Supporting Information

Accelerating Explicit Solvent Models of Heterogeneous Catalysts with Machine Learning Interatomic Potentials

Benjamin W. J. Chen*, Xinglong Zhang*, Jia Zhang

*Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #16–16 Connexis, Singapore 138632, Singapore*

*Corresponding authors: benjamin_chen@ihpc.a-star.edu.sg, zhang_xinglong@ihpc.a-star.edu.sg

Contents

S1. Generation of Training and Test Sets for Bulk Water

S1.1 Ab initio molecular dynamics (AIMD)

After a box of bulk water was set up, classical MD was first performed to equilibrate the solvent system, sampling the system with in the NVT ensemble. This was carried out using the GROMACS molecular dynamics package (version 2019.6)[1–6] with the Optimised Potential for Liquid Simulations (OPLS-AA) forcefield[7,8] with periodic boundary conditions (PBC). Initial system energy minimisation was carried out to remove any unphysical clashes in the initial guess structures (F_{max} < 10 kJ mol^{-1} nm^{-1}) or until machine precision. A 6 Å cut-off was applied for both short-range electrostatic and Van der Waals interactions.[9] Long-range electrostatics were treated with a fourth-order Particle Mesh Ewald (PME) method. The linear constraint solver algorithm (LINCS)[10] with constraints on H-bonds was used for simulations. The system temperature was kept constant at 300 K using the velocity rescaling method,[11] with a time constant of 100 fs. Initial random velocities were drawn from the Maxwell-Boltzmann distribution at 300 K. The system was equilibrated at constant volume (NVT), with a 1 fs timestep, for 50 ns.

Ab initio MD was performed with the method outlined in the Methods section (see main text). The last image from the end of the classical NVT equilibration was used as the starting configuration for AIMD. The AIMD NVT ensemble was sampled at a timestep of 1 fs for a total of 100 ps with the Nosé-Hoover thermostat.

S1.2 Generation of Training Datasets

Table S1 Summary of bulk water training and test sets for evaluation of the MLIPs.

Dataset	Number of molecules	Temp / K	Number of datapoints	Dataset description
T1	100	300	800	Every 3rd point of **T2**
T2		300	2400	800 points from each of 1V, 1.2V, 0.8V
E1		300	200	Every 3rd point of **E2**
E2		300	600	200 points from each of 1V, 1.2V, 0.8V
E3		473	1000	1V
E4	60	300	1000	1V
E5		473	1000	1V
E6	200	300	1000	1V
E7		473	1000	1V

We prepared bulk water in a cubic box at the experimental density, which we term 1V (1× volume) for simplicity. We additionally prepared water at reduced and increased densities corresponding to 1.2V and 0.8V, respectively, fixing the number of water molecules in the system at 100. The procedure outlined in Section S1.1 was carried out for bulk water at the three different densities (volumes) at 300 K. This generated total of 30000 datapoints calculated at RPBE-D3 level of theory, with 10000 points at each density.

We denote this dataset as **O1** (original dataset). From **O1**, a dataset of 1000 datapoints was constructed by taking every 30th datapoint. This prevents similar datapoints from being

represented too frequently. This dataset is split with an 80:20 ratio to give the **T1** training set with 800 datapoints (**T** for train) and the **E1** test set with 200 datapoints (**E** for error).

To glean insights into how dataset size affects the performance of different machine learning (ML) models, a second dataset for ML training was prepared. This larger dataset of 3000 datapoints was constructed by taking every 10th datapoint from **O1**. This dataset is split with an 80:20 ratio to give **T2** with 2400 datapoints and **E2** with 600 datapoints. **T1** and **T2** will be separately used to assess the performance of the different ML models.

S1.3 Generation of Test Datasets

Apart from test datasets **E1** and **E2**, we further prepared different datasets to test the ability of the different ML models to extrapolate to systems at different temperatures and with different amounts of water.

A system of 100 water molecules at experimental density (1V) was subjected to AIMD at a temperature of 473K (200ºC or twice the boiling point of water). This is to test the ability of ML models to extrapolate to configurations that are further away from equilibrium than those that were obtained at 300 K. The 10000 datapoints in this high temperature simulation were trimmed by taking every 10th datapoint to assemble a new test set, **E3**, with a total of 1000 datapoints.

To test the ability of the ML models to extrapolate to smaller systems, a system of 60 water molecules at experimental density (1V) was subjected to AIMD at 300 K and 473 K, separately. In each case, the 10000 datapoints generated were trimmed by taking every 10th datapoint to assemble a new test set, each with a total of 1000 datapoints. These are denoted as **E4** and **E5** for the data generated at 300 K and 473 K, respectively.

To test the ability of the ML models to extrapolate to larger systems, a system of 200 water molecules at experimental density (1V) was subjected to AIMD at 300 K and 473 K, separately. In each case, the 10000 datapoints generated were trimmed by taking every 10th datapoint to assemble a new test set, each with a total of 1000 datapoints. These are denoted as **E6** and **E7** for the data generated at 300 K and 473 K, respectively.

All the datasets described above are summarised in Table S1.

S2. Performance of Selected MLIPs

In this section, we first briefly describe the different MLIPs used in Section S2.1. In the Sections S2.2 to S2.5, we will then discuss the performance of ML potentials trained on the smaller dataset **T1** and on the larger dataset **T2**.

S2.1 Performance Measure via Root Mean Square Error

To measure the performance of each ML model, we calculated the Root Mean Square Error (RMSE) for the energy and force errors for each system as follows:

$$RMSE = \sqrt{\sum (y_{ML} - y_{DFT})^2 / n}$$

where y is either the energy or the force, subscripts "ML" and "DFT" refer to these values calculated using either the ML potential or DFT, respectively, and n is the number of configurations in the dataset. The RMSE for energy per atom is obtained by dividing the RMSE calculated above by the number of atoms in the system, and therefore tends to be smaller for large systems such as our simulations with 100 water molecules.

S2.2 Brief Overview of Selected MLIPs

Three machine learning interatomic potentials (MLIPs), namely, the moment tensor potential (MTP)[12], ANI,[13] and Schnet[14] were used to train and predict the quantum mechanical energy and forces of bulk solvent systems using the datasets constructed in Section S1. We herein briefly outline how these models work. Interested readers are encouraged to consult the original works for more details.

MTP is a systematically improvable, non-parametric interatomic potential based on linear regression using invariant polynomial basis functions. The underlying assumption is that the total energy of a structure can be approximated by the sum of the atomic potentials of all atoms in the structure, i.e.:

$$E^{\mathrm{mtp}} = \sum_{i=1}^{n} V\left(\mathfrak{n}_i\right)$$

where $V(n_i)$ is the local atomic potential of i^{th} atom that can be expanded as a linear combination of basis functions B_α,

$$\left(\mathfrak{n}_i\right) = \sum_{\alpha} \xi_\alpha B_\alpha\left(\mathfrak{n}_i\right)$$

.

ξ_α are parameters found by fitting to training data.

The basis functions B_α can be represented as moment tensors or moments (hence the name of this potential) containing both the radial and the angular parts for the description of atomic environments:

$$M_{\mu,\nu}\left(\mathfrak{n}_i\right) = \sum_{j} f_\mu\left(\left|r_{ij}\right|, z_i, z_j\right) \underbrace{\mathbf{r}_{ij} \otimes \ldots \otimes \mathbf{r}_{ij}}_{\nu \text{ times}};$$

where f_μ are functions that describe the radial part (with a hyperparameter N_Q defining the size of the radial basis, see original work[12,15]) and $\underbrace{\mathbf{r}_{ij} \otimes \ldots \otimes \mathbf{r}_{ij}}_{\nu \text{ times}}$ describes the angular part.

The parameters μ and ν together define the level of the moment via

$$\operatorname{lev} M_{\mu,\nu} = 2 + 4\mu + \nu,$$

giving another important hyperparameter of MTP model, $\mathrm{lev}_{\mathrm{max}}$. These two hyperparameters N_Q and $\mathrm{lev}_{\mathrm{max}}$ should be chosen to balance both the accuracy and cost of MTP.

Both Schnet and ANI are ML models based on neural networks. Schnet is a type of deep learning that aims to predict properties from molecular structures using graph neural networks. The molecular structures are encoded by both the nuclear charges (***Z***) and the interatomic distances (***D***). Atom types are described by a vector of coefficients, $\mathbf{c}_i$, while ***D*** is expanded to yield feature vectors encoding interatomic interactions between atoms *i* and *j*, $\mathbf{v}_{ij}$. The atom types are iteratively refined by interatomic interactions

$$\mathbf{c}_i^{(t+1)} = \mathbf{c}_i^{(t)} + \sum_{j \neq i} \mathbf{v}_{ij}$$

to map out local and neighboring atomic environments. The total energy is the sum of all predicted atomic energies.

ANI is a neural network based on modified Behler and Parrinello symmetry functions (BPSFs). The BPSFs are used to compute atomic environment vector (AEV) $\vec{G}_i^X$ which describes both the radial and angular features in atom *i*'s local environment via

$$\begin{aligned}\vec{G}_i^X &= G_m^{\mathrm{R}} * G_m^{\mathrm{A_{mod}}} \\ &= \left[\sum_{j \neq i}^{\mathrm{all}} \mathrm{e}^{-\eta(R_{ij}-R_\mathrm{s})^2} f_\mathrm{C}\left(R_{ij}\right)\right] * \left[2^{1-\zeta} \sum_{j,k \neq i}^{\mathrm{all}} \left(1+\cos\left(\theta_{ijk}-\theta_\mathrm{s}\right)\right)^\zeta \times \exp\left[-\eta\left(\frac{R_{ij}+R_{ik}}{2}-R_\mathrm{s}\right)^2\right]\right].\end{aligned}$$

The radial parameters η changes the width of the Gaussian functions and *Rs* shifts the centre of their peaks whereas the angular parameter ζ changes the width of the peaks in angular environment (Figure S1).

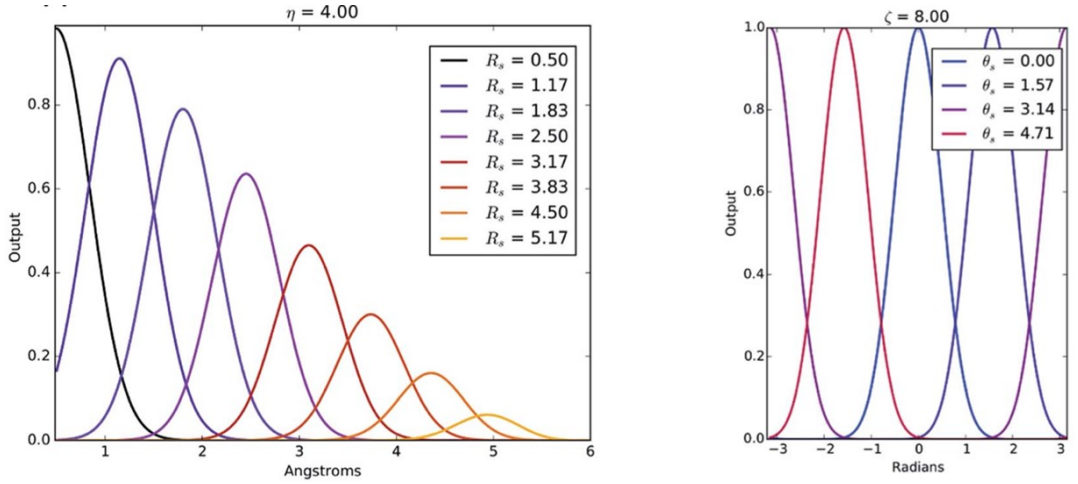

Figure S1 Examples of the symmetry functions used in ANI with different parameter sets for (left) radial symmetry functions, and (right) modified angular symmetry functions. Figure adapted from reference 13.

S2.3 Performance of Moment Tensor Potential (MTP)

For base case MTP training, Chebyshev polynomial basis functions were used. The minimal distance between atomic interactions in the training set, *min_dist*, was set to 0.8 Å, whereas the cutoff radius, *max_dist*, introduced to ensure a smooth potential when atoms leave or enter the

interaction neighborhood, was set to 5.0 Å. An energy weight of 1 and a force weight of 0.000167 were used. The size of the radial basis, N_Q, was set to 8. A total of 500 cycles of training iterations were performed.

S2.3.1 Number of basis functions

Keeping the base case settings constant, we tuned the value of lev_{max}, a key hyperparameter for MTP as it determines the number of basis functions used (Table S2). Independent of training set used, the energy per atom and force errors are consistent, with energy per atom error of the order of 10^{-4} eV and the force error of the order of 10^{-2} eV Å^{-1}. The energy and force errors decrease slightly with increasing lev_{max}. We settled on a value of lev_{max}=18, which provides a good balance of speed and accuracy.

Table S2 Screening of the lev_{max} hyperparameter for MTP. Energy per atom and force RMSEs are provided for MTP trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

MTP potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
lev_{max}	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
16	0.35	0.36	0.54	0.41	0.74	0.26	0.40
18	**0.28**	**0.27**	**0.48**	**0.36**	**0.60**	**0.21**	**0.39**
20	0.26	0.29	0.44	0.34	0.54	0.20	0.31
Force RMSEs / eV Å^{-1}							
16	0.055	0.055	0.067	0.050	0.066	0.052	0.068
18	**0.045**	**0.045**	**0.054**	**0.040**	**0.053**	**0.043**	**0.055**
20	0.046	0.056	0.041	0.055	0.043	0.057	0.046

S2.3.2 Replicate training with optimized hyperparameters

We present in Table S3 the results of three replicates of MTP trained with our optimized hyperparameters: lev_{max}=18, using the **T1** dataset.

Table S3 Energy per atom and force RMSEs for MTP potentials trained on **T1** dataset at three independent runs with optimized hyperparameters and tested on **E1** and **E3–7**. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

MTP potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Run	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1	0.28	0.27	0.48	0.36	0.60	0.21	0.39
2	0.27	0.27	0.53	0.35	0.63	0.22	0.46
3	0.25	0.27	0.64	0.37	0.69	0.23	0.56
Average	0.27	0.27	0.56	0.36	0.64	0.22	0.47
Std. Err.	0.01	0.00	0.09	0.01	0.03	0.01	0.05
Force RMSEs / eV Å^{-1}							
1	0.045	0.045	0.054	0.040	0.053	0.043	0.055
2	0.044	0.045	0.053	0.040	0.052	0.042	0.054
3	0.044	0.044	0.051	0.040	0.050	0.042	0.052
Average	0.045	0.045	0.053	0.040	0.052	0.042	0.054
Std. Err.	0.000	0.000	0.001	0.000	0.001	0.000	0.001

S2.3.3 MTP training using larger training dataset T2 (2400 datapoints)

To study the effect of training dataset size on the performance of MTP, we carried out similar benchmarking studies using a larger training dataset **T2** (2400 datapoints) We tuned the value of lev_{max} as before, giving the errors shown in Table S4. Similar to the case with the T1 dataset, we settled on a value of lev_{max}=18.

Table S4 Screening of the lev_{max} hyperparameter for MTP with the **T2** dataset. Energy per atom and force RMSEs are provided for MTP trained on **T2** (2400 datapoints) and tested on **E2–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for the definition of the datasets.

MTP potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
lev_{max}	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
16	0.34	0.33	0.66	0.43	0.70	0.25	0.53
18	**0.31**	**0.32**	**0.54**	**0.39**	**0.63**	**0.24**	**0.38**
20	0.26	0.28	0.40	0.32	0.52	0.21	0.29
Force RMSEs / eV Å^{-1}							
16	0.054	0.053	0.063	0.049	0.062	0.051	0.064
18	**0.049**	**0.048**	**0.058**	**0.044**	**0.057**	**0.046**	**0.059**
20	0.043	0.043	0.051	0.039	0.050	0.041	0.052

We additionally present in Table S5 the results of three replicates of MTP trained with our optimized hyperparameters: lev_{max}=18, using the **T1** dataset.

Table S5 Energy per atom and force errors for MTP potentials trained on **T2** datasets at three independent runs with lev_{max} = 18 and tested on **E2–7**. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

MTP potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
Run	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
1	0.31	0.32	0.54	0.39	0.63	0.24	0.38
2	0.28	0.29	0.58	0.38	0.65	0.22	0.45
3	0.31	0.33	0.78	0.41	0.79	0.00	0.64
Average	0.30	0.31	0.63	0.39	0.69	0.16	0.49
Std. Err.	0.01	0.01	0.07	0.01	0.05	0.08	0.08
Force RMSEs / eV Å^{-1}							
1	0.049	0.048	0.058	0.044	0.057	0.046	0.059
2	0.048	0.048	0.057	0.042	0.056	0.045	0.059
3	0.053	0.053	0.065	0.047	0.064	0.050	0.067
Average	0.050	0.050	0.060	0.045	0.059	0.047	0.061
Std. Err.	0.002	0.002	0.002	0.001	0.002	0.001	0.003

S2.4 Performance of Schnet

In the base case, Schnet training was performed with 6 interaction layers using a cosine cutoff function with a cutoff radius of 10.0 Å for atomic interactions. A total of 128 feature vectors for each atomic type and 50 Gaussian functions for interatomic distance expansion were used. An initial learning rate of 1e^{-4} with a minimal learning rate of 1e^{-6}, and a learning rate decay of 0.8 and learning rate patience of 25 was used. The tradeoff between energy and force, parameter ρ, for the overall cost function was set to 0.1. Atomic energy references for 'H' atom of -0.5

Hartree and for 'O' atom of -75.0645 Hartree were used. All models are trained on mini-batch stochastic gradient descent using the ADAM optimizer[16].

We first used the smaller training set, **T1**, to perform hyperparameter tuning, as for neural network (NN)-based ML models, the trained results can vary vastly on the hyperparameter choices.

S2.4.1 Batch size

The batch size is one of the key hyperparameters for NN-based MLIPs. Therefore, we first screened for the optimal batch size, using 1000 training epochs each for ease of comparison (Table S6). A batch size of 4 gives the best results overall. However, the force errors for both the training and test sets are one order of magnitude worse than the force prediction from MTP (Table S3). The energy per atom errors for the training set for batch sizes 4, 8, and 16 are of the same order of magnitude (but still larger) than the energy per atom error from MTP (Table S3) However, the test errors are 1–2 orders of magnitude worse than MTP, indicating Schnet's poorer ability to interpolate and extrapolate when only a small training set (800 datapoints) is used, as NN models are known to perform better with large datasets. We examine in Section S2.3.6 if training using the larger **T2** dataset with 2400 datapoints can help improve the test prediction.

Table S6 Screening of the batch size hyperparameter for Schnet. Energy per atom and force RMSEs are provided for Schnet potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

Schnet potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Batch Size	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
4	0.69	1.16	11.20	5.82	9.30	1.41	10.50
8	**0.72**	**1.20**	**11.20**	**6.14**	**8.54**	**2.16**	**11.40**
16	0.85	1.40	13.00	4.52	12.10	5.98	7.05
24	1.04	1.54	15.00	3.82	20.30	16.70	3.58
32	1.04	1.62	16.40	4.06	16.60	6.00	9.64
Force RMSEs / eV $Å^{-1}$							
4	0.278	0.299	0.378	0.325	0.397	0.284	0.361
8	**0.287**	**0.298**	**0.374**	**0.321**	**0.392**	**0.277**	**0.366**
16	0.312	0.318	0.401	0.337	0.414	0.300	0.420
24	0.348	0.353	0.455	0.372	0.468	0.338	0.453
32	0.357	0.361	0.473	0.379	0.480	0.343	0.474

S2.4.2 Number of epochs

Using a batch size of 8, we next examined the effect of the number of epochs on the performance of Schnet-trained potentials for energy per atom and force predictions (able S7). In mini-batch stochastic gradient descent, the method used by Schnet and ANI for cost function minimisation, this is the number of times that all the sub-samples have passed through the NN for training. Training for too few epochs would underfit the model, whereas training for too many epochs may overfit the model. Note that when we specified a limit of 1500 and 2000

epochs, the trained potentials converged after a total of 1326 and 1458 epochs, respectively. The energy per atom and force errors for the training and test sets improved marginally with an increasing number of epochs, although they are still of the same order of magnitude.

able S7 Screening of the max epochs hyperparameter for Schnet. Energy per atom and force RMSEs are provided for Schnet potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

Schnet potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Max Epochs	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1000	**0.72**	**1.20**	**11.20**	**6.14**	**8.54**	**2.16**	**11.40**
1500	0.72	1.19	10.80	7.51	8.32	2.56	10.90
2000	0.78	1.17	10.60	6.76	8.28	5.87	14.30
Force RMSEs / eV Å^{-1}							
1000	**0.287**	**0.298**	**0.374**	**0.321**	**0.392**	**0.277**	**0.366**
1500	0.285	0.297	0.374	0.320	0.391	0.276	0.285
2000	0.282	0.296	0.372	0.319	0.392	0.276	0.362

S2.4.3 Learning rate

The learning rate controls the step size for gradient descent along the loss function. A large learning rate may approach the cost function minimum faster, but the model may miss the minimum on the loss function and oscillate about this minimum. On the other hand, if a learning rate is too small, the model has a higher possibility of reaching the minimum but at a much longer training time. With a batch size of 8 and 1000 training epochs, we further tested two other values, 1×10^{-3} and 1×10^{-5}, in addition to the initial learning rate (lr) of 1×10^{-4} used (Table S8). The results suggest that altering the initial learning rates does not significantly improve the training results.

Table S8 Screening of the initial learning rate hyperparameter for Schnet. Energy per atom and force RMSEs are provided for Schnet potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

Schnet potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Initial Learning Rate	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1×10^{-3}	1.90	2.08	21.80	4.16	21.50	1.83	22.00
1×10^{-4}	**0.72**	**1.20**	**11.20**	**6.14**	**8.54**	**2.16**	**11.40**
1×10^{-5}	1.79	2.24	18.20	22.80	3.96	40.90	58.40
Force RMSEs / eV Å^{-1}							
1×10^{-3}	0.642	0.644	0.862	0.648	0.849	0.668	0.879
1×10^{-4}	**0.287**	**0.298**	**0.374**	**0.321**	**0.392**	**0.277**	**0.366**
1×10^{-5}	0.533	0.535	0.701	0.566	0.764	0.515	0.665

S2.4.4 Other Hyperparameters

Table S9 Screening of other miscellaneous hyperparameters for Schnet: the energy force tradeoff parameter, atomic reference values, the number of Gaussian functions, and the number of feature vectors. Energy per atom and force RMSEs are provided for Schnet potentials trained on **T1** and tested on **E1** and **E3–7**. None of these hyperparameter values tested below were used as the base case values were found to be better. See Table S1 for definitions of the datasets.

Schnet potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Hyperparameter	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
ρ							
0.05	0.76	1.22	11.00	5.98	8.98	2.08	7.23
0.15	0.69	1.20	12.00	5.84	9.95	2.14	12.10
Atomic Reference							
Null	0.75	1.17	11.60	5.45	9.58	1.31	10.70
Pseudopotentials	0.69	1.20	11.40	5.56	9.19	2.23	11.50
Number of Gaussian Functions							
100	0.71	1.24	11.70	4.99	9.84	3.26	12.90
Number of Feature Vectors							
256	0.72	1.19	12.50	4.95	11.00	1.51	12.70
Force RMSEs / eV $Å^{-1}$							
ρ							
0.05	0.284	0.295	0.371	0.319	0.389	0.277	0.395
0.15	0.291	0.301	0.379	0.322	0.399	0.277	0.362
Atomic Reference							
Null	0.287	0.298	0.373	0.321	0.394	0.278	0.361
Pseudopotentials	0.286	0.297	0.374	0.320	0.393	0.275	0.366
Number of Gaussian Functions							
100	0.284	0.299	0.376	0.321	0.395	0.275	0.359
Number of Feature Vectors							
256	0.281	0.300	0.380	0.324	0.398	0.282	0.365

Using a batch size of 8 and 1000 training epochs, we further tested the effect of (1) varying the tradeoff between energy and force errors, ρ; (2) using different atomic reference values; (3) using different number of Gaussian functions; and (4) using different number of feature vectors. (Table S10).

Note that the atomic reference values are removed from the target property by the offset transforms in the AtomWise neural network training module during training and added back to the prediction after training. We tested "Null" and "Pseudopotential" values for the elements. For "Null", Schnet defaults to applying no such offset transforms. The values from

"Pseudopotential" calculation of each single atom at the same level of theory can also be obtained and used as atomic reference.

We note that the test errors are not significantly improved by changing the above hyperparameters.

S2.4.5 Replicate training with optimized hyperparameters

In summary, the best set of hyperparameters we obtained were: batch size of 8, 1000 training epochs, learning rate of 1×10^{-4}, learning rate decay of 0.8, learning rate patience of 25, energy-force tradeoff ρ of 0.1, atomic reference of -0.500 eV for H atoms and -75.0645 eV for O atoms, 50 Gaussian functions, 128 features, and 6 interaction layers. Using these optimized hyperparameters, three Schnet potentials were independently trained to obtain the averaged errors (Table S10).

Table S10 Energy per atom and force RMSEs for three independently trained Schnet potentials on the **T1** dataset with optimized hyperparameters and tested on **E1** and **E3–7**. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

Schnet potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Run	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1	0.72	1.20	11.20	6.14	8.54	2.16	11.40
2	0.71	1.19	11.90	4.88	11.20	1.20	11.30
3	0.78	1.24	11.00	5.03	10.40	4.63	4.97
Average	0.74	1.21	11.40	5.35	10.10	2.66	9.23
Std. Err.	0.02	0.02	0.29	0.40	0.80	1.02	2.13
Force RMSEs / eV Å^{-1}							
1	0.287	0.298	0.374	0.321	0.392	0.277	0.366
2	0.293	0.302	0.381	0.323	0.400	0.277	0.372
3	0.288	0.298	0.371	0.322	0.393	0.278	0.358
Average	0.289	0.299	0.375	0.322	0.395	0.278	0.365
Std. Err.	0.002	0.001	0.003	0.001	0.003	0.000	0.004

S2.4.6 Schnet training using larger training dataset T2 (2400 datapoints)

With the best set of hyperparameters from tuning Schnet using T1 dataset, we assessed the performance of Schnet using T2 dataset as train set. The effect of different batch sizes was assessed again, due to the different number of datapoints used in training (Table S11). A batch size of 24 produces the best energy per atom errors although in general, both the energy per atom and force errors are of the same order of magnitude for the different batch sizes tested. Three replicates for a batch size of 24 were run to obtain the averaged errors (Table S12).

Table S11 Screening of the batch size hyperparameter for Schnet. Energy per atom and force RMSEs are provided for Schnet potentials trained on **T2** and tested on **E2–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

Schnet potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
Batch Size	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
4	0.42	0.80	10.00	6.61	7.41	1.17	10.10
8	0.43	0.77	9.25	7.53	6.25	2.42	11.20
16	0.44	0.79	10.60	6.00	8.16	2.52	10.80
24	**0.67**	**0.96**	**12.70**	**5.80**	**11.60**	**2.12**	**8.60**
32	0.56	0.91	11.80	6.24	10.30	7.25	7.66
Force RMSEs / eV Å^{-1}							
4	0.268	0.286	0.374	0.324	0.394	0.285	0.360
8	0.270	0.285	0.371	0.321	0.392	0.280	0.357
16	0.284	0.291	0.368	0.317	0.388	0.274	0.366
24	**0.298**	**0.302**	**0.382**	**0.324**	**0.401**	**0.279**	**0.393**
32	0.301	0.305	0.384	0.326	0.397	0.291	0.460

Table S12 Energy per atom and force RMSEs for Schnet potentials trained on **T2** dataset at three independent runs with optimized hyperparameters and tested on **E2–7**. Averages and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

Schnet potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
Run	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
1	0.67	0.96	12.70	5.80	11.60	2.12	8.60
2	0.71	0.97	12.40	6.88	9.28	2.20	12.80
3	0.62	0.91	12.60	6.17	9.17	4.29	9.20
Average	0.66	0.95	12.60	6.28	10.00	2.87	10.20
Std. Err.	0.03	0.02	0.08	0.32	0.79	0.71	1.31
Force RMSEs / eV Å^{-1}							
1	0.298	0.302	0.382	0.324	0.401	0.279	0.393
2	0.295	0.300	0.374	0.322	0.398	0.279	0.354
3	0.295	0.300	0.376	0.322	0.395	0.285	0.388
Average	0.296	0.301	0.377	0.323	0.398	0.281	0.378
Std. Err.	0.001	0.001	0.002	0.001	0.002	0.002	0.012

S2.5 Performance of ANI

In the base case, ANI training was performed with an initial learning rate of 1e^{-3} and an early stopping learning rate of 1e^{-5}. A weight decay of 1e^{-4} and a dropout rate of 0.0 were used. The radial cutoff radius, R_{CR}, was set to 5.2 Å, while the angular cutoff radius, R_{CA}, was set to 3.5 Å. Eta_R = 16.0 Å (η in equation 3 in reference 13) was used.

S2.5.1 Batch size

ANI potentials were trained for 1000 epochs with different batch sizes (Table S13). Both the energy per atom and force errors for the training and test sets for all different batch sizes fall on the same order of magnitude. Overall, the force errors are about one order of magnitude worse than the force predictions from MTP (Table S3). A batch size of 16 gives the best errors for both energy and force predictions.

Table S13 Screening of the batch size hyperparameter for ANI. Energy per atom and force RMSEs are provided for ANI potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

ANI potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Batch Size	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
4	2.61	3.01	4.55	3.96	4.77	3.74	4.63
8	1.23	1.39	2.26	1.26	2.48	0.73	2.25
16	**1.15**	**1.15**	**1.47**	**1.47**	**1.89**	**1.29**	**1.03**
24	1.45	1.56	8.12	1.68	7.97	1.40	8.20
32	1.67	1.78	7.21	2.06	6.89	1.24	7.15
64	1.32	1.51	8.77	1.88	8.58	1.07	9.02
128	1.22	2.02	21.30	2.86	20.90	1.24	21.10
Force RMSEs / eV $Å^{-1}$							
4	0.161	4.500	1.380	0.173	0.943	0.153	3.760
8	0.354	0.367	0.461	0.356	0.455	0.367	0.461
16	**0.182**	**0.183**	**0.304**	**0.176**	**0.331**	**0.183**	**0.334**
24	0.473	0.476	0.646	0.461	0.642	0.478	0.870
32	0.527	0.530	0.897	0.571	0.865	0.559	0.979
64	0.567	0.567	0.737	0.573	0.724	0.597	0.752
128	0.488	0.640	0.785	0.653	0.770	0.681	0.809

S2.5.2 Number of epochs

Using a batch size of 16, we next examined the effect of the number of training epochs (Table S14). While the train errors generally improved with increasing number of epochs, the prediction errors do not. This is especially so for the force errors, which become much poorer in test sets **E5** and **E7**, for example, with an increasing number of epochs, indicating overfitting of the models.

Table S14 Screening of the max epochs hyperparameter for ANI. Energy per atom and force RMSEs are provided for ANI potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

ANI potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Max Epochs	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1000	**1.15**	**1.15**	**1.47**	**1.47**	**1.89**	**1.29**	**1.03**
2000	0.85	1.00	6.58	1.29	3.15	0.74	3.65
5000	0.43	9.54	5.87	4.26	1.16	4.16	13.80

Force RMSEs / eV Å^{-1}							
1000	**0.182**	**0.183**	**0.304**	**0.176**	**0.331**	**0.183**	**0.334**
2000	0.394	0.408	2.300	0.402	1.670	0.435	2.320
5000	0.108	3.740	5.740	2.200	0.176	3.630	6.830

S2.5.3 Weight decay

Weight decay is a regularisation technique to reduce the complexity of the fitting function by adding a term that penalises the complex functions to the overall cost function. The complexity of a function $f(\mathbf{x})=\mathbf{w}^{\top}\mathbf{x}$ can be measured by some norm of its weight vector, e.g., $\|\mathbf{w}\|^2$. Thus, one can add the norm of the weight vector to the cost function for overall minimization:

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^{\top}\mathbf{x}^{(i)} + b - y^{(i)}\right)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 .$$

By doing so, we have replaced the original goal of minimising the prediction loss on the training dataset, with new goal of minimising the sum of prediction loss and the penalty term. Now, if the weight vector becomes too large, i.e., the fitting function becomes too complex, then the ML algorithm will minimise the weight norm $\|\mathbf{w}\|^2$ (decrease function complexity), rather than minimising the training error.

In addition to the original weight decay value of 1×10^{-4}, we further tested a range of different values (Table S15). The errors for energy per atom and forces are similar across all weight decay values, indicating that weight decay does not have a significant effect on the quality of the trained ANI potential.

Table S15 Screening of the weight decay hyperparameter for ANI. Energy per atom and force RMSEs are provided for ANI potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

ANI potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Decay	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
0	1.76	1.79	3.10	1.73	3.00	1.28	2.77
1×10^{-4}	**1.15**	**1.15**	**1.47**	**1.47**	**1.89**	**1.29**	**1.03**
1×10^{-3}	1.96	1.97	6.20	1.59	5.99	0.96	6.09
1×10^{-2}	1.38	1.57	7.17	1.82	6.99	0.90	7.23
1×10^{-1}	1.26	1.39	7.94	1.84	8.00	1.05	8.07
Force RMSEs / eV Å^{-1}							
0	0.251	0.252	1.480	0.244	0.811	0.347	2.090
1×10^{-4}	**0.182**	**0.183**	**0.304**	**0.176**	**0.331**	**0.183**	**0.334**
1×10^{-3}	0.422	0.424	0.929	0.420	0.929	0.434	0.825
1×10^{-2}	0.499	0.529	0.761	0.496	1.150	0.584	0.868
1×10^{-1}	0.464	0.465	0.544	0.469	0.532	0.485	0.556

S2.5.4 Dropout rate

Applying dropout in a NN ML model involves randomly choosing a neuron and then leave it out of training, ignoring both the inputs and outputs at that neuron. When a dropout is applied, other neurons have to take the missing neuron's place to learn the representations of the system. This gives multiple independent internal representations learned by a collection of 'trimmed' networks. In doing so, the network becomes less sensitive to the weights of a particular neuron or group of neurons, allowing it to better generalise and avoid overfitting.

Using a batch size of 16 and 1000 training epochs, we examined the effect of the different dropout rates (Table S16). With increasing dropout rate, the errors for energy per atom and forces become worse, indicating that dropout does not help to improve the ANI potential.

Table S16 Screening of the dropout hyperparameter for ANI. Energy per atom and force RMSEs are provided for ANI potentials trained on **T1** and tested on **E1** and **E3–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

ANI potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Dropout	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
0.0	**1.15**	**1.15**	**1.47**	**1.47**	**1.89**	**1.29**	**1.03**
0.1	1.31	1.35	6.48	1.53	6.80	0.77	6.70
0.5	1.73	1.98	16.10	2.50	16.90	1.42	15.50
0.8	1.87	1.95	23.30	2.91	23.10	1.20	23.20
Force RMSEs / eV Å^{-1}							
0.0	**0.182**	**0.183**	**0.304**	**0.176**	**0.331**	**0.183**	**0.334**
0.1	0.471	0.474	0.826	0.478	1.610	0.498	0.680
0.5	1.020	0.967	2.610	0.628	3.330	0.654	2.980
0.8	0.659	0.675	1.060	0.679	1.020	0.708	1.060

S2.5.5 Replicate training with optimized hyperparameters

Table S17 Energy per atom and force RMSEs for ANI potentials trained on **T1** dataset at three independent runs with optimized hyperparameters and tested on **E1** and **E3–7**. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

ANI potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
Run	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
1	1.15	1.15	1.47	1.47	1.89	1.29	1.03
2	1.74	1.80	1.42	1.37	2.17	0.98	1.24
3	1.97	2.01	1.36	2.29	1.79	1.98	0.95
Average	1.62	1.65	1.41	1.71	1.95	1.42	1.07
Std. Err.	0.25	0.26	0.03	0.29	0.11	0.29	0.09
Force RMSEs / eV Å^{-1}							
1	0.182	0.183	0.304	0.176	0.331	0.183	0.334
2	0.247	0.246	0.420	0.243	1.270	0.418	0.890
3	0.194	0.197	0.318	0.187	0.339	0.195	0.329
Average	0.208	0.209	0.348	0.202	0.647	0.265	0.518
Std. Err.	0.020	0.019	0.037	0.021	0.312	0.076	0.186

To summarize, the best set of optimized hyperparameters is a batch size of 16, training epochs of 1000, dropout rate of 0.0, initial learning rate of 1×10^{-4}, early stopping learning rate of 1×10^{-5}, and a weight decay of 1×10^{-4}. Using these optimized hyperparameters, we independently trained three separate ANI potentials to obtain the averaged errors (Table S17).

S2.5.6 ANI training using larger training dataset T2 (2400 datapoints)

With the best set of hyperparameters from tuning ANI training using **T1** dataset, we assessed the performance of ANI potential from using **T2** dataset as train set. The effect of batch sizes was assessed again, due to the different number of datapoints used in training (Table S18). A batch size of 32 and 64 produces the best energy per atom errors and a batch size of 64 produces the best force errors. Three replicates for a batch size of 64 were run to obtain the averaged errors (Table S19).

Table S18 Screening of the batch size hyperparameter for ANI. Energy per atom and force RMSEs are provided for ANI potentials trained on **T2** and tested on **E2–7**. The selected hyperparameter value is highlighted with bold blue font. See Table S1 for definitions of the datasets.

ANI potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
Batch Size	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
16	0.97	1.02	1390.00	1.34	382.00	0.80	267.00
32	1.29	1.28	3.41	1.34	9.06	0.97	3.93
64	**2.21**	**2.18**	**5.33**	**1.54**	**5.39**	**0.95**	**5.35**
128	1.53	1.55	10.60	2.02	10.30	1.21	10.80
Force RMSEs / eV Å^{-1}							
16	0.416	1.110	685.000	0.398	279.000	0.424	388.000
32	0.307	0.382	3.090	0.297	3.130	0.308	3.900
64	**0.381**	**0.380**	**0.505**	**0.379**	**0.499**	**0.394**	**0.514**
128	0.575	0.575	0.776	0.575	0.767	0.599	0.795

Table S19 Energy per atom and force RMSEs for ANI potentials trained on **T2** dataset at three independent runs with optimized hyperparameters and tested on **E2–7**. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

ANI potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
Run	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
1	2.21	2.18	5.33	1.54	5.39	0.95	5.35
2	1.27	1.23	1.45	1.55	1.94	1.07	1.17
3	1.13	1.20	6.57	1.35	6.66	0.80	6.51
Average	1.54	1.53	4.45	1.48	4.66	0.94	4.34
Std. Err.	0.34	0.32	1.54	0.07	1.41	0.08	1.62
Force RMSEs / eV Å^{-1}							
1	0.381	0.380	0.505	0.379	0.499	0.394	0.514
2	0.317	0.317	0.392	0.300	0.395	0.310	0.481
3	0.427	0.448	0.718	0.417	0.705	0.435	0.838
Average	0.375	0.382	0.539	0.365	0.533	0.380	0.611
Std. Err.	0.032	0.038	0.096	0.035	0.091	0.037	0.114

S2.6 Comparison of MTP, Schnet, and ANI performance

Here, we compare the performance results of the three MLIPs after hyperparameter tuning (Table S20). Each MLIP was trained three times separately on **T1** and **T2**. MTP performs the best out of the three MLIPs for both datasets.

Table S20 Energy per atom and force RMSEs for MTP, Schnet and ANI potentials with optimized hyperparameters (top) trained on **T1** and tested on **E1** and **E3–7**, and (bottom) trained on **T2** datasets and tested on **E2–7**. Three independent runs were conducted for each potential for each training set. Average and standard errors of the three runs are also given. See Table S1 for definitions of the datasets.

ML potential trained on T1 (800 datapoints)							
Energy per atom RMSEs / meV							
ML Model	**T1**	**E1**	**E3**	**E4**	**E5**	**E6**	**E7**
MTP	0.27 ± 0.01	0.27 ± 0.00	0.34 ± 0.17	0.36 ± 0.01	0.64 ± 0.03	0.22 ± 0.01	0.47 ± 0.05
Schnet	0.74 ± 0.02	1.21 ± 0.02	11.40 ± 0.29	5.35 ± 0.40	10.10 ± 0.80	2.66 ± 1.02	9.23 ± 2.13
ANI	1.62 ± 0.25	1.65 ± 0.26	1.41 ± 0.03	1.71 ± 0.29	1.95 ± 0.11	1.42 ± 0.29	1.07 ± 0.09
Force RMSEs / eV $Å^{-1}$							
MTP	0.045 ± 0.000	0.045 ± 0.000	0.053 ± 0.001	0.040 ± 0.000	0.052 ± 0.001	0.042 ± 0.000	0.054 ± 0.001
Schnet	0.289 ± 0.002	0.299 ± 0.001	0.375 ± 0.003	0.322 ± 0.001	0.395 ± 0.003	0.278 ± 0.000	0.365 ± 0.004
ANI	0.208 ± 0.020	0.209 ± 0.019	0.348 ± 0.037	0.202 ± 0.021	0.647 ± 0.312	0.265 ± 0.076	0.518 ± 0.186
ML potential trained on T2 (2400 datapoints)							
Energy per atom RMSEs / meV							
ML Model	**T2**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**
MTP	0.30 ± 0.01	0.31 ± 0.01	0.63 ± 0.07	0.39 ± 0.01	0.69 ± 0.05	0.16 ± 0.08	0.49 ± 0.08
Schnet	0.66 ± 0.03	0.95 ± 0.02	12.60 ± 0.08	6.28 ± 0.32	10.00 ± 0.79	2.87 ± 0.71	10.20 ± 1.31
ANI	1.54 ± 0.34	1.53 ± 0.32	4.45 ± 1.54	1.48 ± 0.07	4.66 ± 1.41	0.94 ± 0.08	4.34 ± 1.62
Force RMSEs / eV $Å^{-1}$							
MTP	0.050 ± 0.002	0.050 ± 0.002	0.060 ± 0.002	0.045± 0.001	0.059 ± 0.002	0.047 ± 0.001	0.061 ± 0.003
Schnet	0.296 ± 0.001	0.301 ± 0.001	0.377 ± 0.002	0.323 ± 0.001	0.398 ± 0.002	0.281 ± 0.002	0.378 ± 0.012
ANI	0.375 ± 0.032	0.382 ± 0.038	0.539 ± 0.096	0.365 ± 0.035	0.533 ± 0.091	0.380 ± 0.037	0.611 ± 0.114

In Figure S2, we plot the energy and force errors for using the training results from the three replicate runs. We did not observe any significant improvement in the performance of ANI and Schnet by using a larger **T2** training set of 2400 datapoints, as compared with the smaller **T1** training set of 800 datapoints. This potentially results from the fact that the configurations from **T2** may be similar to those in **T1**, which results from taking every 3rd point of **T2**. Training sets with more structurally diverse configurations, which are correspondingly most costly to generate, might therefore be necessary to improve the performance of ANI and Schnet.

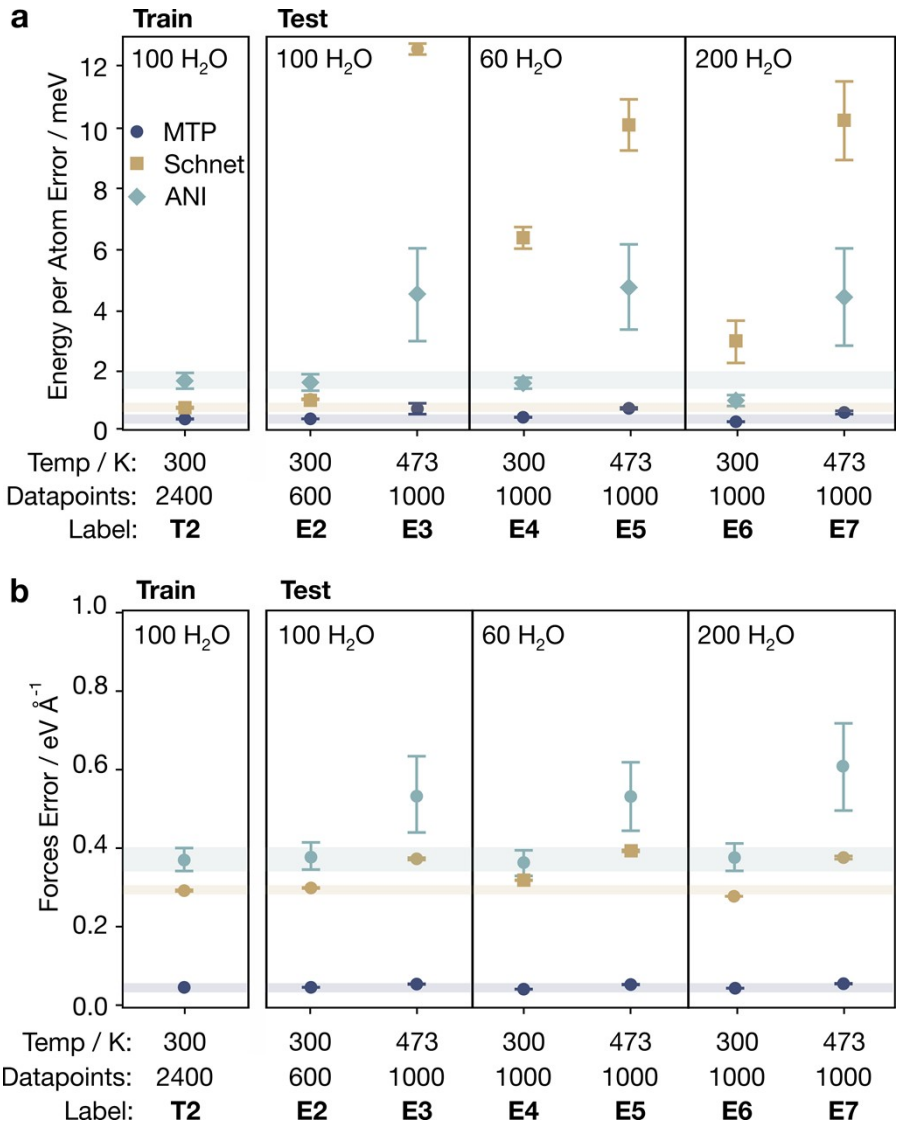

Figure S2 Train and test errors for MTP, Schnet, and ANI trained potentials using the larger dataset **T2** with individually-optimized hyperparameters. (a) Root mean square errors (RMSEs) of the energy per atom. (b) RMSEs of the force components. Error bars represent standard errors of each ML potential from triplicate runs. The training set (**T2**) consists of configurations of 100 H_2O molecules at 300 K, whereas the test sets (**E2**–**E7**) consist of configurations with varying number of H_2O molecules (100, 60 and 200) at different temperatures (300 K and 473 K), to probe extrapolation ability (see Table S1 for details of each training and test set). Pale horizontal bars across all panels indicate the test set errors to aid comparison.

S3. Determining the threshold MV grade, γ_{thres}

The MV grade, γ, is related to the condition number of the training matrix, and therefore predicts the degree of extrapolation of a configuration compared with the configurations in the training set.[17] In our active-learning MLaMD simulations, we constantly monitor the value of γ for configurations encountered in the simulations to determine if they should be selected for training. While a constant threshold of $2 < \gamma_{thres} < 10$ is typically used for this purpose[15], in this work we use an adaptive method for determining γ_{thres}. Specifically, we define γ_{thres} as follows:

$$\gamma_{thres} = min(3\sigma_\gamma + \bar{\gamma}, \gamma_{DFT}),$$

where $\bar{\gamma}$ is the average γ over the past n evaluations of γ, σ_γ is the standard deviation of the past n evaluations, and γ_{DFT} is the γ of the last DFT-evaluated configuration.

This adaptive method therefore checks for abnormally high γ values out of the γ values recently obtained, which is indicative of an extrapolated configuration in the current region that the MLaMD simulation is exploring. Yet, the MV grade is not allowed to be greater than that of the last DFT evaluated configuration, which prevents γ_{thres} from increasing without bound.

The main advantage of this scheme is to increase the efficiency of active learning MLaMD simulations by reducing the number of false positives: configurations with $\gamma > \gamma_{thres}$ but yet having low DFT errors. Such false positives may occur as γ does not correlate perfectly with DFT errors. For example, it is empirically found that a configuration with a force error of 0.3 eV Å^{-1} could have a γ between 2–15.[17] A too low fixed value of γ_{thres} may therefore catch too many false positives, whereas a too high fixed value of γ_{thres} may lead to an inaccurate simulation. An adaptive threshold assumes that the correlation between γ and DFT errors is better when considering configurations similar to each other. Our scheme therefore attempts to set γ_{thres} appropriately based on the region of configuration space the MLaMD simulation is in, so as to only trigger DFT evaluations when absolutely necessary.

Crucially, we note that this adaptive threshold does not significantly affect the accuracy of our MLaMD simulations, as noted by their low errors with respect to ab initio calculations (Section S3).

S4. Accuracy of MLaMD Simulations for Heterogeneous Catalysts

In this section, we assess the accuracy of our MLaMD simulations for predicting adsorption energies at water-metal interfaces. We had showcased the adsorption of CO* and OH* on Cu(111) in the main text; to test the ability of our method to extend to different facets and a wider range of adsorbates, we additionally performed simulations of 5 more adsorbates—namely OH*, CO*, COH*, HCO*, and OCCHO*—over solvated Cu(211) (Figure S3). The Cu(211) slabs were modelled as 3×1 unit cells with 12 layers, with the bottom 6 layers fixed at their optimized bulk lattice constants of 3.57 Å for Cu (experimental[18]: 3.61). The Brillouin zone was sampled by a Generalized Regular grid with 4 irreducible and 9 reducible k-points, as generated by the autoGR package.[19] All other calculation parameters are the same as for the Cu(111) slabs (see Methods section in the main text).

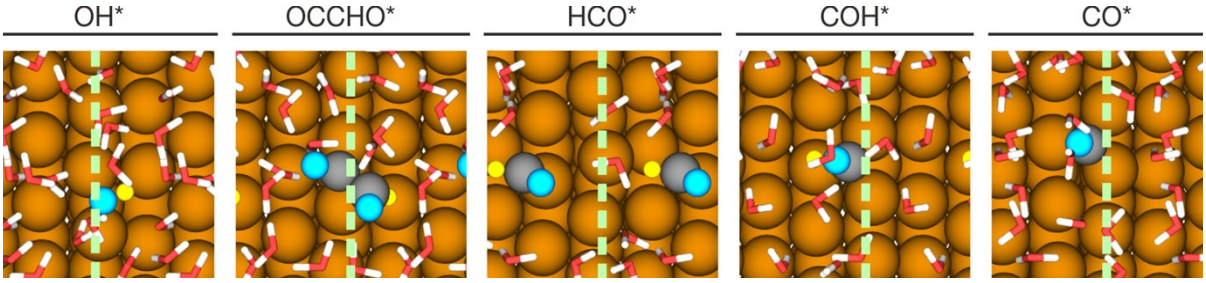

Figure S3 Top view atomic illustrations of various adsorbates on explicitly solvated Cu(211). Structures were sampled from MLaMD simulations. Water molecules are rendered in a licorice representation, other atoms are represented as spheres. Dashed light green lines mark the (211) step edges. Color code: brown–Cu, red–O, white–H, grey–C, cyan–O atoms of adsorbates, yellow–H atom of OH.

To holistically assess the accuracy of the MLaMD simulations for all 7 systems (2 adsorbates on Cu(111) + 5 adsorbates on Cu(211)), we performed three types of analyses, as elaborated in more detail in the subsections below

S4.1 Ab initio Evaluation of Sampled Configurations from MLaMD Trajectories

We sampled the 8 production simulations of a MLaMD workflow (Figure S6) at intervals of 25 ps each, creating a test set of 160 configurations for each system. Single point calculations were then performed to obtain the DFT energies and forces of each configuration, which were compared with the energies and forces obtained from MLaMD simulations. The energetic and force RMSEs and MAEs are shown in Figure **S4** and Figure **S5**, respectively, and are also tabulated in Table S21. This test makes sure that accurate energies can be obtained throughout the entire MLaMD simulation, which is crucial since the trajectory consists of predictions from multiple trained MTP models due to the active learning process wherein the MTP is retrained upon introduction of new data.

Encouragingly, we find good agreement of the MLaMD energies and forces (RMSEs of ~1 meV atom^{-1}) and forces (RMSEs of ~0.06 eV Å) with the single point DFT calculations. This indicates that the MLaMD simulations are of similar accuracy as AIMD simulations, and that the MLIPs were able to fit the training configurations well without overfitting.

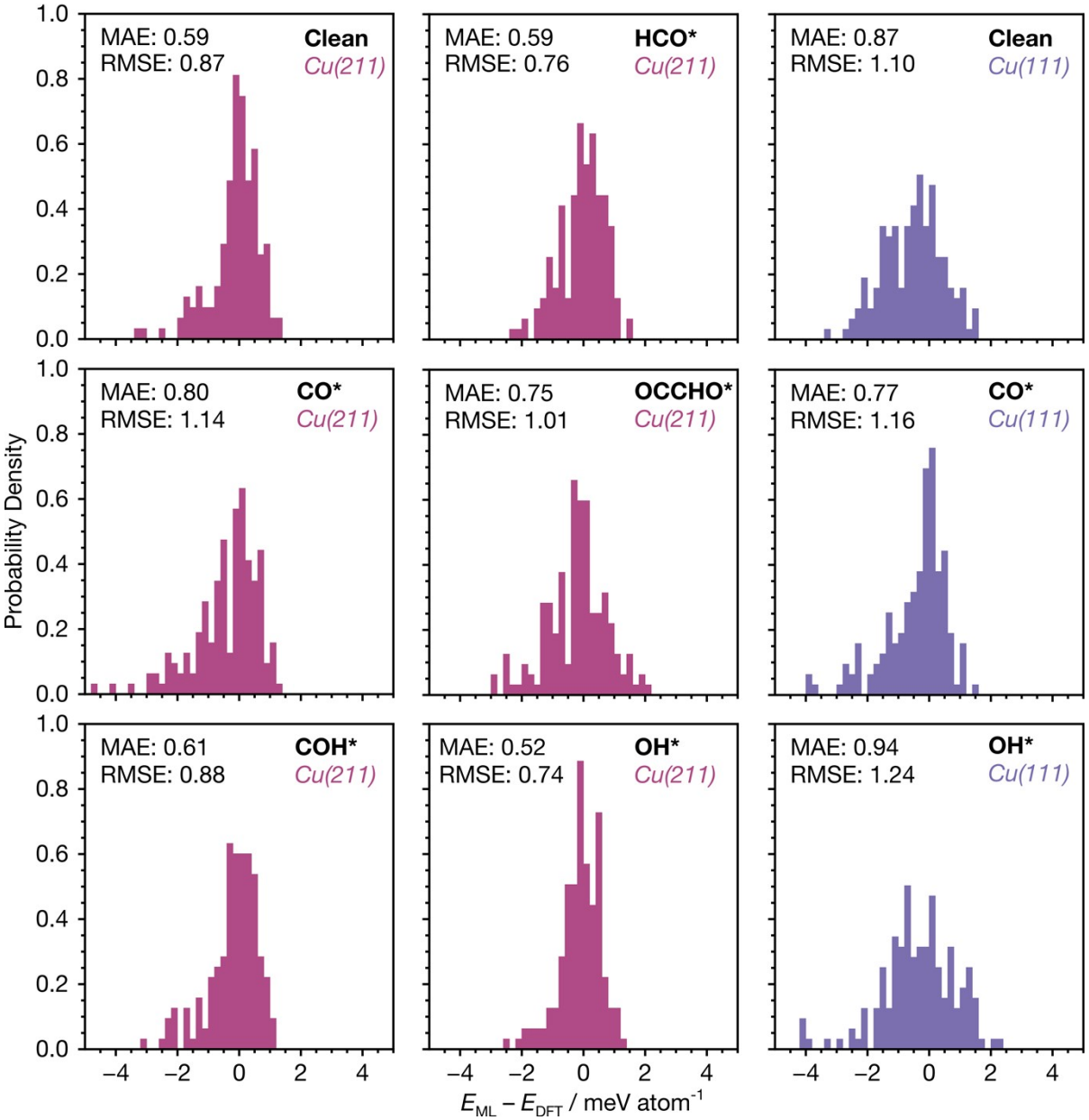

Figure S4 Normalized histograms of energetic errors in MLaMD simulations of solvated adsorbates on Cu(111) and Cu(211) surfaces as compared with single point ab initio calculations. For each system, 160 configurations were obtained by taking 20 samples at 25 ps intervals from each of the eight 500 ps replicas in the production runs of a single MLaMD workflow (Figure S6). The MLaMD per atom energies (E_{ML}) were then compared with those from the ab initio calculations (E_{DFT}) to obtain the histograms for each system. MAEs and RMSEs for each system are provided.

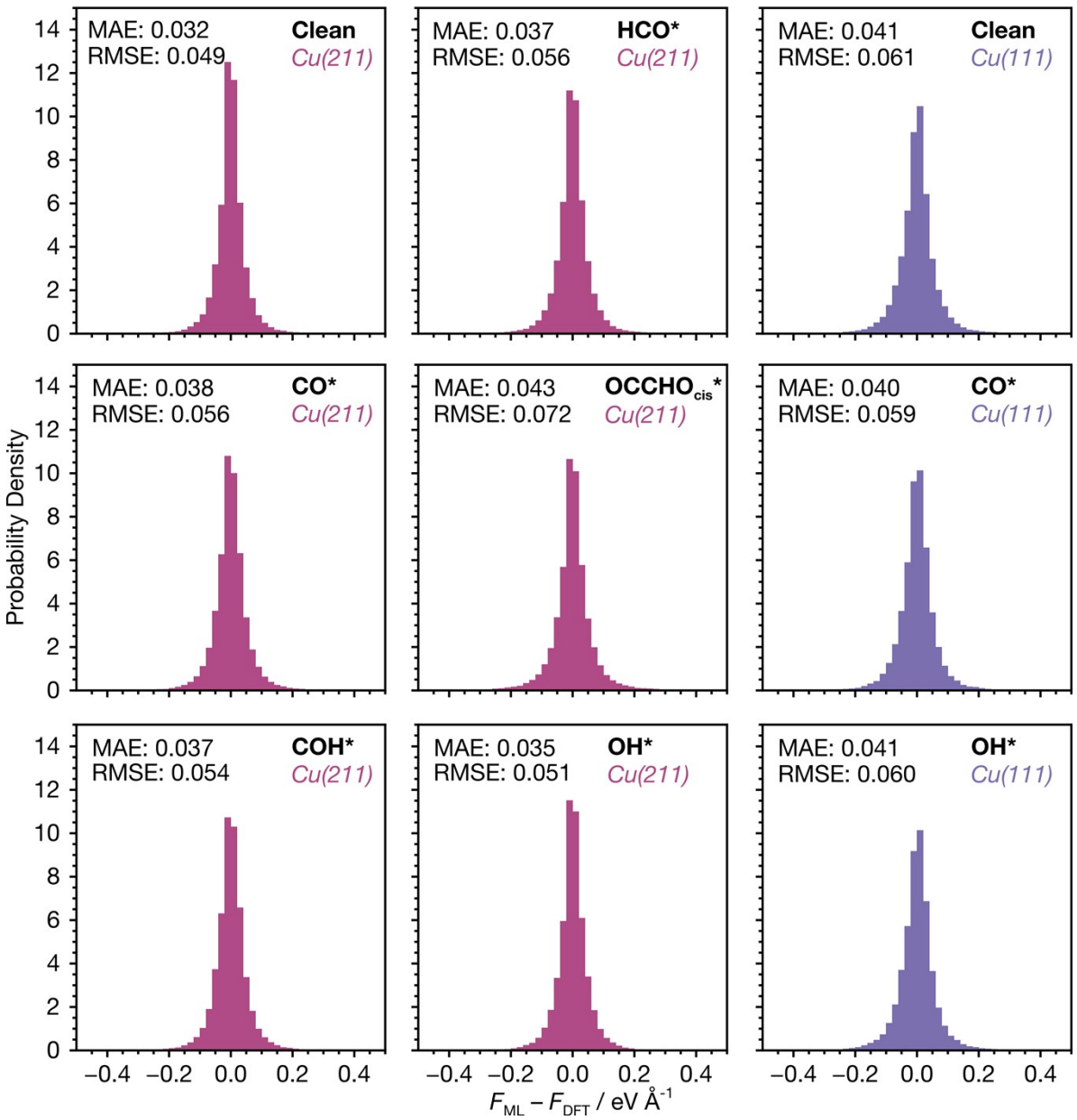

Figure S5 Normalized histograms of force component errors in MLaMD simulations of adsorbates on Cu(111) and Cu(211) surfaces as compared with single point ab initio calculations. For each system, 160 configurations were obtained by taking 20 samples at 25 ps intervals from each of the eight 500 ps replicas in the production runs of a single MLaMD workflow (Figure S6). The MLaMD force components (F_{ML}) were then compared with those from the ab initio calculations (F_{DFT}) to obtain the histograms for each system. MAEs and RMSEs for each system are provided.

Table S21 Energy per atom and force root mean square errors (RMSEs) and mean absolute errors (MAEs) for MLaMD simulations of adsorbates on Cu(111) and Cu(211) surfaces. For each system, 160 configurations were obtained by taking 20 samples at 25 ps intervals from each of the eight 500 ps replicas in the production runs of a single MLaMD workflow (Figure S6). Histograms of the energetic and force errors are given in Figure **S4** and Figure **S5**, respectively.

Adsorbate	Energies		Forces	
	RMSE / meV atom^{-1}	MAE / meV atom^{-1}	RMSE / eV Å^{-1}	MAE / eV Å^{-1}
Cu(111)				
Clean	1.10	0.87	0.061	0.041
CO*	1.16	0.77	0.059	0.040
OH*	1.24	0.94	0.060	0.041
Cu(211)				
Clean	0.87	0.59	0.049	0.032
CO*	1.14	0.80	0.056	0.038
COH*	0.88	0.61	0.054	0.037
HCO*	0.76	0.59	0.056	0.037
OCCHO*	1.01	0.75	0.072	0.043
OH*	0.74	0.52	0.051	0.035

S4.2 MTP Training Errors

We additionally directly evaluated the ability of the MTP model to fit the configurations encountered in the MLaMD simulations by collating the training errors with the training set built up from the active learning in the MLaMD simulations (Table S22).

Table S22 MTP training set energy and force root mean square errors (RMSEs) and mean absolute errors (MAEs) of the last trained potential for MLaMD simulations of adsorbates on Cu(111) and Cu(211) surfaces. Values shown are for one of the three workflow replicates conducted for each system. The number of training set configurations is also provided.

Adsorbate	Number of Training Configurations	Energies		Forces	
		RMSE / meV atom^{-1}	MAE / meV atom^{-1}	RMSE / eV Å^{-1}	MAE / eV Å^{-1}
Cu(111)					
Clean	392	0.88	0.61	0.071	0.046
CO*	525	0.72	0.55	0.062	0.042
OH*	322	0.70	0.55	0.062	0.042
Cu(211)					
Clean	353	0.55	0.42	0.056	0.037
CO*	561	0.66	0.50	0.064	0.041
COH*	556	0.62	0.46	0.061	0.041
HCO*	588	0.65	0.50	0.066	0.042
OCCHO*	660	0.79	0.62	0.083	0.048
OH*	439	0.63	0.47	0.062	0.041

We again find good agreement of the DFT energies and forces with the MTP predicted energies (RMSEs of ~0.8 meV atom^{-1}) and forces (RMSEs of ~0.06 eV Å^{-1}. This indicates that MTP is flexible enough to accurately fit the wide range of configurations encountered in our entire simulation.

S4.3 Comparison of Catalytic Quantities with the Literature

Lastly, we compared the predictions from our MLaMD simulations with those from the literature, specifically, from AIMD simulations in the excellent work by Heenan et al.[20] (Table S23). Two key quantities were compared: (1) adsorption energies, and (2) the number of hydrogen bonds formed with the adsorbate.

Table S23 Comparison of key properties predicted by MLaMD versus AIMD simulations (ref. 20), including adsorption energies and the number of hydrogen bonds formed. For the literature values, uncertainties for the adsorption energies are standard errors over *n* runs, whereas uncertainties for the number of hydrogen bonds formed are standard deviations over *n* runs. For the MLaMD simulations, uncertainties for adsorption energies are the standard errors of the difference between the mean energies of the clean and adsorbate+slab simulations with *n* runs each (note that uncertainties for the gas-phase adsorbate simulations are negligible), whereas the uncertainties for the number of hydrogen bonds formed are standard deviations over *n* runs. *n*=3 for the MLaMD values (i.e., 3 MLaMD workflow replicates), and *n*=4 for the literature values (i.e., 4 AIMD simulations).

System	Adsorption Energies / eV		Number of Hydrogen Bonds Formed	
	MLaMD	Literature	MLaMD	Literature
Cu(111)				
CO*	–0.62 ± 0.09	–0.77 ± 0.06	0.20 ± 0.08	0.32 ± 0.03
OH*	–0.45 ± 0.11	–0.11 ± 0.05	2.88 ± 0.06	2.39 ± 0.10
Cu(211)				
CO*	–0.67 ± 0.20	–0.69 ± 0.05	0.14 ± 0.04	0.28 ± 0.05
COH*	0.05 ± 0.21	0.31 ± 0.05	1.38 ± 0.12	1.77 ± 0.27
HCO*	–0.25 ± 0.21	–0.35 ± 0.05	1.38 ± 0.17	1.16 ± 0.13
OCCHO*	–0.65 ± 0.15	–0.99 ± 0.08	3.14 ± 0.11	2.98 ± 0.67
OH*	–0.67 ± 0.19	–0.29 ± 0.03	1.87 ± 0.09	2.04 ± 0.21

We find generally good agreement in both quantities for all adsorbates on both Cu(211) and Cu(111). One notable exception is OH*, for which the difference in adsorption energies predicted from the MLaMD and literature are 0.34 and 0.38 eV on Cu(111) and Cu(211), respectively. As discussed in detail in Section 2.3.2 of the main text, the MLaMD simulations predict stronger binding of OH* due to better sampling of the configurational space. This leads to lower energy states being accessed in the longer timescales of the MLaMD simulations (500 ps), as compared with the shorter timescales of the AIMD simulations (~30 ps). OH* is more affected than other adsorbates due to its strong hydrogen bonding network, which leads to lower mobility of water[20] and therefore slower energetic convergence.

Another noticeable discrepancy is for OCCHO*/Cu(211), where there is a 0.34 eV difference between the MLaMD and AIMD simulations. We believe that this difference is again due to the better sampling of our MLaMD simulations. As seen in Figure S10, Cu(211) systems in general exhibit behaviour similar to the OH*/Cu(111) system, with water restructuring events occurring at long simulation times. The Cu(211) systems therefore show slower energetic convergence than the Cu(111) systems (Figure S9). This slower convergence is due to the small Cu(211) unit cell used, which is especially narrow in the *x*-direction (6.18 Å × 10.09 Å). The

small unit cell hinders the mobility of water, as the flexibility of water structures is reduced as a result of the artificially enforced periodicity of the simulations.[21]

The longer timescales and increased number of replicas for our MLaMD simulations (8 replicas per MLaMD bundle) provide better sampling than the AIMD simulations (4 replicas) and are therefore more accurate estimates of the adsorption energies. The discrepancy of 0.34 eV is also within the range of energies of individual simulations, which are around 0.5 eV for both the MLaMD simulations (Figure S10) and the AIMD simulations by Heenan et al.[20] This is consistent with our hypothesis that the discrepancy may be caused by the AIMD simulations having poorer sampling and averaging over fewer replicas.

S5. Other Supplementary Figures

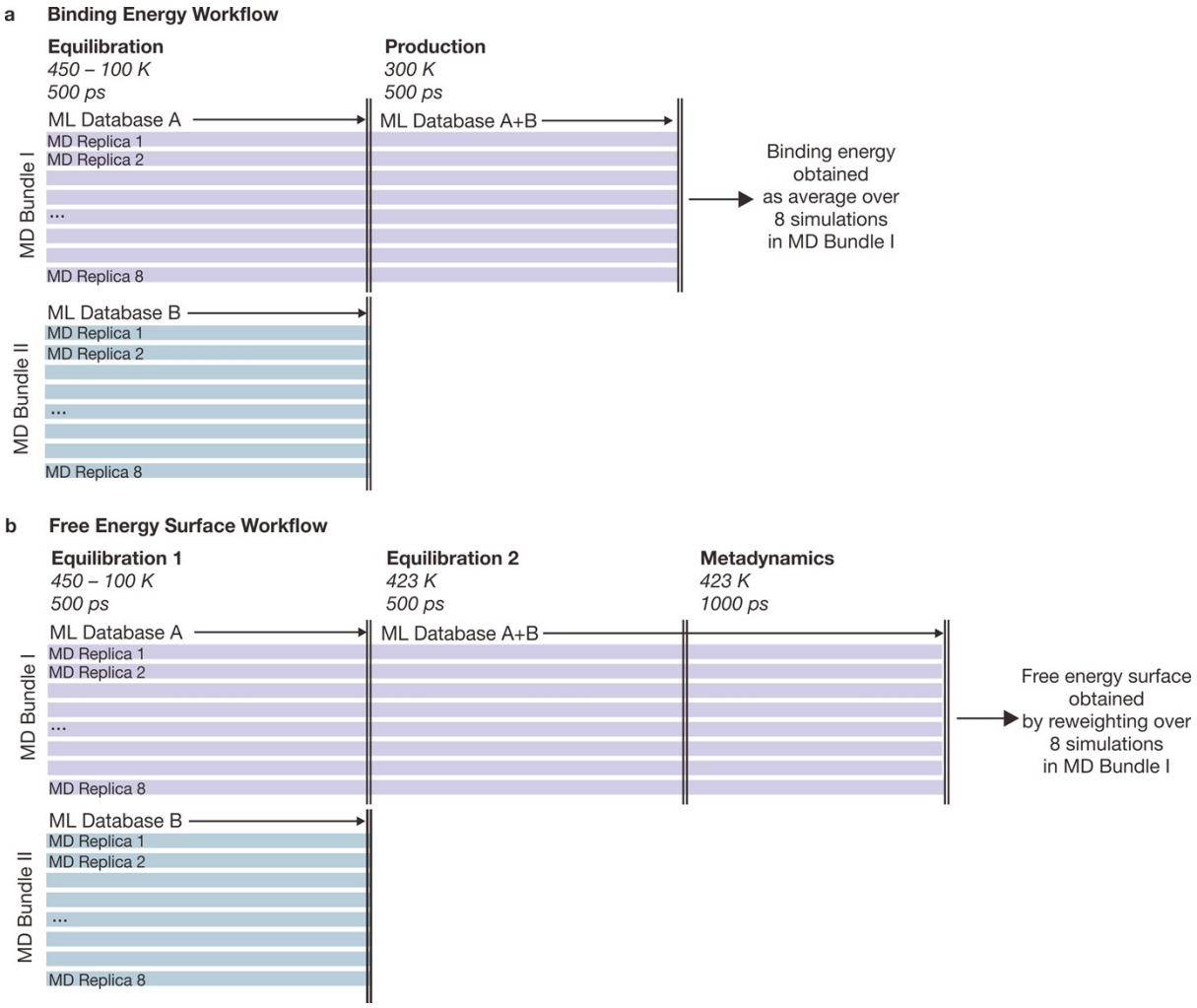

Figure S6 Scheme of workflows for calculating (a) binding energies and (b) free energy surfaces using MLaMD simulations. See Methods Section in main text for more detailed descriptions of each procedure.

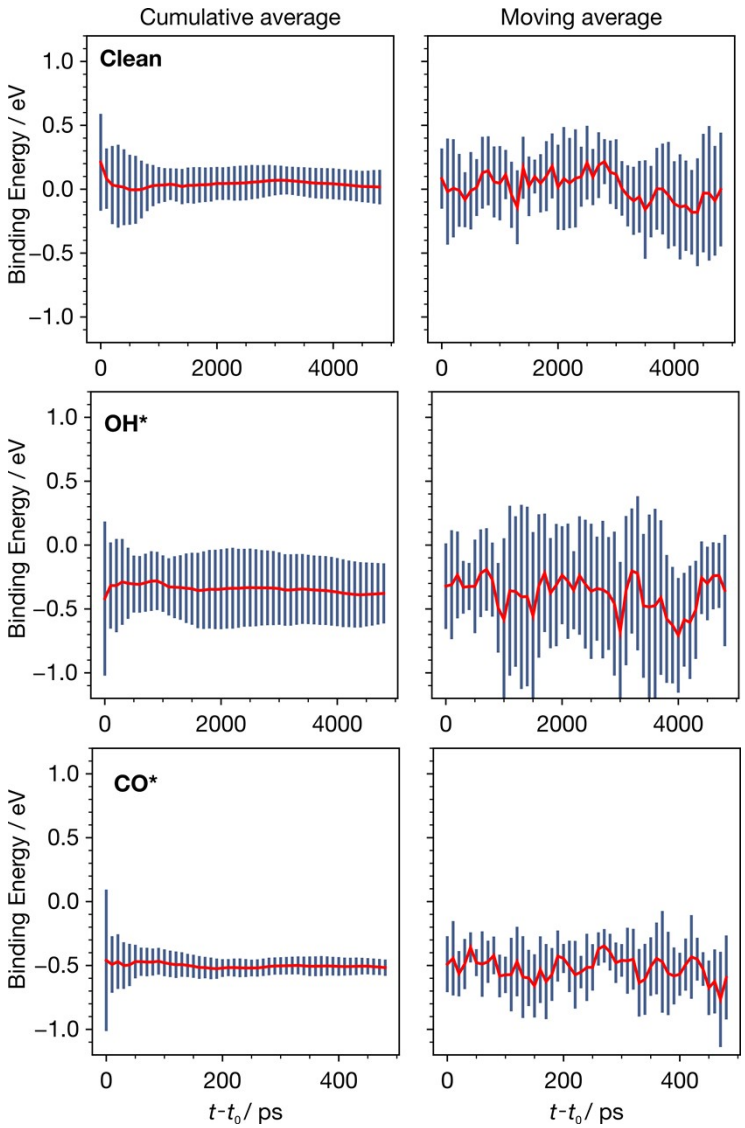

Figure S7 Evolution of the energetics of MLaMD trajectories versus time for clean Cu(111), OH*/Cu(111) and CO*/Cu(111). (left) Cumulative average of the energies. (right) Moving average of the energies with a window of 10 ps. Red lines indicate the energies whereas blue bars indicate error bars across 8 replicas. Energies are binding energies with respect to Cu(111), CO(g), H_2O(g), and H_2(g), where "(g)" indicates a gas-phase species. Times are zeroed to t_0=10 ps, which is the initial discarded portion of the trajectory.

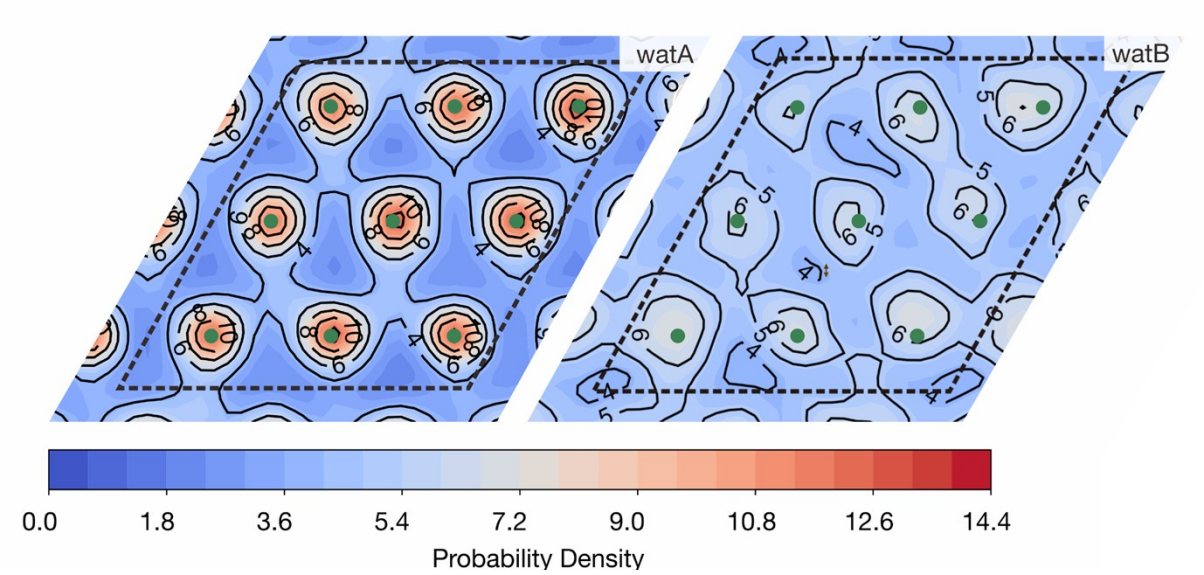

Figure S8 Probability density of watA and watB molecules in the *xy*-plane (i.e., top view of the system) within 4.6 Å of the Cu(111) surface. Small green circles and dashed black lines mark the positions of the Cu atoms and the unit cell, respectively.

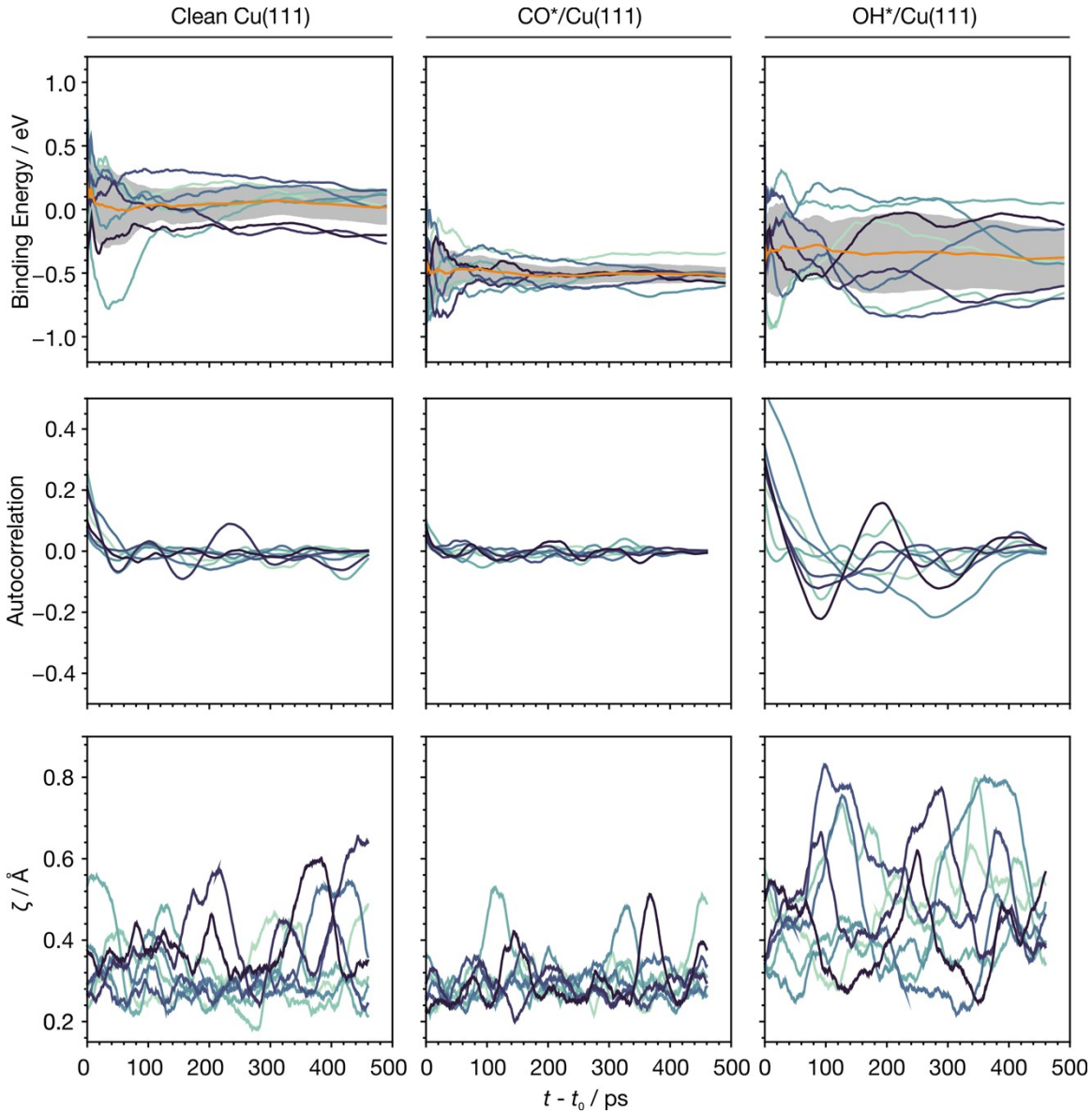

Figure S9 Evolution of MLaMD simulations versus time for Cu(111) systems. (top) Moving average of binding energies smoothed over a window of 10 ps. Binding energies are with respect to clean Cu(111), CO(g), H_2O(g), and H_2(g), where "(g)" indicates a gas-phase species. (middle) Moving average of the energy autocorrelation function smoothed over a window of 30 ps. (bottom) Moving average of ζ smoothed over a window of 30 ps. The simulation time, t, is zeroed to t_0=10 ps, which is the initial discarded portion of the trajectory. All 8 runs of the MLaMD bundle are shown.

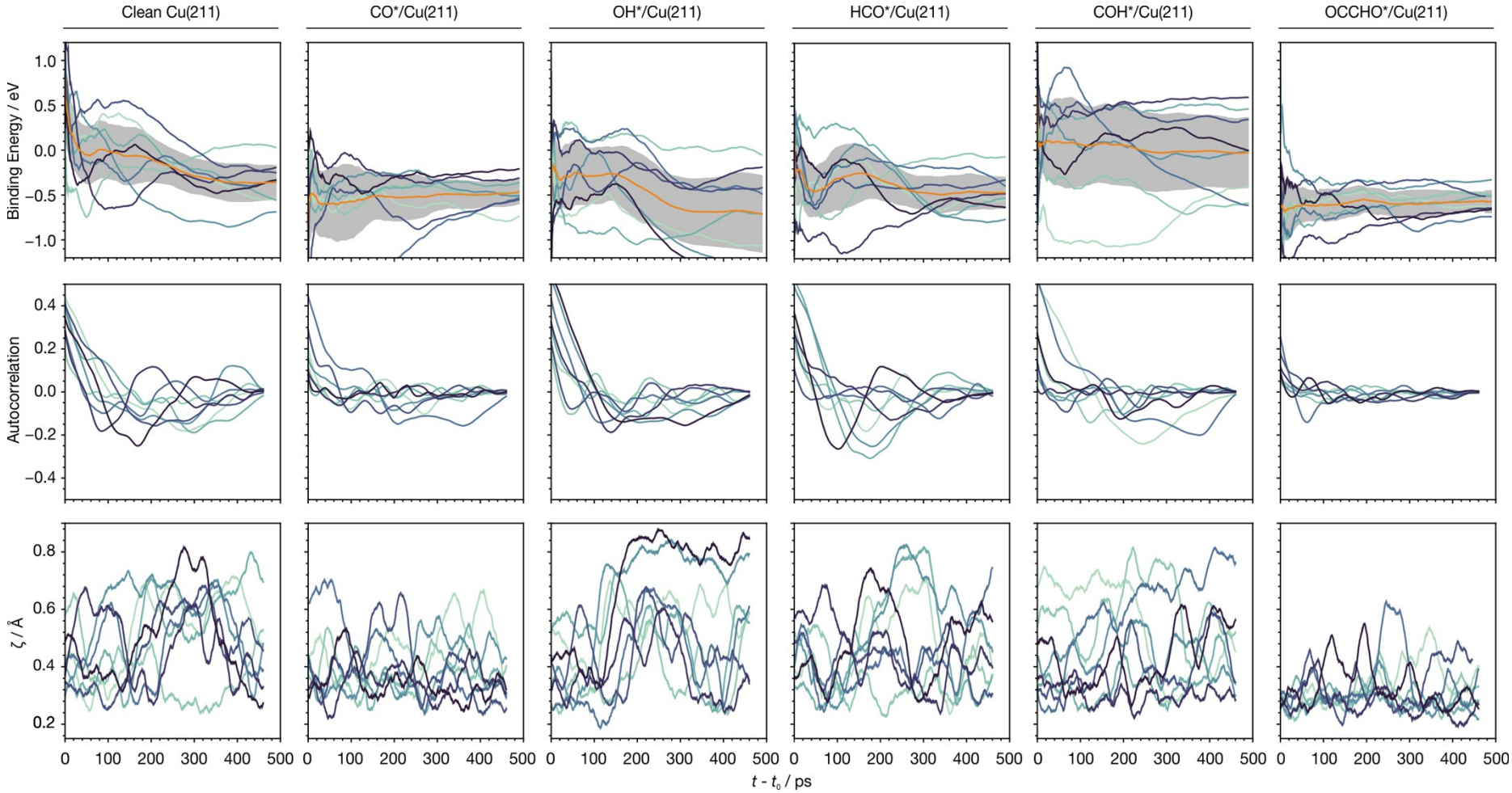

Figure S10 Evolution of MLaMD simulations versus time for Cu(211) systems. (top) Moving average of binding energies smoothed over a window of 10 ps. Binding energies are with respect to clean Cu(111), CO(g), H_2O(g), and H_2(g), where "(g)" indicates a gas-phase species. (middle) Moving average of the energy autocorrelation function smoothed over a window of 30 ps. (bottom) Moving average of ζ smoothed over a window of 30 ps. The simulation time, t, is zeroed to t_0=10 ps, which is the initial discarded portion of the trajectory. All 8 runs of the MLaMD bundle are shown.

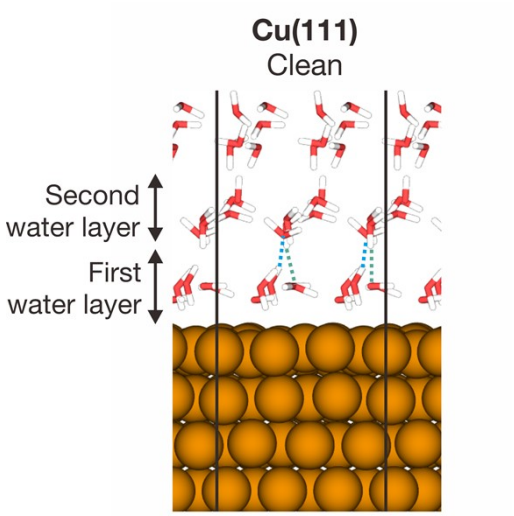

Figure S11 Side view of the hydrogen bonding between the first and second water layer on clean Cu(111). Blue and green dotted lines mark the donation and acceptance of hydrogen bonding, respectively, by the first water layer from the second water layer. Solid black lines mark the unit cell. Color code: brown–Cu, red–O, white–H, grey–C, cyan–O atoms of adsorbates, yellow–H atom of OH.

S6. Other Supplementary Tables

Table S24 Calculation of the speed up of MLaMD simulations as compared with AIMD simulations for adsorption energy calculations over solvated Cu(111). One full workflow for calculating a binding energy consists of two equilibration bundles of 8 MD replicates each, and one production bundle with 8 MD replicates (see Figure S6 and Methods in main text for more details). The speed up is calculated as the number of DFT calculations required for each MLaMD workflow divided by the 12,000,000 timesteps involved for an entire workflow (1,500,000 per replicate, 8 replicates each).

System	Number of DFT calculations			Speed-Up
	Equilibration Bundle 1	Equilibration Bundle 2	Production Bundle	
Clean Cu(111)				
Workflow 1	194	287	134	19512
Workflow 2	192	226	136	21660
Workflow 3	200	189	119	23622
CO/Cu(111)*				
Workflow 1	269	317	182	15625
Workflow 2	230	246	187	18100
Workflow 3	281	371	170	14599
OH/Cu(111)*				
Workflow 1	243	237	126	19802
Workflow 2	205	329	144	17699
Workflow 3	246	221	128	20168

Table S25 DFT-calculated static binding energies of OH* on different sites of Cu(111) in vacuum. "#" indicates unstable binding on the site, energies were then obtained by constraining the x- and y-coordinates of the O atom of OH* at the site.

	Binding Energy /eV		
	Hollow (fcc)	Bridge	Top
OH*	0.09	0.16[#]	0.55[#]

References

(1) Berendsen, H. J. C.; van der Spoel, D.; van Drunen, R. GROMACS: A Message-Passing Parallel Molecular Dynamics Implementation. *Comput. Phys. Commun.* **1995**, *91*, 43–56.

(2) Lindahl, E.; Hess, B.; van der Spoel, D. GROMACS 3.0: A Package for Molecular Simulation and Trajectory Analysis. *J. Mol. Model.* **2001**, *7*, 306–317.

(3) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J. Comput. Chem.* **2005**, *26*, 1701–1718.

(4) Hess, B.; Kutzner, C.; Van Der Spoel, D.; Lindahl, E. GRGMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435–447.

(5) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; Van Der Spoel, D.; et al. GROMACS 4.5: A High-Throughput and Highly Parallel Open Source Molecular Simulation Toolkit. *Bioinformatics* **2013**, *29*, 845–854.

(6) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindah, E. Gromacs: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1–2*, 19–25.

(7) Jorgensen, W. L.; Tirado-Rives, J. The OPLS Potential Functions for Proteins. Energy Minimizations for Crystals of Cyclic Peptides and Crambin. *J. Am. Chem. Soc.* **1988**, *110*, 1657–1666.

(8) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

(9) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A Smooth Particle Mesh Ewald Method. *J. Chem. Phys.* **1995**, *103*, 8577–8593.

(10) Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. LINCS: A Linear Constraint Solver for Molecular Simulations. *J. Comput. Chem.* **1997**, *18*, 1463–1472.

(11) Bussi, G.; Donadio, D.; Parrinello, M. Canonical Sampling through Velocity Rescaling. *J. Chem. Phys.* **2007**, *126*, 14101.

(12) Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **2016**, *14*, 1153–1173.

(13) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: An Extensible Neural Network Potential with DFT Accuracy at Force Field Computational Cost. *Chem. Sci.* **2017**, *8*, 3192–3203.

(14) Schütt, K. T.; Sauceda, H. E.; Kindermans, P. J.; Tkatchenko, A.; Müller, K. R. SchNet - A Deep Learning Architecture for Molecules and Materials. *J. Chem. Phys.* **2018**, *148*.

(15) Novikov, I. S.; Gubaev, K.; Podryabinkin, E. V.; Shapeev, A. V. The MLIP Package: Moment Tensor Potentials with MPI and Active Learning. *Mach. Learn. Sci. Technol.* **2021**, *2*, 025002.

(16) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* **2014**.

(17) Podryabinkin, E. V.; Shapeev, A. V. Active Learning of Linearly Parametrized Interatomic Potentials. *Comput. Mater. Sci.* **2017**, *140*, 171–180.

(18) Kittel, C. Introduction to Solid State Physics; Johnson, S., Ed.; John Wiley & Sons, Inc.: New York, 2004; pp 1–704.

(19) Morgan, W. S.; Christensen, J. E.; Hamilton, P. K.; Jorgensen, J. J.; Campbell, B. J.; Hart, G. L. W.; Forcade, R. W. Generalized Regular K-Point Grid Generation on the Fly. *Comput. Mater. Sci.* **2020**, *173*, 109340.

(20) Heenen, H. H.; Gauthier, J. A.; Kristoffersen, H. H.; Ludwig, T.; Chan, K. Solvation at Metal/Water Interfaces: An Ab Initio Molecular Dynamics Benchmark of Common Computational Approaches. *J. Chem. Phys.* **2020**, *152*, 144703.
(21) Natarajan, S. K.; Behler, J. Neural Network Molecular Dynamics Simulations of Solid-Liquid Interfaces: Water at Low-Index Copper Surfaces. *Phys. Chem. Chem. Phys.* **2016**, *18*, 28704–28725.