Environmental Science Advances

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Environ. Sci.: Adv.*, 2024, **3**, 1438

Accurate modeling of the potential energy surface of atmospheric molecular clusters boosted by neural networks†

Jakub Kubečka, *[a] Daniel Ayoubi, [a] Zeyuan Tang, [b] Yosef Knattrup, [a] Morten Engsvang, [a] Haide Wu [a] and Jonas Elm [a]

The computational cost of accurate quantum chemistry (QC) calculations of large molecular systems can often be unbearably high. Machine learning offers a lower computational cost compared to QC methods while maintaining their accuracy. In this study, we employ the polarizable atom interaction neural network (PaiNN) architecture to train and model the potential energy surface of molecular clusters relevant to atmospheric new particle formation, such as sulfuric acid–ammonia clusters. We compare the differences between PaiNN and previous kernel ridge regression modeling for the Clusteromics I–V data sets. We showcase three models capable of predicting electronic binding energies and interatomic forces with mean absolute errors of <0.3 kcal mol^{-1} and <0.2 kcal mol^{-1} Å^{-1}, respectively. Furthermore, we demonstrate that the error of the modeled properties remains below the chemical accuracy of 1 kcal mol^{-1} even for clusters vastly larger than those in the training database (up to $(H_2SO_4)_{15}(NH_3)_{15}$ clusters, containing 30 molecules). Consequently, we emphasize the potential applications of these models for faster and more thorough configurational sampling and for boosting molecular dynamics studies of large atmospheric molecular clusters.

Received 1st July 2024
Accepted 9th August 2024

DOI: 10.1039/d4va00255e

rsc.li/esadvances

Environmental significance

Atmospheric aerosol particles significantly impact human health, atmospheric chemistry, and climate. Yet, aerosol processes remain poorly understood, introducing major uncertainties in climate models. New particle formation is a process driven by formation of stable molecular clusters that grow into aerosols. While theoretical and experimental studies qualitatively agree for simple systems, massive errors are observed in complex systems. Unfortunately, accurate quantum chemical calculations for these systems are computationally demanding. However, machine learning can replicate those at a fraction of the computational cost. We trained neural networks on various systems and demonstrated their effectiveness for large molecular clusters. This is a significant step toward large-scale, *ab initio* modeling of atmospheric nucleation that will potentially reduce uncertainties in climate predictions.

1 Introduction

The formation and growth of molecular clusters in the atmosphere drive the gas-to-particle conversion process known as new particle formation (NPF). These aerosols contribute to a net cooling effect on the Earth.[1] Almost 50% of newly formed aerosols act as cloud condensation nuclei (CCN), enhancing cloud formation and the albedo effect.[2,3] Additionally, due to their tiny size, aerosols can transport various molecules, viruses, and bacteria deeply into the respiratory system, posing health risks.[4–6]

While the exact chemical species relevant to aerosol formation mechanisms are poorly understood, many studies present insight into the most relevant components, such as sulfuric acid, ammonia, dimethylamine, and various oxidation products of volatile organic compounds.[7–10] Other studies have employed computational quantum chemistry (QC) and modeling to investigate the first crucial steps of formation mechanisms; molecular cluster formation.[11–13] One of the main paths is through stable inorganic acid–base salts. Accurate QC methods are required to capture the chemistry of common acid–base clusters, which are stabilized by proton transfer; a bond-breaking/-formation reaction. However, these methods are computationally expensive and scale significantly with molecular size, which often limits computational studies to small clusters, typically with less than 10 molecules. To overcome the computational limitation while maintaining accuracy, alternative approaches for future practical research are needed.

[a]*Department of Chemistry, Aarhus University, Langelandsgade 140, 8000 Aarhus C, Denmark. E-mail: ja-kub-ecka@chem.au.dk; Tel: +420 724946622*

[b]*Center for Interstellar Catalysis, Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, 8000 Aarhus C, Denmark*

† Electronic supplementary information (ESI) available: SI-1 – package and data availability, SI-2 – rankings of hyperparameter optimization searches, SI-3 – force vector angle deviation analysis, SI-4 – ωB97X-D/6-31++G(d,p) and B97-3c correlation, SI-5 – performance of force modeling, SI-6 – data reduction test. See DOI: https://doi.org/10.1039/d4va00255e

Machine learning (ML) offers a versatile and powerful solution for accelerating time-consuming processes characterized by repeating patterns.[14] In chemistry, ML has found applications in modeling potential energy surfaces (PES), predicting molecular structures, identifying and classifying molecules, enhancing molecular dynamics (MD) simulations, predicting kinetics and other properties, and gaining deeper insights into molecular behavior.[15–20] To date, and to the best of our knowledge, there are few studies utilizing ML for atmospheric molecular clusters.[21–25] Our recent studies focused on using kernel ridge regression (KRR) models (implemented in QML[26,27]) for modeling the cluster binding energies.[28–33] Despite its simplicity, KRR achieved chemical accuracy (<1 kcal mol^{-1}) when modeling the energies of sulfuric acid–water clusters, using only a few hundred structures for training and the rest of the database for testing.[28] However, Knattrup *et al.*[34] demonstrated that the computational costs and accuracy of KRR for modeling of density functional theory (DFT) binding energies could be substituted by fast but (compared to DFT) less accurate DFT-3c[35,36] methods with similar results. Moreover, for accurate modeling, the computational costs of the KRR approach (kernel construction and Cholesky decomposition) almost reach those of QC methods. Additionally, enlarging the configurational space by incorporating more non-equilibrium structures and increasing system complexity by introducing more atom/molecule types would require very large training databases to maintain low errors, increasing costs even further.

This work investigates whether neural networks can address these challenges. We utilize one of the commonly applied NN architectures, the polarizable atom interaction neural network (PaiNN),[37] implemented in SchNetPack,[38,39] to model the energies and forces of typical atmospheric molecular clusters. We demonstrate the accuracy and speed of the trained models and discuss their potential applications in future studies, including those involving large and complex training databases.

2 Methodology

2.1 Databases

This work uses structures from three molecular cluster databases (*cf.* Fig. 1 and SI-1†). The first is the sulfuric acid (SA) and water (W) system studied in our previous work.[28] To sum up, we had collected ∼1.7k equilibrium structures from other studies[40–48] and used Born–Oppenheimer MD (BOMD) simulations with energies and forces computed at a low QC level (the PM7 semi-empirical method) starting from each equilibrium structure in order to expand the database with several non-equilibrium structures. Thus, overall, this database contains ∼18k structures, which consist of a subset of the $SA_{0-5}W_{0-15}$ clusters. The system properties are evaluated for each geometry at the ωB97X-D[49]/6-31++G(d,p) level of theory.

The second SA and ammonia (AM) cluster database has been taken from Engsvang and Elm.[29] They used the GFN1-xTB[50,51] level of theory to produce ∼4k equilibrium structures and ∼7.7k non-equilibrium structures. In this work, we used the equilibrium structures as starting points for BOMD simulations at 300 K while producing an additional ∼6.4k structures. The overall database thus consists of ∼18k $SA_nAM_{n,n\pm1,n\pm2}$ clusters,

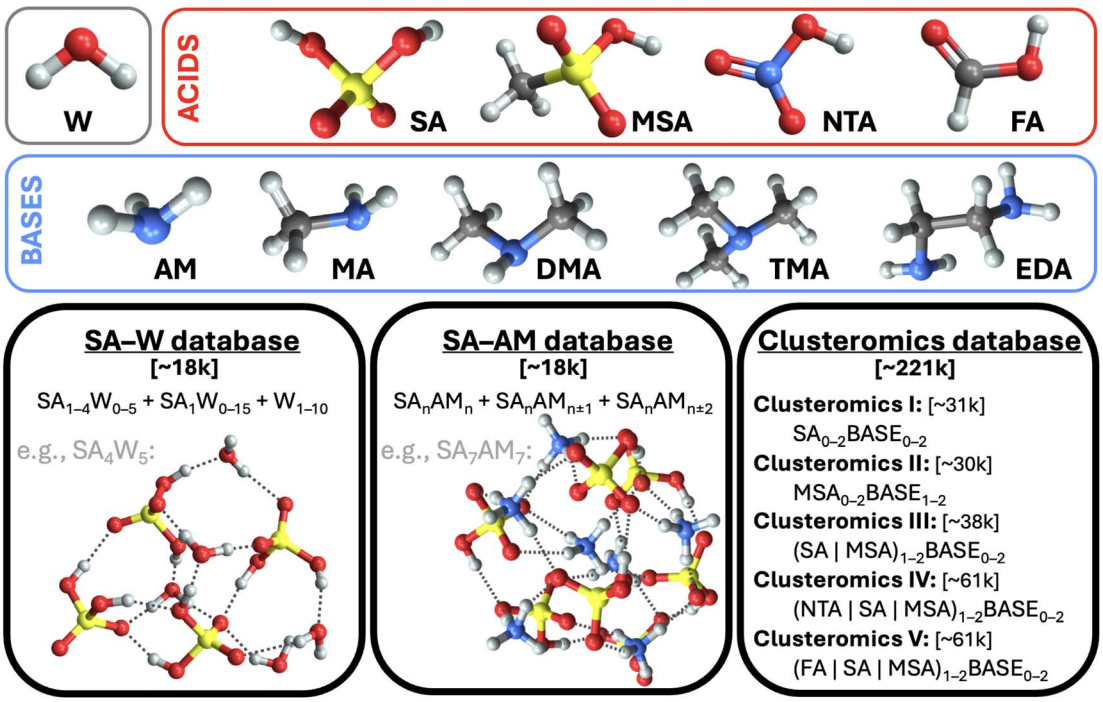


Fig. 1 The three databases of molecular cluster structures used in this work. Legend: water (W), ammonia (AM), methylamine (MA), dimethylamine (DMA), trimethylamine (TMA), ethylenediamine (EDA), and sulfuric (SA), methanesulfonic (MSA), nitric (NTA), and formic (FA) acids. Color code: sulfur (yellow), oxygen (red), hydrogen (white), nitrogen (blue), carbon (gray).

where $n \in$ 1–9. The GFN1-xTB level cannot provide accurate binding energies, and using ωB97X-D/6-31++G(d,p) would be too computationally demanding for all cluster sizes. Therefore, using the same methodology as Engsvang and Elm,[29] the energies and forces were recalculated at the B97-3c[35] level.

Although both databases are similar in size, the SA–AM database contains larger clusters, an extra atom type (nitrogen), and a more complex scheme of cluster binding patterns than just O–H interactions, as in the SA–W database. We additionally recalculated the SA–W database at the B97-3c[35] level. Therefore, we have three different datasets (SA–$W^{\omega B97X\text{-}D/6\text{-}31++G(d,p)}$, SA–$W^{B97\text{-}3c}$, and SA–$AM^{B97\text{-}3c}$) to compare for the suitability of ML. As such, we can examine the role of the theory level and database complexity in ML training.

Finally, the third database of structures originates from Knattrup *et al.*,[32] who compiled small (up to 4-molecule) clusters composed of various NPF precursors such as (SA, MSA = methanesulfonic, NTA = nitric, and FA = formic) acids and (AM, MA = methylamine, DMA = dimethylamine, TMA = trimethylamine, and EDA = ethylenediamine) bases. Knattrup *et al.* combined the Clusteromics I–V databases[12,52–55] consisting of equilibrium structures optimized at ωB97X-D/6-31++G(d,p). Furthermore, they employed MD simulations at GFN1-xTB[50,51] to greatly expand the database to ∼221k structures, denoted Clusterome. Ultimately, the system properties (single-point energies) were evaluated for each geometry at r^2SCAN-3c.[36] The entire Clusterome database offers an opportunity to examine the capability of NN for learning on structurally complex (clusters composed of nine different molecules) and large databases or appropriately reduced databases.[56]

We emphasize the importance of consistency during data preparation. For instance, a mismatch of quantum chemistry (QC) program versions or methods could cause great errors in the modeling. Therefore, to deal with the database and the QC evaluations systematically, we have used the JK framework, a set of computational tools for handling molecular clusters.[33] The following QC programs were used for QC calculations: xtb 6.4,[57] Gaussian 16 B.01,[58] and ORCA 5.0.3.[59,60]

2.2 Data properties

Each cluster structure (*XYZ* Cartesian coordinates) in the databases has an associated electronic energy and, in some cases, also forces derived from electronic energy gradients. For ML modeling, the structure must have a suitable molecular representation (explained in Section 2.3) since different *XYZ* coordinates can correspond to the same structure when translated, mirrored, or atom-wise permuted since *XYZ*s are translation/rotation/atom-permutation noninvariant. Furthermore, relative energies (*e.g.*, atomization or binding energies) are preferable to absolute electronic energies, as they exhibit a lower spread of the modeled values between different clusters, which simplifies the data fitting. Therefore, we use the electronic binding energies ΔE, *i.e.*, the energies released upon cluster (C) formation from its monomers (M) at energy-minimum configuration

$$\Delta E = E_{\mathrm{C}} - \sum_{i \in \mathrm{M}} E_i^{\min}. \quad (1)$$

The gradient of E_{C} is the same as for ΔE, *i.e.*, still corresponding to intramolecular forces. When evaluating the ML model quality, we use mean absolute errors (MAEs) and root mean squared errors (RMSEs) between the predicted and true properties. In the case of interatomic forces ($\vec{F}$), we compare all force components separately.

In our previous work,[28] we demonstrated that the Δ-ML approach[61–65] could enhance the accuracy of predictions by a factor of four compared to direct-ML. Here, direct learning refers to modeling the high-level binding energy (here denoted as ΔE^{high}). Δ-ML models the difference ($\Delta\Delta E$) between a slow, high-level theory (ΔE^{high}) and a fast, low-level theory (*e.g.*, GFN1-xTB,[50,51] here denoted as ΔE^{low}),

$$\Delta\Delta E = \Delta E^{\mathrm{high}} - \Delta E^{\mathrm{low}}. \quad (2)$$

If the low- and high-level methods correlate, the model does not need to learn as much chemistry, as the fast method captures most of it. While the combination of Δ-ML and NN is uncommon as the improvement is not marginal and Δ-ML is less practical for final application (*e.g.*, MD simulations), we only use it for consistency and eventual comparisons with our previous studies. Therefore, when Δ-ML is used, we used GFN1-xTB as the 'low' method.

2.3 Neural network model

In this work, we use the polarizable atom interaction NN (PaiNN[37]) model, the successor to the SchNet[66] model. Based on Liao *et al.*,[67] other NN models (*e.g.*, TorchMD-NET,[68] NequIP,[69] and Equiformer[67]) can achieve better accuracy when modeling energies and forces. However, these models require longer training times while the accuracy (MAEs) is only improved by a factor of two or less.

PaiNN employs message passing for 3D-embedded graphs, where the molecular representation is constructed on the fly for each graph node (atom) based on its surrounding environment (by default 5 Å). Atomic features are updated through a sequence of interaction layers, which, followed by an atom-centered neural network, allows modeling of the final property of interest (*e.g.*, energy). For a more detailed description, we recommend consulting the original reference.[37]

The PaiNN model is implemented in the SchNetPack[38,39] program. Communication with SchNetPack is also performed *via* the JK framework.[33]

2.3.1 Model hyperparameters. Hyperparameters investigated in this work include molecular representation, neural network, and training settings that can significantly impact the model accuracy and performance. For PaiNN, some crucial hyperparameters are the initial learning rate (LR), number of epochs (EPOCHS; training length), batch size (BS), number of interaction layers (INT), number of atom basis features (AB), and number of radial basis functions (RB) within a cutoff distance (CUTOFF).

2.3.2 Model training. The quality of training is determined by MAE between the modeled and true values. Typically, the majority (we use 90%) of the training database is used as the training subset, and through a sequence of batch iterations/

epochs, the MAE is minimized. The remaining portion (10%) of the database is used for validation to prevent overfitting of the training database. In such cases, the training is often stopped when the validation MAE consistently increases. However, there are two other cases when the training is typically stopped: (1) when the validation MAE has reached a plateau, *i.e.*, no improvement occurred for a certain number of epochs (early stopping, ES, threshold), or (2) the validation MAE has reached the desired accuracy.

When training a model with multiple properties, it is common to use a loss function combining the MAEs of all the model properties. In the case of energies and forces, our loss function is

$$\mathscr{L} = \xi \cdot \left\langle \left| \Delta E_{\mathrm{C}}^{\mathrm{true}} - \Delta E_{\mathrm{C}}^{\mathrm{model}} \right| \right\rangle + (1-\xi) \cdot \left\langle \left| \Delta F_{\mathrm{C}}^{\mathrm{true}} - \Delta F_{\mathrm{C}}^{\mathrm{model}} \right| \right\rangle, \quad (3)$$

where $\langle|\cdot|\rangle$ corresponds to MAE and ξ to trade-off between the two modeled properties. This work sets the trade-off to 1% when training on both energies and forces.

Training on the full database can become computationally demanding in the case of very large databases. Smith *et al.*[56] suggested selecting a small subset of the full database (*e.g.*, 2%) and training several models, each initiated with a different random seed. The largest deviations (or deviations greater than a certain cutoff) between the models' predictions on the remaining part (98%) of the database can be used to identify and select the problematic structures for expanding the training database. This process is iteratively repeated until the final trained models consistently predict properties of the full database within the desired accuracy. We test this database reduction in Section 3.5.

3 Results

3.1 Understanding the NN training

3.1.1 Training curves. We will first investigate the training behavior before moving towards the NN-model training on the full databases. Here, we used two NN models (NN-small and NN-big) defined by choosing suitable hyperparameters based on intuition. The NN-small model (BS = 100, LR = 10^{-4}, AB = 64, INT = 3, RB = 15, and CUTOFF = 5 Å) contained 154k trainable parameters and NN-big (BS = 100, LR = 10^{-4}, AB = 256, INT = 5, RB = 30, and CUTOFF = 5 Å) contained 3.8 M trainable parameters. The other hyperparameters were the same as for the final model (see section 3.1.2). The models have been separately trained on electronic binding energies (ΔE) for random samples of 1k and 16k structures from the SA–W database.

Fig. 2 shows the training MAE evolutions for the four training cases. Due to the model complexity, the NN-big models reached training MAEs of 0.05 kcal mol^{-1} [16k] and 0.3 kcal mol^{-1} [1k], an order of magnitude lower MAEs than their NN-small counterparts, which reached MAEs of 0.3 kcal mol^{-1} [16k] and 2.5 kcal mol^{-1} [1k]. After a few hundred to several thousand epochs, only fine-tuning of the accuracy occurs, except for the case of NN-big [1k], where the fitting accuracy is still significantly improved. It should be noted that in the case of NN-big [1k], the final improvement might already be overfitting the

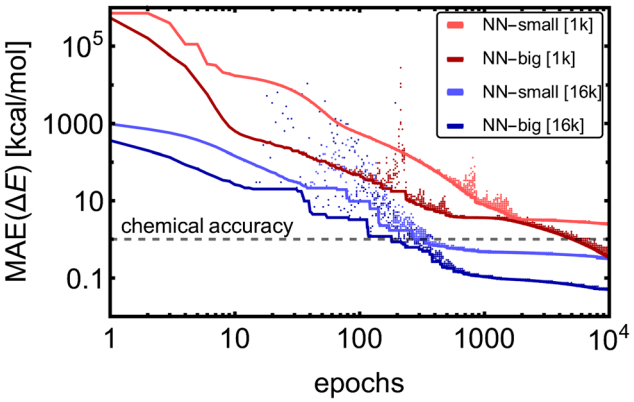


Fig. 2 Mean absolute errors (MAEs) of electronic binding energies (ΔE) for the training SA–W database with different training set sizes (1k/16k) and different numbers of NN fitting parameters (NN-small/NN-big). Dots show the training MAE from each epoch, and lines show the MAE of the best model found so far.

training database, and validation should be examined simultaneously (see Section 3.1.3). Yet, these results do not show a clear trend between the NN model complexity and the number of epochs required to reach the fine-tuning regime. The larger training datasets [16k] converge approximately 16 times faster to similar values compared to the small datasets [1k] because we use 16 times more training batches within each epoch. To sum up, the training MAEs of NN-small [1k] can be lowered by ~2 orders magnitude by greater variation in the training datasets [1k → 16k] and by enlarging the model [NN-small → NN-big].

Training a moderately complex neural network from scratch on a large dataset boosted by a GPU takes from several hours to a few days. However, this can vary widely depending on the specific circumstances: model complexity in the form of the number of fittable parameters, training dataset size and complexity (*e.g.*, 1k/16k data of SA–W/SA–AM will differ), hardware (number and speed of CPUs/GPUs), optimization technique (*e.g.*, Adam or stochastic gradient descent (SGD[70])), learning strategy (*e.g.*, learning rate or transfer learning = adaptive learning), and number of epochs. Training is the bottleneck of NN modeling, as predicting properties of thousands of structures takes a couple of minutes when employing a single CPU. Table 1 illustrates the training times required for one epoch of different datasets. Various commonly-used hyperparameters can alter the computational time by a factor of 0.5–2. The main conclusions from Table 1 are that the training on larger databases (16k *vs.* 1k) is proportionally slower (*i.e.*, 16 times), training on more complex databases (SA–AM *vs.* SA–W) is slightly slower (by ~25%), and training more properties ($(\Delta E, \vec{F})$ *vs.* ΔE) doubles the times in our case. While utilizing GPUs make the training faster by more than an order of magnitude, we typically only have access to a few GPUs but can utilize thousands of CPUs. Therefore, the wide range of training runs required for hyperparameter optimization is performed using only CPUs with fewer epochs.

3.1.2 Hyperparameter optimization. Non-surprisingly, the previous section illustrates that model complexity and training set size have an impact on the accuracy of the final model. Using

Table 1 Approximate computational times for training cases and hardware setups. Here, x CPU = x Intel Xeon Platinum cores, and 1 GPU + 4 CPUs = Nvidia V100-16GB + 4 Intel Xeon Gold cores

System	Training on	Train size	1 CPU	4 CPUs	1 GPU + 4 CPUs
SA–W	ΔE	1k	∼4 min per epoch	∼2 min per epoch	∼8 s per epoch
SA–W	ΔE	16k	∼1 h per epoch	∼0.5 h per epoch	∼50 s per epoch
SA–W	ΔE + forces	1k	∼8 min per epoch	∼4 min per epoch	∼16 s per epoch
SA–AM	ΔE	1k	∼6 min per epoch	∼3 min per epoch	∼10 s per epoch

the same training database, it is clear after a few (10–100 s) epochs which set of hyperparameters performs better in training (see Fig. 2). Such an assumption is not universally valid as some setups could converge slower but to more accurate values. Nevertheless, with this approach, we should be able to find the model that most rapidly converges to low MAEs. Therefore, in this section, we seek hyperparameters that reach the lowest validation MAEs after 200 epochs.

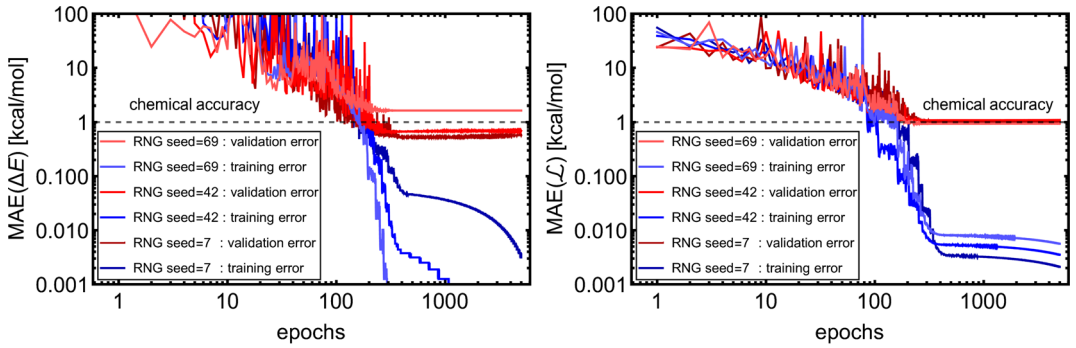
Taking into account the most important hyperparameters (AB, INT, RB, LR, and BS), we used an in-house numerical optimization script to find optimal hyperparameters for the NN-model trained on ΔE of 1k SA–W data. However, as the outcome of NN training is not predictable and depends on the initial seeding of the network, the optimizer was continuously cycling. It could not converge even when we averaged the results over three independent trainings. Consequently, we performed a simpler brute-force grid search, where the grid consisted of AB = (32, 128, 512), INT = (3, 4, 5), RB = (10, 20, 30), LR = (10^{-3}, 10^{-4}, 10^{-5}), and BS = (2, 5, 10, 50, 100). For simplicity, we kept the remaining hyperparameters at the default values. Many of the training runs reached validation MAEs down to 0.6–2 kcal mol^{-1} for the 1k SA–W data. Subsequently, we performed the same grid search for training on $(\Delta E, \vec{F})$ of 1k SA–W data, on ΔE of 16k SA–W data, using Δ-learning on ΔE of 1k SA–W data, and also on ΔE of 1k SA–AM data. While the larger (16k) and more complex (SA–AM) dataset preferred more complex NN models, in the overall ranking, the model presented in Table 2 seems the most suitable for all the studied systems, and we will use this setup for the training of the following NNs. Although additional hyperparameter optimizations will likely not reach significantly more accurate models, they should be performed on new systems, as some of the 'optimal' hyperparameters are on the edge of the grid search values. The rankings are presented in the SI-2.†

Table 2 The final hyperparameters used for all the following models in this work. The optimized hyperparameters are highlighted in bold

Category	Hyperparameter	Value
Representation	AB = atom basis	**128**
Representation	INT = interaction layers	**5**
Representation	RB = radial basis	**20**
Representation	CUTOFF = cut off	5 Å
Model	Properties	ΔE or $(\Delta E, \vec{F})$
Model	Loss function	Eqn (3)
Model	ΔE *vs.* $\vec{F}$ trade off ξ	100% or 1%
Model	Data used for validation	10%
Optimization	Optimizer	AdamW
Optimization	LR = learning rate	**0.0001**
Optimization	Learning rate scheduler	OFF
Optimization	Epochs	Varies between ∼10^2–10^3
Optimization	Early stopping	OFF
Optimization	BS = batch size	**2**
Initiation	RNG seed	42 or (7, 42, 69)

The final model comprises nearly 1 M trainable parameters, where this number is primarily defined by AB, INT, and RB. While we did not vary the CUTOFF, it is another crucial parameter to consider, especially for the dynamics of molecular cluster formation. Although the interaction of distant atoms within a cluster will be propagated *via* message-passing through other atoms,[71] studying cluster evaporation or collisions using molecular dynamic simulations might be inaccurate due to the lack of long-range interactions. However, this is beyond the scope of the current manuscript. At the same time, using a large CUTOFF parameter makes the NN more complex and computationally more expensive, *i.e.*, more difficult to train. Another option for properly incorporating long-range interactions would be using a different ML model (*e.g.*, PhysNet[72] or SpookyNet[73]) that combines short-ranged interaction modeling with small CUTOFF and long-range/dispersion corrections calculated, *e.g.*, from atomic positions and partial charges.[74,75] Further, for minimizing the loss function, we utilize the Adam optimizer with weight decay (AdamW) and L_2 regularization.[76] The learning rate (LR) affects the step size of the gradient descent algorithm during training. An excessively high rate may cause overshooting or overfitting, whereas a very low rate may result in slow convergence. Although the LR of 10^{-3} has reached the best ranking in our grid search, we reduced the LR to 10^{-4} to reduce instabilities during training. Ideally, an LR scheduler should be used to progressively lower the learning rate. However, we disabled it for consistency, with the schedule multiplier set to 1. We use a very small batch size of 2 compared to the typical 100 or 200. This size represents the number of training samples used in one optimization iteration. Smaller batch sizes can lead to faster convergence as more iterations are performed within one epoch. Conversely, larger batch sizes (typically limited by computer memory) can provide more accurate iteration during optimization and, thus, can be used for fine-tuning.

3.1.3 Training and validation. Using the optimal hyperparameters, we separately trained the NN model on ΔE and $(\Delta E, \vec{F})$ of 1k of SA–W data. We initiated three separate training runs for each model to eliminate any randomness, using random-number-generator (RNG) seeds of 7, 42, and 69. Fig. 3 shows training and validation errors for all the runs. All models reach validation MAEs of ∼1 kcal mol^{-1}. Smaller variations in



(a) Training only on energies. (b) Training on energies and forces.

Fig. 3 Training and validation error evolution with epochs for 1k of the SA–W system with three times differently initiated training. RNG = random number generator.

validation MAEs among differently initiated models are observed when training on both energies and forces, as the loss function variation is significantly reduced by averaging over $3N + 1$ times more values (*i.e.*, $3N$ $\vec{F}$ components and 1 ΔE). In all cases, only ~300 epochs are required to converge the model. After 200–300 epochs, further training appears redundant as it only overfits the training database and, in some cases, even increases the validation MAE. Due to low variations with different RNG seeds, we further only use seed 42. Note that variations among differently initiated models would become more apparent when the training database is also altered (see Section 3.4 and 3.5).

3.1.4 Learning curve. To investigate the learning curve, we used the SA–W database of 18k clusters calculated at ωB97X-D/6-31++G(d,p). The same data were used in our previous work[28] employing kernel ridge regression (KRR), implemented in the QML program.[26,27] For consistency, we also use 520 largest SA_4W_5 clusters for testing and the rest for training and validation. The data sampling and simulations are repeated three times for statistics. Each model uses 1000 epochs for training.

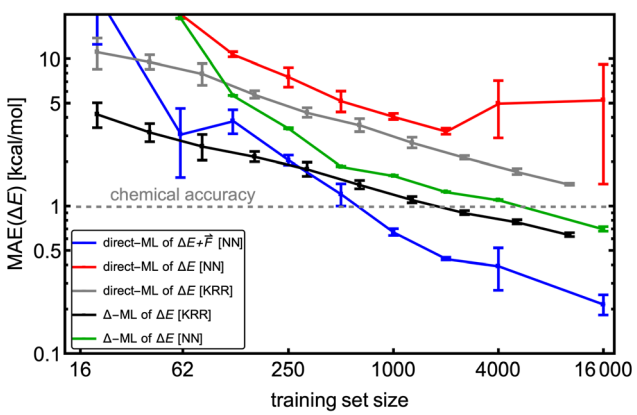


Fig. 4 The learning curves for training on the SA–W clusters while testing on the largest SA_4W_5 clusters excluded from the training. Different lines correspond to direct-/Δ-learning of KRR/NN model trained on $(\Delta E/\Delta E, \vec{F})$. The kernel-ridge-regression (KRR) results are taken from Kubečka *et al.*[28] The error bars represent the standard deviation. Note the logarithmic axes.

We model the electronic binding energies (ΔE). However, separate models are also trained on ΔE, $(\Delta E, \vec{F})$, and Δ-learning with GFN1-xTB used as the low method (see eq. (2)).

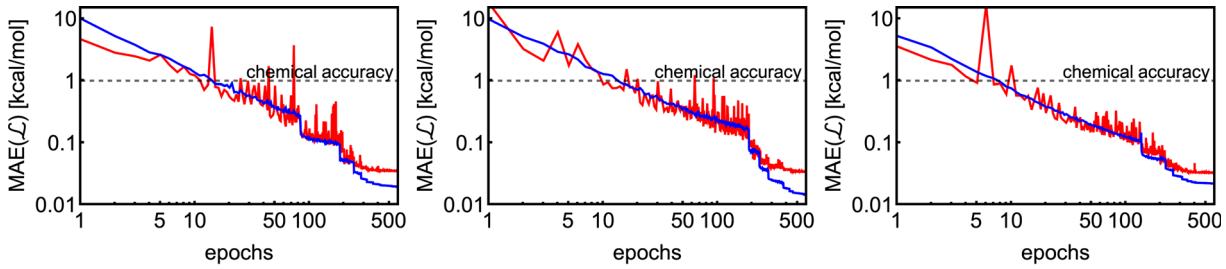
We use the same NN model for all systems and database sizes. Fig. 4 presents all the KRR[28] and our new NN learning curves. KRR performs better with small (<~200) training data sizes compared to NN. For both KRR and NN, Δ-learning improves the accuracy by a factor of 2–3. When training solely on energies, this NN model achieves ~1.2 × MAE of the corresponding KRR model with the same training set size, reaching the MAE of 0.7 kcal mol^{-1} at 16k training data. Therefore, this model reaches below the chemical accuracy of 1 kcal mol^{-1}. The NN trained on both $(\Delta E, \vec{F})$ outperforms others with MAE of ΔE lower than 0.3 kcal mol^{-1}. Note that for NN, the mean absolute errors (MAEs) are slightly greater than the validation MAEs presented in the previous section, as the testing is performed on the largest SA_4W_5 clusters while validation is performed on the 10% of the data cut from the training databases.

Interestingly, direct learning of the NN model appears to be more susceptible to initialization and data choice. The large variations in the direct learning of the NN model on energies (red line) for large training sizes indicate that this choice of hyperparameters makes the model very sensitive to the initial conditions, and a search for more suitable hyperparameters at these sizes would be beneficial. Chen *et al.*[18] also demonstrated variation in their NN modeling (with NN model termed as VSpecNN) and suggested averaging over three independent NN models reduces MAEs of energies and forces by ~30%.

Most importantly, NN computational times for larger training sizes significantly outperform the KRR times. Also, predictions of the NN model are orders of magnitude faster than for KRR, which scales quadratically with the training set size or even cubically for large sets.

3.2 The model performance

We trained three NN models on the full SA–$W^{\omega B97X\text{-}D/6\text{-}31++G(d,p)}$, SA–$W^{B97\text{-}3c}$, and SA–$AM^{B97\text{-}3c}$ databases, while training on both energies and forces. Fig. 5 shows the evolutions of the loss function, which are mainly dependent on the model and system complexity and the learning rate. Both SA–W cases behave

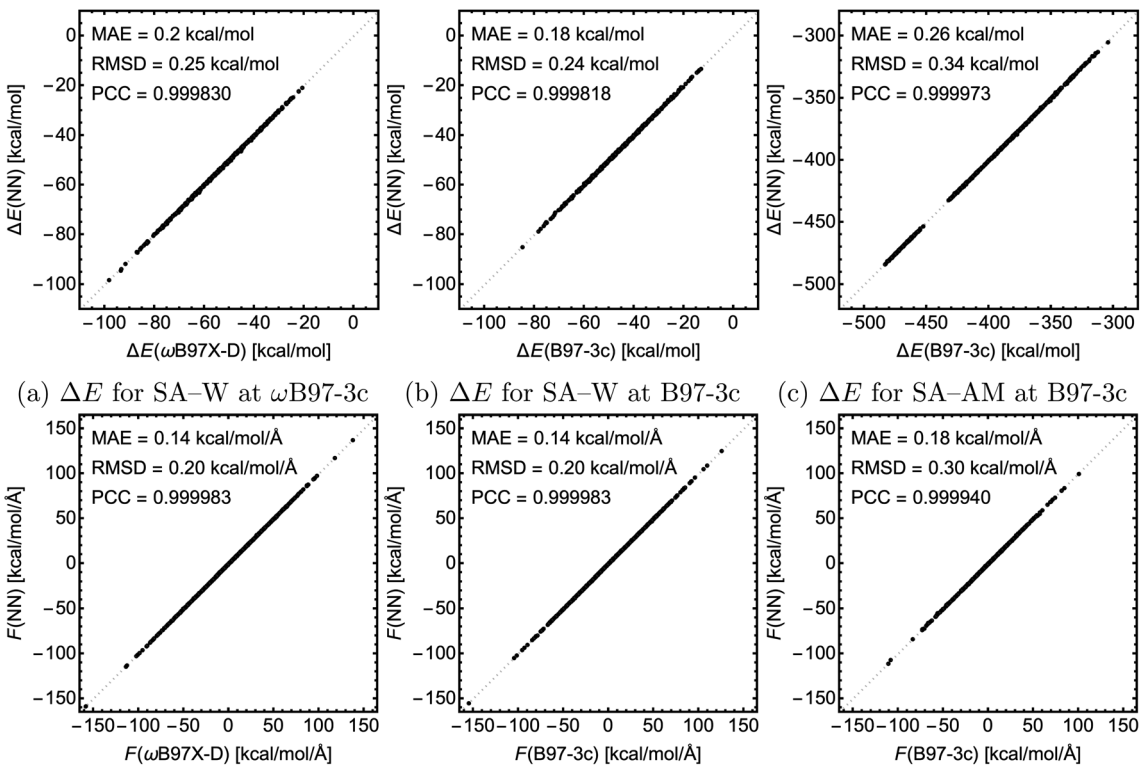


(a) SA–W at ωB97X-D (b) SA–W at B97-3c (c) SA–AM at B97-3c

Fig. 5 Training (blue) and validation (red) evolution of loss function for the full databases (~18k) of the SA–W and SA–AM systems.

similarly as expected due to a high correlation between the B97-3c and ωB97X-D/6-31++G(d,p) binding energies. The SA–AM system also performs similarly to the SA–W cases as the SA–AM database complexity (cluster sizes and number of atom types) appears proportionally compensated by the NN model extension with one additional atomic feature set. All training runs clearly reached a plateau after 600 epochs. Each training took nearly three days using 1 GPU and 4 CPUs. All training and validations achieved mean absolute errors below 0.1 kcal mol^{-1}. The best model (*i.e.*, the model with the lowest validation loss) from each of the three training runs was subsequently used for testing.

For testing, we used the largest clusters in the database which were excluded from the training, *i.e.*, SA_4W_5 for the SA–W systems and (SA_8AM_{10}, SA_9AM_{10}, and SA_9AM_{11}) for the SA–AM system. Fig. 6 shows the correlation of the NN-modeled and the QC-calculated energies and forces across all systems. For chemical predictions, these models are quite accurate with RMSD and MAE < 1 kcal mol^{-1}, and with a very high correlation with the target method (PCC ≈ 1). The same applies to the force component predictions with RMSD and MAE < 1 kcal mol^{-1} Å^{-1} and force directions from angle deviation analysis with mean error < 1° (see SI-3†). In the case of SA–W, there is almost no apparent difference when modeling the two QC methods (ωB97X-D and B97-3c) due to their high correlation (see SI-4†). The SA–AM modeling shows only slightly larger MAE and RMSD than the SA–W modeling, likely for the same reasons as mentioned before, *i.e.*, the SA–AM database complexity appears



(a) ΔE for SA–W at ωB97-3c (b) ΔE for SA–W at B97-3c (c) ΔE for SA–AM at B97-3c

(d) $\vec{F}$ for SA–W at ωB97-3c (e) $\vec{F}$ for SA–W at B97-3c (f) $\vec{F}$ for SA–AM at B97-3c

Fig. 6 The correlation between the NN-modeled and the QC-calculated electronic binding energies and force components for the three studied cases. MAE = mean absolute error, RMSD = root mean squared difference, and PCC = Pearson correlation coefficient.

proportionally compensated by the NN model expansion. Similar to the previous section, note that all MAEs are slightly greater than the validation MAEs (see Fig. 5) as the testing is performed on the largest clusters while validation is performed on 10% of the data cut from the training database.

3.3 Application to large clusters

Here, we expand the above examination of the SA–AM model to even larger clusters and show the trend of MAEs with increasing cluster size. We took 25–50 random non-equilibrium clusters from the work of Wu *et al.*[77] for each SA_nAM_n cluster size, where $n \in 2$–15. Note that we only used clusters smaller than $SA_{10}AM_{10}$ for the training. In Fig. 7, we observe that the MAEs of the electronic binding energies and force components are almost linearly increasing with cluster size. This proportionality to cluster size arises from the sum of atomic contributions, with each atom likely contributing with a similar error. Additionally, the molecules are differently polarized within the large clusters, which might not be well-captured by the NN model, resulting in increasing MAE of the force components as well slightly increased growth of MAEs of the energies for the $SA_{10–15}AM_{10–15}$ clusters. Nevertheless, the MAEs consistently remain below the chemical accuracy threshold of 1 kcal mol^{-1} or 1 kcal mol^{-1} $Å^{-1}$.

We have a fast method that quite accurately learns the reference QC method. Fig. 8 shows the typical computational times required for optimization and vibrational-frequency calculations for the SA–AM clusters during configurational sampling.[77] Note that the times are multiplied by a factor of 8 as 8 CPUs were used even though the scaling of QC methods with the number of CPUs is not exactly linear. The frequency calculations take up approximately 15–50% of the computational time for the large clusters and up to 80% for the small. The scaling behavior of the B97-3c method is relatively moderate, exhibiting an almost linear trend in contrast to the poor scaling of other QC methods (*e.g.*, ωB97X-D or even the coupled-cluster methods). Regardless, many such calculations must be performed during thorough configurational sampling, which is computationally demanding. Utilizing the above-trained NN model for SA–AM, we also show the times required for geometry optimization with the same optimization criteria as for the QC optimization (see Fig. 8). The final geometry will be close to the true minimum, but post-optimization at the target QC level has to be applied to reach the same geometry. For example, we performed configurational sampling and took one SA_7AM_7 structure optimized at GFN1-xTB. Subsequent optimization of this molecule at B97-3c (within ORCA) lasted 10 CPU-hours and required ~80 iteration steps. The RMSD of the initial and final structure is 0.28 Å. Performing optimization (within ASE) with the trained NN takes 1 min per 100 iterations and approx. 100 iterations are required to reach the same optimization criteria as in the default settings of the ORCA program. The RMSD between both final structures is 0.04 Å, and with more iterations used in the NN optimization (300), the RMSD improvement is already negligible (lowered to 0.03 Å). When taking the NN-optimized structure, less than an extra 2 CPU-hours were required for B97-3c optimization to reach the minimum structure, which differs from the fully QC-optimized structure by ~0.01 Å in RMSD (considered as the same geometry). Although there is already a 5-fold speed-up in a single optimization, there will be a massive overall speed-up by omitting numerous energetically high-lying configurations after the NN pre-optimization. This clearly underlines that NN techniques will play an important role in future studies of large molecular clusters.

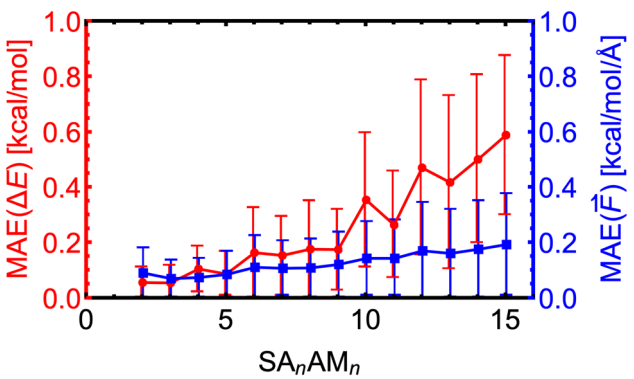


Fig. 7 Absolute error distributions of the modeled energies and force components for the SA_nAM_n clusters trained at the B97-3c level. The error bars show the standard deviation.

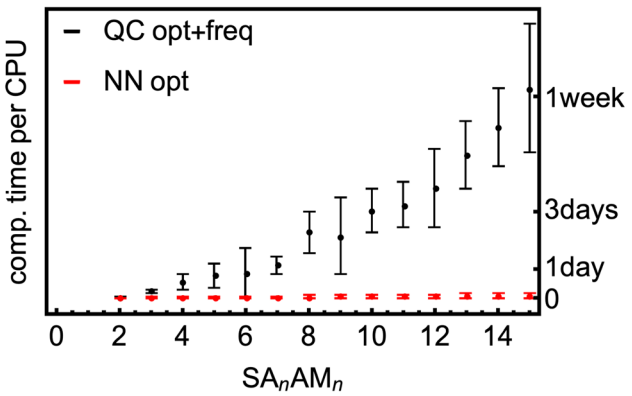


Fig. 8 SA–AM cluster size dependence of the computational times for the optimization and vibrational frequency calculations at the B97-3c level of theory compared to the short times required for NN (pre-) optimization.

3.4 Molecular-cluster dynamics *via* neural network

With an NN that can model forces, we can mimic Born–Oppenheimer molecular dynamics (BOMD) simulations of clusters at a significantly lower computational cost. We used the NN model from the previous section trained on SA–AM$^{B97\text{-}3c}$ to demonstrate this on a 100 ps long MD simulation. The simulations are performed using the Atomic Simulation Environment (ASE[78,79]), with a timestep of 0.1 fs, the Nosé–Hoover thermostat with friction frequency of 0.01 fs^{-1} and target temperatures (T) of 300 and 450 K. The entire 100 ps simulation only took ~9 hours on 1 CPU. With reduced data dumping and a well-optimized simulation script, 1 ns simulation could be achieved within 1–3 CPU-days. Although the choice of thermostat or other parameters might not be ideal for real applications,

this work aims to demonstrate the NN model's ability to quickly simulate large molecular clusters at the accuracy of a QC level of theory. Clearly, for MD simulations longer than a few picoseconds, the whole process of data generation, QC single-point and gradient calculation, NN training, and NN-boosted MD will become computationally faster than running BOMD at the QC level itself.

Fig. 9a illustrates the time evolution of the electronic binding energy. 100 uniformly sampled (every 1 ps step) structures were recalculated at the B97-3c level to further validate the model. Fig. 9b shows a high correlation between NN-modeled and QC-re-calculated energies. A similar satisfactory correlation is observed for the force components (see SI-5†). At the low temperature (300 K), the MAE of 0.24 kcal mol^{-1} is similar to errors observed during the comparison for structures from Wu *et al.*[77] (see Fig. 7). With higher temperature (450 K), the model performance is decreased and the MAE of 0.46 kcal mol^{-1} is almost twice as large as the MAE at the lower temperature. This can be attributed to the difference in data generation, and higher accuracy can be gained by expanding the database with appropriate structures. In other words, the training database was constructed by extracting structures from a short MD simulation at 300 K at GFN1-xTB (see Section 2.1), but MD simulations on the B97-3c potential energy surface and even at the higher temperature of 450 K will visit untrained parts of the configurational space. When we experimented with simulations at temperatures of 500 K or more, we experienced frequent simulation failures due to molecule fragmentation caused by inaccurate NN modeling. Nevertheless, below 500 K, the molecular cluster integrity remains unchanged during these simulations.

Even though the NN model maintains energy and force accuracy during the MD simulation, we also recommend examining whether other dynamics properties, such as radial distribution function and diffusivity, have been preserved (*e.g.*, see the work of Fu *et al.*[80]). MD simulations can enhance our understanding of many cluster properties. For instance, we

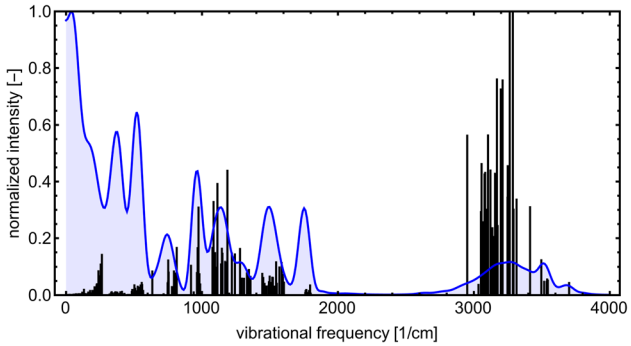
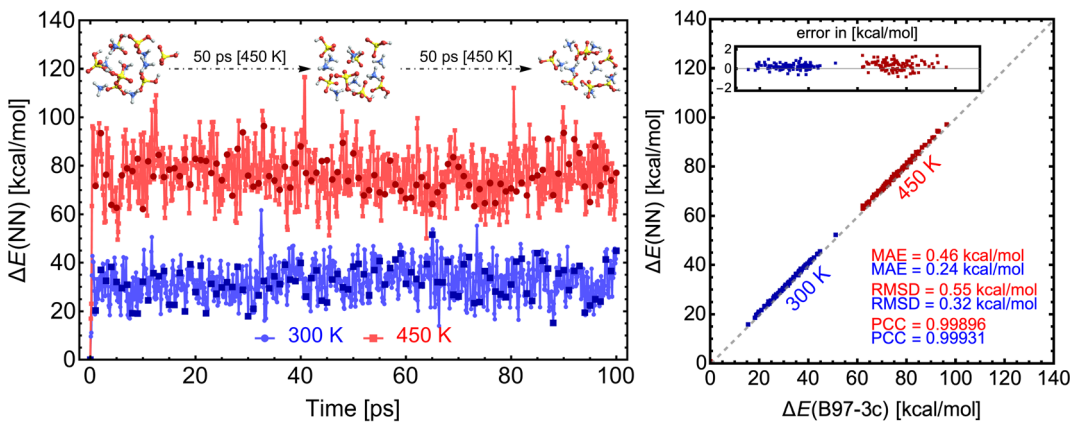


Fig. 10 The power spectrum for SA_7AM_7 at 300 K obtained from ML-boosted MD simulation (blue) and spectrum of harmonic-vibrational frequencies for the lowest found energy-minimum configuration at B97-3c level (black lines). All peaks are normalized according to the highest intensity.

used the TRAVIS program[81,82] to analyze the vibrations stored in the MD trajectory at 300 K. Fig. 10 demonstrates the analyzed power spectrum and compares it to the equilibrium harmonic vibrational frequencies of the lowest free energy SA_7AM_7 conformer. The QC harmonic interpretation of the vibrational frequencies is insufficient for these weakly bound clusters, crowded with many anharmonic and low-frequency vibrations.[83] Hence, MD simulation becomes important for interpreting some cluster behaviors. PaiNN was also designed to model directional properties such as dipole moments and polarizabilities. Modeling these properties along the generated trajectory allows calculating IR or Raman molecular spectra from molecular dynamics simulations. However, we omit them as this is beyond the scope of the current work.

3.5 Large database reduction

To construct a large database, we utilize the Clusteromics I–V databases (see Fig. 1). Electronic binding energies (ΔE) of these data were calculated at the r^{2}SCAN-3c[36] and GFN1-xTB[50,51] levels, making them suitable for Δ-ML. Knattrup *et al.*[32] used 5-fold



(a) Time evolution (b) Method correlation

Fig. 9 Molecular dynamics simulations of a SA_7AM_7 cluster with energies and forces predicted *via* the NN model. The right graph presents the correlation of NN-modeled and QC-re-calculated energies of 100 uniformly sampled structures from the MD trajectory. MAE = mean absolute error, RMSD = root mean squared difference, and PCC = Pearson correlation coefficient.

cross-validation and showed that the KRR model could reach MAEs lower than 1 kcal mol^{-1} for each Clusteromics separately when trained on very few data (~50–100). Note that the KRR-model error is low (compared to KRR modeling in Fig. 4) as the test/validation is also performed on small clusters, *i.e.*, not only the largest clusters in the database. Knattrup *et al.*[32] reported difficulties related to computational times for their large (>1k) training databases. Here, we performed similar training of the NN model (training for 1000 epochs) for each training set size and evaluated the model on the entire corresponding Clusteromics dataset. In Fig. 11, we demonstrate that the NN model again does not outperform the KRR model but consistently reaches the chemical accuracy of 1 kcal mol^{-1} for each Clusteromics set, with a training set requiring at least 2k of random data. We used the same 'optimal' hyperparameters of the NN model as in the previous sections, *i.e.*, optimized for slightly different systems, which is likely why the NN model reaches MAEs of 1 kcal mol^{-1} for 2k training dataset while the KRR model reaches 0.2 kcal mol^{-1}. The KRR has a large advantage when tested on data similar to the training data. Testing on structures that are different from the training database would potentially make the KRR and NN performance more comparable for large training sets. Nevertheless, training and validation times for NN modeling again outperform KRR and are no longer the main bottleneck of ML model applications. For instance, predicting 1000 energies with the trained NN model takes a few minutes using 1 CPU, outperforming KRR by orders of magnitudes, which requires days and many CPUs for training databases with more than 1k structures.

The Clusteromics I–V database combined into one ~221k large Clusterome database[32] offers a playground for data filtering/reduction. Fig. 11 demonstrates our ability to train the NN model for significantly larger training sizes than the KRR model. Despite only performing one training for each training set size, we see a smooth decrease in MAE down to ~0.3 kcal mol^{-1} for the 64k training dataset. Due to computational times, we have added the 128k and full Clusteromics MAE only based on the 100-epoch performance relative to the 64k database. Surprisingly, the MAEs are not much worse than in the case of separate Clusteromics modelings, likely due to the fact that the accuracy of electronic binding energies is mainly driven by the description of hydrogen bonds across all the Clusteromics datasets.

Finally, we applied the database reduction/active learning (AL) procedure presented by Smith *et al.*[56] We took 0.25k data, trained the NN model, and used the predictions on the full Clusteromics database to identify the next potential candidates for training. Here, we use the terms supervised AL, where we obtain the errors by comparing the predicted ΔE value to the true r^{2}SCAN-3c value, and unsupervised AL, where we obtain the errors as the standard deviation of the predicted ΔE values between three NN models (also known as 'query by committee'), initiated with different RNG seeds. The worst performing 0.25k data are added to the training set for the next iteration, where the NN model is again fully trained from scratch. Fig. 11 shows that the two methods (red and brown lines) are sensitive to the choice of the first 0.25k data but perform quite similarly after a few iterations. Unfortunately, they do not seem to perform better than the random sampling (green line). We even tested a random sampling of 0.25k data from the worst-performing systems (*e.g.*, with error threshold >1 kcal mol^{-1}) and continued training the NN model from the previous step (*i.e.*, no training from scratch) with no significant differences. We believe the database reduction will not significantly reduce MAEs for systems with similar chemical features. The driving mechanism of cluster formation is hydrogen bonding. The more training data, the more accurate the NN model. However, perhaps no particular outliers introduce hard-to-model hydrogen bonds or

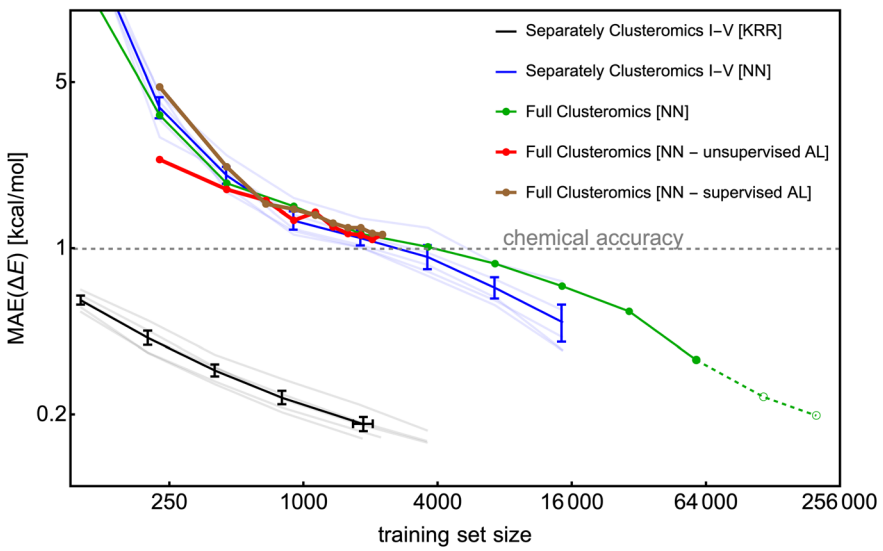


Fig. 11 The learning curves for the KRR and NN models trained on separate/full Clusteromics I–V datasets. The error bars represent the standard deviation over the Clusteromics I–V modelings. No error bars are assigned for the full Clusteromics modeling. The last two points of NN modeling on full Clusteromics are only extrapolated estimates based on 100-epoch training.

offer a significantly greater improvement when introduced to the training set. In SI-6,† we used KRR and tested the active learning on a small database (0.25k). There, active learning reduces the MAE by a factor of <1.5 compared to random sampling. However, active learning even reduces the maximum error by a factor of ∼2. Another option to test in the future would be selecting the worst MAEs per atom.

To conclude, although active learning appears to offer only a small MAE improvement for molecular clusters, it can help to eliminate outliers.

4 Conclusion

We used several databases of quantum chemistry (QC) data for typical atmospheric molecular clusters and showed that machine learning could easily substitute the computationally demanding QC calculations. Specifically, we used the polarizable atom interaction neural network (PaiNN) to model the cluster's binding energies, both with or without interatomic forces. We show that hyperparameter variation (*e.g.*, reducing batch size) leads to faster converging NN training without compromising accuracy. We demonstrate that NNs do not outperform the accuracy of the KRR modeling by Kubečka *et al.*,[28] but the computational times for NNs are significantly lower. Similar to the case of KRR modeling, we find that Δ-learning improves the accuracy ∼4-fold. We use the energies and interatomic forces to train NN models for sulfuric acid–water clusters and sulfuric acid–ammonia clusters with quite reliable performance compared to the trained QC methods even when tested on larger structures excluded from the training: mean absolute errors of <0.26 kcal mol^{-1} for energies and <0.18 kcal mol^{-1} Å^{-1} for force components, and root mean squared displacements of <0.34 kcal mol^{-1} and ∼0.30 kcal mol^{-1} Å^{-1}, respectively.

Furthermore, we show that these NN models will be very useful for application in configurational samplings of larger molecular clusters, as the atomic error contribution remains constant with increasing cluster size. While single-point energy evaluation at a high-level QC level for large clusters, with ten or more molecules, takes hours and often more than a day, the same evaluation with NN is nearly instant (∼seconds). We demonstrated this by comparing B97-3c and the NN model and showed the model's ability to optimize geometries and reproduce the B97-3c close-to-equilibrium structures.

Finally, we tested database reduction methods employing supervised and unsupervised active learning. The data reduction slightly improves the NN model performance for the same data sizes compared to random selection. We speculate that this could be caused by the fact that the driving mechanism of cluster binding is hydrogen bonding, which gets better described with more data, but there is no particular data that would introduce hard-to-model hydrogen bonds. Nevertheless, active learning appears to be a suitable tool for eliminating outliers. Furthermore, we find that the NN model trained only on energy, using Δ-learning, is able to utilize significantly more data than our previous KRR model.[32] While we again confirm that KRR performs better for smaller sizes, NNs also can reach MAEs lower than chemical accuracy of 1 kcal mol^{-1}, while outperforming KRR with respect to the computational times required for both training and predictions. We envision utilizing NN models for a better understanding of cluster dynamics. This includes cluster rigidity, reorganization after formation, reorganization before fragmentation, or even reactions within the cluster or on its surface. Overall, we believe that NN modeling will play a pivotal role in future studies of atmospheric molecular clusters.

Data availability

All computational programs are cited within the article. Additionally, software Mathematica was used for plotting graphs and molecule visualization. All our scripts are available on GitHub (referenced in ESI†) as well as the processed data used for ML training and result data from analysis.

Author contributions

Conceptualization: JK, ZT, JE; data curation: JK, YK; formal analysis: JK, DA, YK; funding acquisition: JK, JE, ZT; investigation: JK, DA; methodology: JK, DA, YK; project administration: JK; software: JK, ZT, DA, ME, HW; resources: JE; supervision: JE; validation: JK, DA; visualization: JK; writing – original draft: JK, JE; writing – review & editing: YK, ZT, DA, ME, HW.

Conflicts of interest

All authors declare that they have no conflicts of interest.

Acknowledgements

Funded by the European Union (MSCA, HYDRO-CLUSTER, project 101105506, and ERC, ExploreFNP, project 101040353), by the Danish National Research Foundation through the Center of Excellence (InterCat, project DNRF150) and through the Center of Excellence for Chemistry of Clouds (DNRF172), by the Independent Research Fund Denmark (9064-00001B), and by the VILLUM FONDEN (Investigator grant, project 16562). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The numerical results presented in this work were obtained at the Centre for Scientific Computing, Aarhus **https://phys.au.dk/forskning/faciliteter/cscaa/**.

References

1 G. Myhre, C. Lund Myhre, B. H. Samset and T. Storelvmo, Aerosols and their Relation to Global Climate and Climate Sensitivity, *Nat. Sci. Educ.*, 2013, **4**, 7.
2 J. Merikanto, D. V. Spracklen, G. W. Mann, S. J. Pickering and K. S. Carslaw, Impact of nucleation on global CCN, *Atmos. Chem. Phys.*, 2009, **9**, 8601–8616.

3 J. Tröstl, W. K. Chuang, H. Gordon, M. Heinritzi, C. Yan, U. Molteni, L. Ahlm, C. Frege, F. Bianchi, R. Wagner, *et al.*, The Role of Low-volatility Organic Compounds in Initial Particle Growth in the Atmosphere, *Nature*, 2016, **533**, 527–531.
4 C. I. Falcon-Rodriguez, A. Osornio-Vargas, I. Sada-Ovalle and P. Segura-Medina, Aeroparticles, composition, and lung diseases, *Front. Immunol.*, 2016, **7**, 1–9.
5 M. Mei, H. Song, L. Chen, B. Hu, R. Bai, D. Xu, Y. Liu, Y. Zhao and C. Chen, Early-life exposure to three size-fractionated ultrafine and fine atmospheric particulates in Beijing exacerbates asthma development in mature mice, *Part. Fibre Toxicol.*, 2018, **15**, 13.
6 World Health Organization (WHO), Ambient (outdoor) air pollution, 2022, **https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health**, (accessed February 7, 2023).
7 K. Lehtipalo, C. Yan, L. Dada, F. Bianchi, M. Xiao, R. Wagner, D. Stolzenburg, L. R. Ahonen, A. Amorim, A. Baccarini, *et al.*, Multicomponent new particle formation from sulfuric acid, ammonia, and biogenic vapors, *Sci. Adv.*, 2018, **4**, eaau5363.
8 S. Yang, Z. Liu, P. S. Clusius, Y. Liu, J. Zou, Y. Yang, S. Zhao, G. Zhang, Z. Xu, Z. Ma, *et al.*, Chemistry of new particle formation and growth events during wintertime in suburban area of Beijing: Insights from highly polluted atmosphere, *Atmos. Res.*, 2021, **255**, 105553.
9 F. R. Nursanto, R. Meinen, R. Holzinger, M. C. Krol, X. Liu, U. Dusek, B. Henzing and J. L. Fry, *What Chemical Species Are Responsible for New Particle Formation and Growth in the Netherlands? A Hybrid Positive Matrix Factorization (PMF) Analysis Using Aerosol Composition (ACSM) and Size (SMPS)*, EGUsphere, 2023, vol. 2023, pp. 1–22.
10 M. Peltola, C. Rose, J. V. Trueblood, S. Gray, M. Harvey and K. Sellegri, Chemical precursors of new particle formation in coastal New Zealand, *Atmos. Chem. Phys.*, 2023, **23**, 3955–3983.
11 J. Elm, J. Kubečka, V. Besel, M. J. Jääskeläinen, R. Halonen, T. Kurtén and H. Vehkamäki, Modeling the formation and growth of atmospheric molecular clusters: A review, *J. Aerosol Sci.*, 2020, **149**, 105621.
12 J. Elm, D. Ayoubi, M. Engsvang, A. B. Jensen, Y. Knattrup, J. Kubčeka, C. J. Bready, V. R. Fowler, S. E. Harold, O. M. Longsworth, *et al.*, Quantum chemical modeling of organic enhanced atmospheric nucleation: A critical review, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2023, **13**, e1662.
13 M. Engsvang, H. Wu, Y. Knattrup, J. Kubečka, A. B. Jensen and J. Elm, Quantum chemical modeling of atmospheric molecular clusters involving inorganic acids and methanesulfonic acid, *Chem. Phys. Rev.*, 2023, **4**, 031311.
14 A. M. Turing, Computing Machinery and Intelligence, *Mind*, 1950, **49**, 460.
15 J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller and A. Tkatchenko, Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems, *Chem. Rev.*, 2021, **121**, 9816–9872.
16 M. Meuwly, Machine Learning for Chemical Reactions, *Chem. Rev.*, 2021, **121**, 10218–10239.
17 D. Kuntz and A. K. Wilson, Machine learning, artificial intelligence, and chemistry: How smart algorithms are reshaping simulation and the laboratory, *Pure Appl. Chem.*, 2022, **94**, 1019–1054.
18 Y. Chen, S. V. Pios, M. F. Gelin and L. Chen, Accelerating Molecular Vibrational Spectra Simulations with a Physically Informed Deep Learning Model, *J. Chem. Theory Comput.*, 2024, **20**(11), 4703–4710.
19 A. R. Katritzky, M. Kuanar, S. Slavov, C. D. Hall, M. Karelson, I. Kahn and D. A. Dobchev, Quantitative Correlation of Physical and Chemical Properties with Chemical Structure: Utility for Prediction, *Chem. Rev.*, 2010, **110**, 5714–5789.
20 E. N. Muratov, J. Bajorath, R. P. Sheridan, I. V. Tetko, D. Filimonov, V. Poroikov, T. I. Oprea, I. I. Baskin, A. Varnek, A. Roitberg, *et al.*, QSAR without borders, *Chem. Soc. Rev.*, 2020, **49**, 3525–3564.
21 E. Lumiaro, M. Todorović, T. Kurten, H. Vehkamäki and P. Rinke, Predicting gas–particle partitioning coefficients of atmospheric molecules with machine learning, *Atmos. Chem. Phys.*, 2021, **21**, 13227–13246.
22 A. Gupta, M. Stulajter, Y. Shaidu, J. Neaton and W. de Jong, Equivariant Neural Networks Utilizing Molecular Clusters for Accurate Molecular Crystal Binding Energy Predictions, *ChemRxiv*, 2023, preprint, DOI: **10.26434/chemrxiv-2023-vk9gh**.
23 N. Hyttinen, A. Pihlajamäki and H. Häkkinen, Machine Learning for Predicting Chemical Potentials of Multifunctional Organic Compounds in Atmospherically Relevant Solutions, *J. Phys. Chem. Lett.*, 2022, **13**, 9928–9933.
24 S. Jiang, Y.-R. Liu, T. Huang, Y.-J. Feng, C.-Y. Wang, Z.-Q. Wang, B.-J. Ge, Q.-S. Liu, W.-R. Guang and W. Huang, Towards fully ab initio simulation of atmospheric aerosol nucleation, *Nat. Commun.*, 2022, **13**, 6067.
25 V. Besel, M. Todorović, T. Kurtén, H. Vehkamäki and P. Rinke, The search for sparse data in molecular datasets: Application of active learning to identify extremely low volatile organic compounds, *J. Aerosol Sci.*, 2024, **179**, 106375.
26 N. J. Browning, F. A. Faber and O. A. von Lilienfeld, GPU-accelerated approximate kernel method for quantum machine learning, *J. Chem. Phys.*, 2022, **157**, 214801.
27 A. S. Christensen, F. A. Faber, B. Huang, L. A. Bratholm, A. Tkatchenko, K. R. Muller and O. A. von Lilienfeld, QML: A Python Toolkit for Quantum Machine Learning, 2017, **https://github.com/qmlcode/qml**.
28 J. Kubečka, A. S. Christensen, F. R. Rasmussen and J. Elm, Quantum Machine Learning Approach for Studying Atmospheric Cluster Formation, *Environ. Sci. Technol. Lett.*, 2022, **9**, 239–244.
29 M. Engsvang, J. Kubečka and J. Elm, Toward Modeling the Growth of Large Atmospheric Sulfuric Acid–Ammonia Clusters, *ACS Omega*, 2023, **8**, 34597–34609.
30 J. Kubečka, I. Neefjes, V. Besel, F. Qiao, H. B. Xie and J. Elm, Atmospheric Sulfuric Acid-Multi-Base New Particle Formation Revealed through Quantum Chemistry

Enhanced by Machine Learning, *J. Phys. Chem. A*, 2023, **127**, 2091–2103.

31 J. Kubečka, Y. Knattrup, M. Engsvang, A. B. Jensen, D. Ayoubi, H. Wu, O. Christiansen and J. Elm, Current and future machine learning approaches for modeling atmospheric cluster formation, *Nat. Comput. Sci.*, 2023, **3**, 495–503.

32 Y. Knattrup, J. Kubečka, D. Ayoubi and J. Elm, Clusterome: A Comprehensive Data Set of Atmospheric Molecular Clusters for Machine Learning Applications, *ACS Omega*, 2023, **8**, 25155–25164.

33 J. Kubečka, V. Besel, I. Neefjes, Y. Knattrup, T. Kurtén, H. Vehkamäki and J. Elm, Computational Tools for Handling Molecular Clusters: Configurational Sampling, Storage, Analysis, and Machine Learning, *ACS Omega*, 2023, **8**, 45115–45128.

34 Y. Knattrup, J. Kubečka and J. Elm, Nitric Acid and Organic Acids Suppress the Role of Methanesulfonic Acid in Atmospheric New Particle Formation, *J. Phys. Chem. A*, 2023, **127**, 7568–7578.

35 J. G. Brandenburg, C. Bannwarth, A. Hansen and S. Grimme, B97-3c: A revised low-cost variant of the B97-D density functional method, *J. Chem. Phys.*, 2018, **148**, 064104.

36 S. Grimme, A. Hansen, S. Ehlert and J. Mewes, r2SCAN-3c: A "Swiss army knife" composite electronic-structure method, *J. Chem. Phys.*, 2021, **154**, 064103.

37 K. T. Schütt, O. T. Unke and M. Gastegger, *Equivariant Message Passing for the Prediction of Tensorial Properties and Molecular Spectra*, 2021.

38 K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko and K. Müller, SchNetPack: A Deep Learning Toolbox For Atomistic Systems, *J. Chem. Theory Comput.*, 2019, **15**, 448–455.

39 K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer and M. Gastegger, SchNetPack 2.0: A neural network toolbox for atomistic machine learning, *J. Chem. Phys.*, 2023, **158**, 144801.

40 J. V. Kildgaard, K. V. Mikkelsen, M. Bilde and J. Elm, Hydration of Atmospheric Molecular Clusters: A New Method for Systematic Configurational Sampling, *J. Phys. Chem. A*, 2018, **122**, 5026–5036.

41 F. R. Rasmussen, J. Kubečka, V. Besel, H. Vehkamäki, K. V. Mikkelsen, M. Bilde and J. Elm, Hydration of Atmospheric Molecular Clusters III: Procedure for Efficient Free Energy Surface Exploration of Large Hydrated Clusters, *J. Phys. Chem. A*, 2020, **124**, 5253–5261.

42 J. Kubečka, V. Besel, T. Kurtén, N. Myllys and H. Vehkamäki, Configurational Sampling of Noncovalent (Atmospheric) Molecular Clusters: Sulfuric Acid and Guanidine, *J. Phys. Chem. A*, 2019, **123**, 6022–6033.

43 V. Besel, J. Kubečka, T. Kurtén and H. Vehkamäki, Impact of Quantum Chemistry Parameter Choices and Cluster Distribution Model Settings on Modeled Atmospheric Particle Formation Rates, *J. Phys. Chem. A*, 2019, **124**, 5931–5943.

44 S. S. Xantheas and T. H. Dunning, AbInitio Studies of Cyclic Water Clusters $(H_2O)n$, n = 1-6. I. Optimal Structures and Vibrational Spectra, *J. Chem. Phys.*, 1993, **99**, 8774–8792.

45 S. S. Xantheas, AbInitio Studies of Cyclic Water Clusters $(H_2O)n$, n = 1-6. II. Analysis of Many-body Interactions, *J. Chem. Phys.*, 1994, **100**, 7523–7534.

46 S. S. Xantheas, AbInitio Studies of Cyclic Water Clusters $(H_2O)n$, n = 1-6. III. Comparison of Density Functional with MP2 Results, *J. Chem. Phys.*, 1995, **102**, 4505–4517.

47 B. Temelso, K. A. Archer and G. C. Shields, Benchmark Structures and Binding Energies of Small Water Clusters with Anharmonicity Corrections, *J. Phys. Chem. A*, 2011, **115**, 12034–12046.

48 D. Manna, M. K. Kesharwani, N. Sylvetsky and J. M. L. Martin, Conventional and Explicitly Correlated ab Initio Benchmark Study on Water Clusters: Revision of the BEGDB and WATER27 Data Sets, *J. Chem. Theory Comput.*, 2017, **13**, 3136–3152.

49 J.-D. Chai and M. Head-Gordon, Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections, *Phys. Chem. Chem. Phys.*, 2008, **10**, 6615–6620.

50 S. Grimme, C. Bannwarth and P. Shushkov, A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z = 1-86), *J. Chem. Theory Comput.*, 2017, **13**, 1989–2009.

51 P. Pracht, E. Caldeweyher, S. Ehlert and S. Grimme, A robust non-self-consistent tight-binding quantum chemistry method for large molecules, *ChemrXiv*, 2019, preprint, pp. 1–19. **https://chemrxiv.org/engage/chemrxiv/article-details/60c742abbdbb890c7ba3851a**.

52 J. Elm, Clusteromics I: Principles, Protocols and Applications to Sulfuric Acid – Base Cluster Formation, *ACS Omega*, 2021, **6**, 7804–7814.

53 J. Elm, Clusteromics II: Methanesulfonic Acid-Base Cluster Formation, *ACS Omega*, 2021, **6**, 17035–17044.

54 Y. Knattrup and J. Elm, Clusteromics IV: The Role of Nitric Acid in Atmospheric Cluster Formation, *ACS Omega*, 2022, **7**, 31551–31560.

55 D. Ayoubi, Y. Knattrup and J. Elm, Clusteromics V: Organic Enhanced Atmospheric Cluster Formation, *ACS Omega*, 2023, **8**, 9621–9629.

56 J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev and A. E. Roitberg, Less is more: Sampling chemical space with active learning, *J. Chem. Phys.*, 2018, **148**, 241733.

57 C. Bannwarth, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher and S. Grimme, Extended tight-binding quantum chemistry methods, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2020, e01493.

58 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, *et al.*, *Gaussian 16 Revision A.03*, Gaussian Inc., Wallingford CT, 2016.

59 F. Neese, The ORCA program system, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2012, **2**, 73–78.

60 F. Neese, Software update: The ORCA program system, version 4.0, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2018, **8**, e1327.
61 P. L. Houston, C. Qu, Q. Yu, P. Pandey, R. Conte, A. Nandi, J. M. Bowman and S. G. Kukolich, Formic Acid–Ammonia Heterodimer: A New Δ-Machine Learning CCSD(T)-Level Potential Energy Surface Allows Investigation of the Double Proton Transfer, *J. Chem. Theory Comput.*, 2024, **20**, 1821–1828.
62 J. M. Bowman, C. Qu, R. Conte, A. Nandi, P. L. Houston and Q. Yu, Δ-Machine Learned Potential Energy Surfaces and Force Fields, *J. Chem. Theory Comput.*, 2023, **19**, 1–17.
63 A. Nandi, C. Qu, P. L. Houston, R. Conte and J. M. Bowman, 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.
64 S. Wengert, G. Csányi, K. Reuter and J. T. Margraf, A Hybrid Machine Learning Approach for Structure Stability Prediction in Molecular Co-crystal Screenings, *J. Chem. Theory Comput.*, 2022, **18**, 4586–4593.
65 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. von Lilienfeld, Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach, *J. Chem. Theory Comput.*, 2015, **11**, 2087–2096.
66 K. T. Schütt, P. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko and K. Müller, *SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions*, 2017.
67 Y.-L. Liao and T. Smidt, Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs, *The Eleventh International Conference on Learning Representations*, 2023.
68 P. Thölke and G. D. Fabritiis, Equivariant Transformers for Neural Network based Molecular Potentials. *International Conference on Learning Representations*, 2022.
69 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinarim, T. E. Smidt and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nat. Commun.*, 2022, **13**, 1–11.
70 C. Darken, J. Chang and J. Moody, Learning rate schedules for faster stochastic gradient search, *Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop*, 1992, pp 3–12.
71 J. Xia, Y. Zhang and B. Jiang, Accuracy Assessment of Atomistic Neural Network Potentials: The Impact of Cutoff Radius and Message Passing, *J. Phys. Chem. A*, 2023, **127**, 9874–9883.
72 O. T. Unke and M. Meuwly, PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges, *J. Chem. Theory Comput.*, 2019, **15**, 3678–3693.
73 O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda and K.-E. Müller, SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects, *Nat. Commun.*, 2021, **12**, 7273.
74 J. Westermayr, S. Chaudhuri, A. Jeindl, O. T. Hofmann and R. J. Maurer, Long-range dispersion-inclusive machine learning potentials for structure search and optimization of hybrid organic–inorganic interfaces, *Digital Discovery*, 2022, **1**, 463–475.
75 D. Anstine, R. Zubatyuk and O. Isayev, AIMNet2: A Neural Network Potential to Meet your Neutral, Charged, Organic, and Elemental-Organic Needs, *ChemRxiv*, 2023, preprint, DOI: **10.26434/chemrxiv-2023-296ch**.
76 D. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.
77 H. Wu, M. Engsvang, Y. Knattrup, J. Kubečka and J. Elm, Improved Configurational Sampling Protocol for Large Atmospheric Molecular Clusters, *ACS Omega*, 2023, **8**, 45065–45077.
78 A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, *et al.*, The atomic simulation environment—a Python library for working with atoms, *J. Condens. Matter Phys.*, 2017, **29**, 273002.
79 S. R. Bahn and K. W. Jacobsen, An object-oriented scripting interface to a legacy electronic structure code, *Comput. Sci. Eng.*, 2002, **4**, 56–66.
80 X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli and T. Jaakkola, *Forces Are Not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations*. 2023.
81 M. Brehm, M. Thomas, S. Gehrke and B. Kirchner, TRAVIS—A free analyzer for trajectories from molecular simulation, *J. Chem. Phys.*, 2020, **152**, 164105.
82 M. Brehm and B. Kirchner, TRAVIS – A Free Analyzer and Visualizer for Monte Carlo and Molecular Dynamics Trajectories, *J. Chem. Inf. Model.*, 2011, **51**, 2007–2023.
83 D. R. Galimberti and J. Sauer, Chemically Accurate Vibrational Free Energies of Adsorption from Density Functional Theory Molecular Dynamics: Alkanes in Zeolites, *J. Chem. Theory Comput.*, 2021, **17**, 5849–5862.