 or QM9 data only, respectively, for both HOPV and Freesolv training. This is surprising, because NNs usually get more accurate when increasing the amount of training data, and new dissimilar data points can help to create more robust models. Here, instead, the introduction of more data points that are dissimilar to the data contained in the final fine-tuning data set apparently impedes the learning process. In particular, for Freesolv, the underlying problem of having a complex learning task and dissimilar methods for generating pre-training and fine-tuning labels could not be solved by augmenting the pre-training data set. This observation is in line with previous work on transfer learning in chemistry, which stated that the pre-training and fine-tuning data sets should be as closely aligned as possible to obtain the best results.[19]

Following this idea, we aimed to instead make the pre-training and fine-tuning data sets more similar to each other: The HOPV data set consists almost entirely of S-functionalized aromatic molecules, while the OE62 data set used for pre-training contains a much more diverse set of molecules, including more structural variety and a wider range of atomic species. Accordingly, we filtered all molecules from the OE62 pre-training data set containing any of the elements As, Se, Br, Te or I, and furthermore removed all molecules that do not contain any sulfur atoms, thereby reducing the data set size from 61,489 to 41,487 molecules. Pre-training with 20 k less molecules from the filtered OE62 data set resulted in a slight loss of performance compared to pre-training of the full OE62+XTB data set (+0.01 eV MAE), which is approximately the same result as obtained after non-selective filtering of the pre-training data set (+0.02 eV MAE, cf. Figure 3). This indicates that increased similarity between pre-training and fine-tuning structures might slightly improve the learning performance by yielding more tailored pre-trained models, but the effect found here is very small.

Finally, previous works have used discriminative fine-tuning to further optimize transfer learning.[29] In this method, the learning rate during fine-tuning is adapted layer-wise, such that the first layers have a much smaller learning rate than the final layers of the model. The idea is that the first layers, which learn to grasp fundamental features of the molecular data, will be sufficiently trained during pre-training, while the last layers, which derive the final predictions, will need more fine-tuning. We implemented discriminative fine-tuning for PaiNN, following the approach outlined by Zaverkin *et al.*[29] Here, the learning rate (lr) for layer n−1 was adjusted as $\mathrm{lr}_{n-1} = \mathrm{lr}_n/5$ with lr = 5e-4 being the learning rate of the last layer. However, we found that this method did not yield any improvements over non-discriminative fine-tuning for transfer learning with PaiNN. In contrast, we found that in general, modifications of the learning rate and learning rate decay had a significant impact on the learning performance, and sticking to the parameters identified by the initial hyperparameter sweep continued to yield the best results.

Conclusions

This study investigated the use of common ML algorithms for use with small data sets, focusing on HOPV (350 data points) and Freesolv (643 data points). We found that both the MPNN PaiNN, and gradient boosting with regression trees operating on SOAP molecular descriptors concatenated to a set of simple molecular descriptors yield accurate results, clearly outperforming the SOAP+NN and SOAP+KRR architectures.

Furthermore, we presented a method for data-efficient ML on small physical chemistry data sets using transfer learning. The pre-training data was obtained from computationally cheapQC calculations corrected by simple linear regression calibrated on the fine-tuning data set. This method utilizes the fact that many cheap ab initio or semi-empirical techniques yield quantitatively incorrect but qualitatively correct data, which can easily be corrected via linear regression. For the HOPV data set, the introduced method outperforms transfer learning with labels obtained from an ML model. Here, pre-training with labels computed by either XTB or LDA result in very similar learning performances, but XTB computations were much faster than LDA computations. Accordingly, labelling the pre-training data with XTB should be preferred.

However, the same method did not yield improved learning results for regression on the Freesolv data set. This was due to the fact that the underlying learning task is much more complex and the data used for pre-training is too dissimilar to the data contained in Freesolv.

Furthermore, the ML results indicate that pre-training with a limited amount of data points may be beneficial, since the obtained models are slightly more biased and less strongly fitted to the pre-training task. Furthermore, we found that augmenting the pre-training data sets with additional structures that are less similar to the fine-tuning data does not improve the learning performance. Finally, for the models trained here, sticking to the hyperparameters identified by the initial hyperparameter sweeps was more efficient than adapting the learning rates as done in discriminative fine-tuning.

In conclusion, for chemistry ML tasks that use small data sets, an effective transfer learning strategy should use pre-training on molecules that are similar to the molecules of the downstream task, a labelling method that is closely aligned to the method used for labelling the fine-tuning data, and an optimized training set size for the pre-training.

Appendix

Machine Learning Model Hyperparameters

We here provide details of the ML models used for regression on both HOPV and Freesolv (*cf.* Table 1).

The final SOAP+KRR models used the following hyperparameters:
HOPV: $r_{cut}^{SOAP} = 5.63$, $n_{max}^{SOAP} = 7$, $l_{max}^{SOAP} = 7$, $alpha_{KRR} = 1.12$, kernel: cosine.
Freesolv: $r_{cut}^{SOAP} = 9.53$, $n_{max}^{SOAP} = 8$, $l_{max}^{SOAP} = 8$, $alpha_{KRR} = 0.59$, kernel: = cosine.

In the SOAP+NN model, the neural network consists of densely connected layers with steadily decreasing numbers of neurons, including batch normalization and a dropout layer with $p = 0.3$ after the first layer. Here, 20 % of the training data was used for validation, and early stopping was employed. The final models used the following hyperparameters:
HOPV: $r_{cut}^{SOAP} = 7.22$, $n_{max}^{SOAP} = 8$, $l_{max}^{SOAP} = 8$, $n_{layers} = 2$, $lr = 0.00860$, nonlinearity: swish.
Freesolv: $r_{cut}^{SOAP} = 9.94$, $n_{max}^{SOAP} = 7$, $l_{max}^{SOAP} = 7$, $n_{layers} = 4$, $lr = 0.00627$, nonlin.: swish.

In the SOAP+SD+GBoost model, the following simple molecular descriptors were employed: total number of atoms, number of atoms for each atom type in the data set, one-hot-encoding for each atom type (present/not present in the respective molecule), mean and standard deviation of the carbon-carbon-distance of neighboring carbon atoms within the molecule. Using these descriptors with the KRR or NN did not yield any improvement for the respective learning algorithm. The final SOAP+SD+GBoost models used the following hyperparameters:
HOPV: $r_{cut}^{SOAP} = 9.71$, $n_{max}^{SOAP} = 5$, $l_{max}^{SOAP} = 8$, $lr_{GBoost} = 0.059$, $n\_estimators_{GBoost} = 295$, $max\_depth_{GBoost} = 4$.

Freesolv: $r_{cut}^{SOAP} = 6.85$, $n_{max}^{SOAP} = 4$, $l_{max}^{SOAP} = 4$, $lr_{GBoost} = 0.250$, $n\_estimators_{GBoost} = 158$, $max\_depth_{GBoost} = 4$.

The final PaiNN models used the following hyperparameters:

HOPV: $radial\_basis = GaussianRBF$, $r_{cut} = 5.0$, $n\_rbf = 20$, $n\_atom\_basis = 30$, $n\_interactions = 3$, $cutoff\_fn = CosineCutoff$, $lr = 5e-4$, $lr\_decay\_factor = 0.5$, $lr\_decay\_patience = 5$, $batchsize = 10$.

Freesolv: $radial\_basis = GaussianRBF$, $r_{cut} = 8.0$, $n\_rbf = 30$, $n\_atom\_basis = 200$, $n\_interactions = 3$, $cutoff\_fn = CosineCutoff$, $lr = 5e-4$, $lr\_decay\_factor = 0.5$, $lr\_decay\_patience = 5$, $batchsize = 10$.

Acknowledgement

TK and AB acknowledge support from the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS). Computing resources were kindly provided by the Freie Universität Berlin hpc cluster Curta[68] and by the Helmholtz-Zentrum Dresden-Rossendorf. The introduction was in part formulated with the help of ChatGPT 3.5.

References

(1) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; Anatole von Lilienfeld, O. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **2013**, *15*, 095003.

(2) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *J. Phys. Chem. Lett.* **2015**, *6*, 2326–2331.

(3) Dral, P. O. Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, *11*, 2336–2347.

(4) Fung, V.; Zhang, J.; Juarez, E.; Sumpter, B. G. Benchmarking graph neural networks for materials chemistry. *npj Comput. Mater.* **2021**, *7*, 1–8.

(5) Singh, K.; Münchmeyer, J.; Weber, L.; Leser, U.; Bande, A. Graph Neural Networks for Learning Molecular Excitation Spectra. *J. Chem. Theory Comput.* **2022**, *18*, 4408–4417.

(6) Schütt, K.; Unke, O.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. Proceedings of the 38th International Conference on Machine Learning. 2021; pp 9377–9388.

(7) Kirschbaum, T.; von Seggern, B.; Dzubiella, J.; Bande, A.; Noé, F. Machine Learning Frontier Orbital Energies of Nanodiamonds. *J. Chem. Theory Comput.* **2023**, *19*, 4461–4473.

(8) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B* **2013**, *87*, 184115.

(9) Parsaeifard, B.; De, D. S.; Christensen, A. S.; Faber, F. A.; Kocer, E.; De, S.; Behler, J.; Lilienfeld, O. A. v.; Goedecker, S. An assessment of the structural resolution of various fingerprints commonly used in machine learning. *Mach. Learn.: Sci. Technol.* **2021**, *2*, 015018.

(10) Unke, O. T.; Chmiela, S.; Sauceda, H. E.; Gastegger, M.; Poltavsky, I.; Schütt, K. T.; Tkatchenko, A.; Müller, K.-R. Machine Learning Force Fields. *Chem. Rev.* **2021**, *121*, 10142–10186.

(11) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quan-

tum Chemistry Approximations: The ?-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.

(12) Zhang, Y.; Ling, C. A strategy to apply machine learning to small datasets in materials science. *npj Comput. Mater.* **2018**, *4*, 1–8.

(13) Mazouin, B.; Schöpfer, A. A.; Lilienfeld, O. A. v. Selected machine learning of HOMO–LUMO gaps with improved data-efficiency. *Mater. Adv.* **2022**, 8306–8316.

(14) Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE* **2020**, *109*, 43–76.

(15) Saenko, K.; Kulis, B.; Fritz, M.; Darrell, T. Computer Vision-ECCV 2010, edited by K. Daniilidis, P. Maragos & N. Paragios, Vol. 6316 of Lecture Notes in Computer Science. 2010.

(16) Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. Proceedings of the IEEE conference on computer vision and pattern recognition. 2015; pp 3156–3164.

(17) Sharma, M.; Holmes, M.; Santamaria, J.; Irani, A.; Isbell, J. CL, & Ram, A.(2007). Transfer learning in real-time strategy games using hybrid CBR/RL. Proceedings of the Twentieth International Joint Conference on Artificial Intelligence. 2007; pp 1041–1046.

(18) Tang, J.; Lou, T.; Kleinberg, J.; Wu, S. Transfer learning to infer social ties across heterogeneous networks. *ACM Trans. Inf. Syst.* **2016**, *34*, 1–43.

(19) Li, H.; Zhao, X.; Li, S.; Wan, F.; Zhao, D.; Zeng, J. Improving molecular property prediction through a task similarity enhanced transfer learning strategy. *Iscience* **2022**, *25*, 105231.

(20) Shui, Z.; Karls, D.; Wen, M.; Tadmor, E.; Karypis, G.; others Injecting domain knowledge from empirical interatomic potentials to neural networks for predicting material properties. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 14839–14851.

(21) Sun, R.; Dai, H.; Yu, A. W. Does GNN Pretraining Help Molecular Representation? *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 12096–12109.

(22) Smith, J. S.; Nebgen, B. T.; Zubatyuk, R.; Lubbers, N.; Devereux, C.; Barros, K.; Tretiak, S.; Isayev, O.; Roitberg, A. E. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **2019**, *10*, 2903.

(23) Käser, S.; Koner, D.; Christensen, A. S.; von Lilienfeld, O. A.; Meuwly, M. *J. Phys. Chem. A* **2020**, *124*, 8853–8865.

(24) Zheng, P.; Zubatyuk, R.; Wu, W.; Isayev, O.; Dral, P. O. Artificial intelligence-enhanced quantum chemical method with broad applicability. *Nat. Commun.* **2021**, *12*, 7022.

(25) Käser, S.; Richardson, J. O.; Meuwly, M. Transfer learning for affordable and high-quality tunneling splittings from instanton calculations. *J. Chem. Theory Comput.* **2022**, *18*, 6840–6850.

(26) Gao, X.; Gao, W.; Xiao, W.; Wang, Z.; Wang, C.; Xiang, L. Supervised Pretraining for Molecular Force Fields and Properties Prediction. *arXiv* **2022**, arXiv:2211.14429.

(27) Zhang, D.; Bi, H.; Dai, F.-Z.; Jiang, W.; Zhang, L.; Wang, H. DPA-1: Pretraining of Attention-based Deep Potential Model for Molecular Simulation. *arXiv* **2022**, arXiv:2208.08236.

(28) Chen, M. S.; Lee, J.; Ye, H.-Z.; Berkelbach, T. C.; Reichman, D. R.; Markland, T. E. Data-efficient machine learning potentials from transfer learning of periodic correlated

electronic structure methods: Liquid water at AFQMC, CCSD, and CCSD (T) accuracy. *J. Chem. Theory Comput.* **2023**, *19*, 4510–4519.

(29) Zaverkin, V.; Holzmüller, D.; Bonfirraro, L.; Kästner, J. Transfer learning for chemically accurate interatomic neural network potentials. *Phys. Chem. Chem. Phys.* **2023**, *25*, 5383–5396.

(30) Lopez, S. A.; Pyzer-Knapp, E. O.; Simm, G. N.; Lutzow, T.; Li, K.; Seress, L. R.; Hachmann, J.; Aspuru-Guzik, A. The Harvard organic photovoltaic dataset. *Sci. Data* **2016**, *3*, 160086.

(31) Duarte Ramos Matos, G.; Kyu, D. Y.; Loeffler, H. H.; Chodera, J. D.; Shirts, M. R.; Mobley, D. L. Approaches for Calculating Solvation Free Energies and Enthalpies Demonstrated with an Update of the FreeSolv Database. *J. Chem. Eng. Data* **2017**, *62*, 1559–1569.

(32) Adamo, C.; Barone, V. Toward reliable density functional methods without adjustable parameters: The PBE0 model. *J. Chem. Phys.* **1999**, *110*, 6158–6170.

(33) Perdew, J. P.; Ernzerhof, M.; Burke, K. Rationale for mixing exact exchange with density functional approximations. *J. Chem. Phys.* **1996**, *105*, 9982–9985.

(34) Schäfer, A.; Horn, H.; Ahlrichs, R. Fully optimized contracted Gaussian basis sets for atoms Li to Kr. *J. Chem. Phys.* **1992**, *97*, 2571–2577.

(35) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297–3305.

(36) 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.

(37) Stuke, A.; Kunkel, C.; Golze, D.; Todorović, M.; Margraf, J. T.; Reuter, K.; Rinke, P.; Oberhofer, H. Atomic structures and orbital energies of 61,489 crystal-forming organic molecules. *Sci. Data* **2020**, *7*, 58.

(38) Mobley, D. L.; Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Des.* **2014**, *28*, 711–720.

(39) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K.-R. SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *J. Chem. Theory Comput.* **2019**, *15*, 448–455.

(40) Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. DScribe: Library of descriptors for machine learning in materials science. *Comput. Phys. Commun.* **2020**, *247*, 106949.

(41) Thiel, W. Semiempirical quantum–chemical methods. *WIREs Comput. Mol. Sci.* **2014**, *4*, 145–157.

(42) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.

(43) Neese, F.; Wennmohs, F.; Becker, U.; Riplinger, C. The ORCA quantum chemistry program package. *J. Chem. Phys.* **2020**, *152*, 224108.

(44) 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.

(45) Bannwarth, C.; Caldeweyher, E.; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.;

Spicher, S.; Grimme, S. Extended tight-binding quantum chemistry methods. *WIREs Comput. Mol. Sci.* **2021**, *11*, e1493.

(46) Barone, V.; Cossi, M. Quantum Calculation of Molecular Energies and Energy Gradients in Solution by a Conductor Solvent Model. *J. Phys. Chem. A* **1998**, *102*, 1995–2001.

(47) Onufriev, A.; Bashford, D.; Case, D. A. Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins* **2004**, *55*, 383–394.

(48) Sigalov, G.; Fenley, A.; Onufriev, A. Analytical electrostatics for biomolecules: Beyond the generalized Born approximation. *J. Chem. Phys.* **2006**, *124*, 124902.

(49) Lange, A. W.; Herbert, J. M. Improving Generalized Born Models by Exploiting Connections to Polarizable Continuum Models. I. An Improved Effective Coulomb Operator. *J. Chem. Theory Comput.* **2012**, *8*, 1999–2011.

(50) Paul, A.; Furmanchuk, A.; Liao, W.-k.; Choudhary, A.; Agrawal, A. Property Prediction of Organic Donor Molecules for Photovoltaic Applications Using Extremely Randomized Trees. *Mol. Inform.* **2019**, *38*, 1900038.

(51) Zhao, Z.-W.; del Cueto, M.; Geng, Y.; Troisi, A. Effect of Increasing the Descriptor Set on Machine Learning Prediction of Small Molecule-Based Organic Solar Cells. *Chem. Mater.* **2020**, *32*, 7777–7787.

(52) Munshi, J.; Chen, W.; Chien, T.; Balasubramanian, G. Transfer Learned Designer Polymers For Organic Solar Cells. *J. Chem. Inf. Model.* **2021**, *61*, 134–142.

(53) Eibeck, A.; Nurkowski, D.; Menon, A.; Bai, J.; Wu, J.; Zhou, L.; Mosbach, S.; Akroyd, J.; Kraft, M. Predicting Power Conversion Efficiency of Organic Photovoltaics: Models and Data Analysis. *ACS Omega* **2021**, *6*, 23764–23775.

(54) Fare, C.; Fenner, P.; Benatan, M.; Varsi, A.; Pyzer-Knapp, E. O. A multi-fidelity machine learning approach to high throughput materials screening. *npj Comput. Mater.* **2022**, *8*, 1–9.

(55) Moore, G. J.; Bardagot, O.; Banerji, N. Deep Transfer Learning: A Fast and Accurate Tool to Predict the Energy Levels of Donor Molecules for Organic Photovoltaics. *Adv. Theory Simul.* **2022**, *5*, 2100511.

(56) Li, H.; Zhao, X.; Li, S.; Wan, F.; Zhao, D.; Zeng, J. Improving molecular property prediction through a task similarity enhanced transfer learning strategy. *iScience* **2022**, *25*, 105231.

(57) Hutchinson, S. T.; Kobayashi, R. Solvent-Specific Featurization for Predicting Free Energies of Solvation through Machine Learning. *J. Chem. Inf. Model.* **2019**, *59*, 1338–1346.

(58) Scheen, J.; Wu, W.; Mey, A. S. J. S.; Tosco, P.; Mackey, M.; Michel, J. Hybrid Alchemical Free Energy/Machine-Learning Methodology for the Computation of Hydration Free Energies. *J. Chem. Inf. Model.* **2020**, *60*, 5331–5339.

(59) Weinreich, J.; Browning, N. J.; von Lilienfeld, O. A. Machine learning of free energies in chemical compound space using ensemble representations: Reaching experimental uncertainty for solvation. *J. Chem. Phys.* **2021**, *154*, 134113.

(60) Borgis, D.; Luukkonen, S.; Belloni, L.; Jeanmairet, G. Accurate prediction of hydration free energies and solvation structures using molecular density functional theory with a simple bridge functional. *J. Chem. Phys.* **2021**, *155*, 024117.

(61) Vermeire, F. H.; Green, W. H. Transfer learning for solvation free energies: From quantum chemistry to experiments. *Chem. Eng. J.* **2021**, *418*, 129307.

(62) Gao, P.; Yang, X.; Tang, Y.-H.; Zheng, M.; Andersen, A.; Murugesan, V.; Hollas, A.; Wang, W. Graphical Gaussian process regression model for aqueous solvation free energy prediction of organic molecules in redox flow batteries. *Phys. Chem. Chem. Phys.* **2021**, *23*, 24892–24904.

(63) Pathak, Y.; Mehta, S.; Priyakumar, U. D. Learning Atomic Interactions through Solvation Free Energy Prediction Using Graph Neural Networks. *J. Chem. Inf. Model.* **2021**, *61*, 689–698.

(64) Chung, Y.; Vermeire, F. H.; Wu, H.; Walker, P. J.; Abraham, M. H.; Green, W. H. Group Contribution and Machine Learning Approaches to Predict Abraham Solute Parameters, Solvation Free Energy, and Solvation Enthalpy. *J. Chem. Inf. Model.* **2022**, *62*, 433–446.

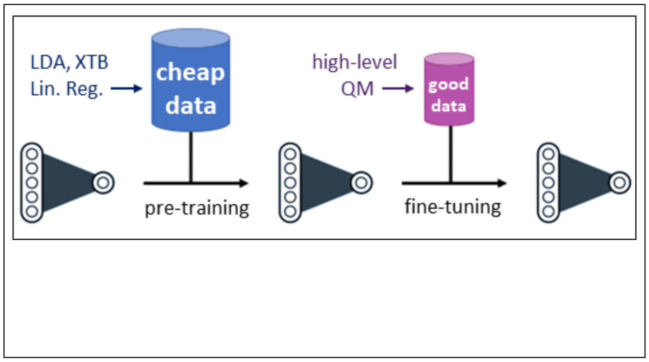
(65) Low, K.; Coote, M. L.; Izgorodina, E. I. Explainable Solvation Free Energy Prediction Combining Graph Neural Networks with Chemical Intuition. *J. Chem. Inf. Model.* **2022**, *62*, 5457–5470.

(66) Zhang, D.; Xia, S.; Zhang, Y. Accurate Prediction of Aqueous Free Solvation Energies Using 3D Atomic Feature-Based Graph Neural Network with Transfer Learning. *J. Chem. Inf. Model.* **2022**, *62*, 1840–1848.

(67) Zhang, Z.-Y.; Peng, D.; Liu, L.; Shen, L.; Fang, W.-H. Machine Learning Prediction of Hydration Free Energy with Physically Inspired Descriptors. *J. Phys. Chem. Lett.* **2023**, *14*, 1877–1884.

(68) Bennett, L.; Melchers, B.; Proppe, B. *Curta: A General-purpose High-Performance Computer at ZEDAT, Freie Universität Berlin*; Freie Universität Berlin, Germany, 2020.

TOC Graphic


LDA, XTB
Lin. Reg.
cheap
data
pre-training
high-level
QM
good
data
fine-tuning