Appendix

A Additional Transfer Learning Results

This section provides additional ablation experiments for the transfer learning approach to provide further evidence for the enhanced data efficiency and good quality of uncertainty quantification of this method beyond the NbSiAs benchmark. The experiments here go significantly beyond the evidence provided in the main manuscript by demonstrating these properties under different circumstances (quality of force predictions, transfer learning between different electronic structure methods). Thus, they additionally demonstrate the robustness of the proposed method. However, for brevity, they were omitted in the main manuscript.

A.1 Empirical Evaluation

We did three additional experiments to evaluate the transfer learning approach, representing likely scenarios where transfer learning might be employed. Notably, these experiments did not involve the on-the-fly active learning workflow but were instead done as tests of the prior over the parameters of the BNN on benchmark datasets. The first test is a transfer learning scenario of finetuning a more general NequIP model trained on a variety of different compounds to a specific molecule of interest not included in the pre-training dataset. More specifically, we pre-train a NequIP model on a dataset consisting of a variety of compounds of the MD17[1] and MD22[2] datasets, which consist of MD trajectories of several molecules at DFT level accuracy, and then fine-tune it on the paracetamol dataset of the MD17 dataset.
The second benchmark is a transfer learning scenario from DFTB level accuracy to DFT level accuracy. In particular, we generate a large dataset of different configurations of a stachyose molecule in DFTB for pre-training and then utilize the stachyose data from the MD22 dataset for the transfer learning task.
The third test scenario is a transfer learning task for reaching CC level accuracy on an ethanol molecule starting from a model pre-trained on the corresponding ethanol data from the MD17 dataset. The CC-level dataset used for this was introduced by Bogojeski et al.[3].

All experiments were done with 8 Monte Carlo samples generated from the same Markov chain, which has been identified as a good tradeoff between computational complexity and quality of uncertainty quantification in our previous work[4]. Details of all the datasets can be found in Appendix C. We set σ_{TL} as 0.2 for all experiments.

A.1.0.1 The Evaluation Metrics:

On all tasks, we evaluate the model's overall accuracy in terms of the Root Mean Square Error (RMSE) of the force components in dependence on the size of the training dataset. We analyze the transfer learning models' accuracy and quality of uncertainty quantification in comparison to a model with a Gaussian mean field prior $p(\boldsymbol{\theta}) \sim N(\mathbf{0}, I)$. For the evaluation of the uncertainties, we compare the Mean Log Likelihoods (MLLs) of the force components or energies as a function of the RMSE for both models. To smooth each predicted distribution of the 8 Monte Carlo samples on this metric, we fit a normal distribution to the means and variances of each predicted distribution and use these smoothed distributions instead. Further, since the main goal of the uncertainty measure is the identification of configurations with a large error in the prediction, we evaluate the models in the task of detecting force components with a large prediction error based on the predicted uncertainty. More specifically, we analyze the corresponding AUC-ROC scores for detecting large errors via the predicted variance and plotting them as a function of the RMSE. On the ethanol and paracetamol datasets, errors of more than 1kcal/molÅ were considered large, while on the more difficult stachyose dataset, the cutoff was set as 3kcal/molÅ because an error of 1kcal/molÅ could not be considered an outlier. Since ethanol is a very small molecule, it was computationally feasible to include a deep ensemble containing 8 models trained from scratch as a second baseline (see Appendix B.6 for the training details).

A.2 Results

As can be seen in Figure 1, very high accuracies were reached for the transfer learning model on the paracetamol dataset, even for small training datasets in terms of the RMSE when compared to the model trained from scratch. Further, there is no major decrease in the quality of uncertainty quantification at a given accuracy as measured by the MLLs and AUC-ROC scores and the plots are almost on top of each other where the RMSEs overlap. However, there might be a very small decrease in quality as indicated by Figure 1.
On the stachyose dataset, again, a clear improvement in accuracy at equal amounts of training samples is visible when compared to

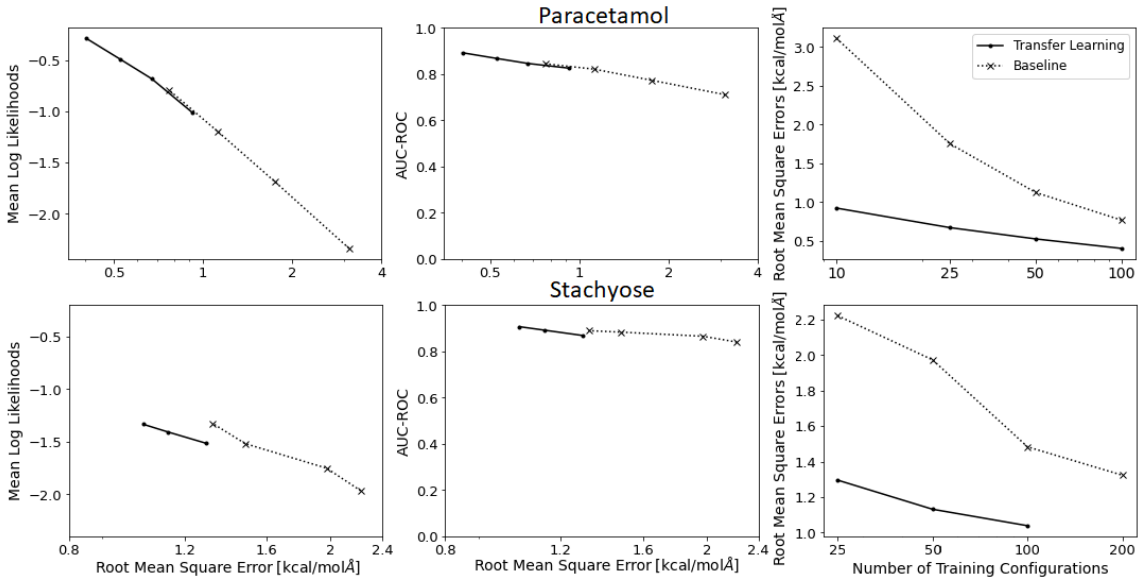

Fig. 1 Results on the paracetamol and stachyose datasets. On the left are the mean log-likelihoods as a function of RMSE (all means and standard deviations in kcal/molÅ). In the middle are the AUC-ROC scores for uncertainty-based detection of force components with a high prediction error. On the right are the Root Mean Square Errors as a function of the number of training configurations

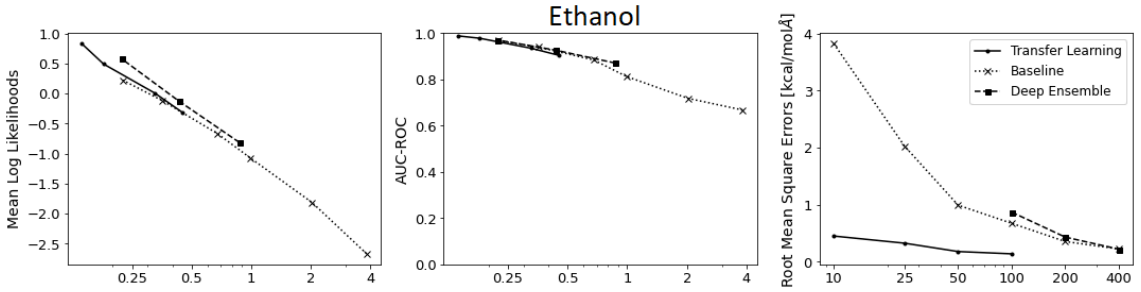

Fig. 2 Results on the ethanol dataset. On the left are the mean log-likelihoods as a function of RMSE (all means and standard deviations in kcal/molÅ). In the middle are the AUC-ROC scores for uncertainty-based detection of predictions with a large error. On the right are the Root Mean Square Errors as a function of the number of training configurations

the baseline model (Figure 1). However, both models have higher RMSEs than their counterparts on the paracetamol dataset at equal amounts of training configurations. The MLLs of the transfer learning model appear to be slightly lower than for a model trained from scratch when controlled for accuracy. The same is true only to a much smaller degree for the AUC-ROC scores. Further analysis revealed that the validation set was too small for the large configuration space of stachyose to properly recalibrate the uncertainties, which led to an overestimation of the errors on the test set for the transfer learned models but not for the baseline models. This also explains the absence of such reduced performance on the AUC-ROC scores, which are invariant under recalibration of uncertainties. Accounting for the slightly wrong calibration by recalibrating the uncertainties on the test set instead of the validation set confirmed miscalibration as the main source of the gap in MLLs. In particular, the gap between the MLLs of the transfer learning models that are closest in RMSE reduced from 0.191 to 0.086. Notably, we also attempted a transfer learning scenario on the stachyose dataset, where the model was only pre-trained on the small molecules of the MD17 dataset. However, this did not lead to any improvement in data efficiency. The most likely explanation for this is that the local atomic environments on the small molecule datasets, which the neural network uses to calculate potential energy contributions, are qualitatively very different from those of the stachyose molecule.
The biggest improvement in accuracy, when compared to the baseline model, was found on the ethanol dataset (Figure 2), with an RMSE of less than 0.5kcal/molÅ with only 10 configurations. There appears to be no decrease in the quality of uncertainty quantification, both in terms of MLLs as well as AUC-ROCs on this benchmark, when compared to the baseline model. Comparing both the baseline model and the transfer learning model to the deep ensemble, almost identical performance can be seen on the outlier detection task, while the deep ensemble has slightly higher MLLs.

Additional analysis was performed by breaking down the MLLs into contributions from force components, whose prediction error falls into a certain interval, e.g. the contribution to the MLLs from samples where the prediction error is in the interval $[0.1, 0.2)$ is given by the sum over the log-likelihoods of all force components in the test set where the prediction error is in the interval $[0.1, 0.2)$ divided by the total number of force components in the test set. The results can shown in Figure 3 demonstrate that the total MLL score is dominated by samples whose prediction error is small. This offers an explanation for the slightly better MLLs without a similar performance gap on the outlier detection task on this benchmark: the deep ensembles achieve slightly better uncertainty quantification on configurations with a small prediction error but not on configurations with a large prediction error.

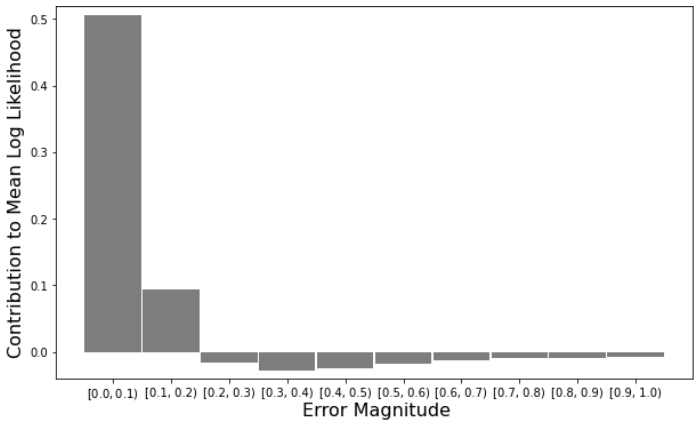

Fig. 3 Decomposition of the MLLs of the force components into contributions of different error magnitudes on the ethanol test set for the deep ensemble trained on 400 training configurations. The force units are in kcal/molÅ.

Notably, for all force transfer learning scenarios, the error of the pre-trained model was quite large with mean absolute errors of 2.31kcal/molÅ on the paracetamol validation set, 4.34kcal/molÅ on the stachyose validation set and 5.12kcal/molÅ on the ethanol validation set. Further, all pre-trained models achieved a validation loss smaller than 0.15kcal/molÅ on their pre-training datasets, strongly indicating that DFTB, DFT and CC methods disagree quite substantially in their force predictions for a given configuration. However, as was already alluded to in the introduction, this can potentially be traced back to simple biases in the simulation methods, such as slightly different equilibrium bond lengths. These small biases in different simulation methods can lead to qualitatively very similar force fields that may disagree substantially on the forces of a given configuration. This would also explain why transfer learning is very efficient in these cases, as the model mostly has to correct for those biases, such as equilibrium bond lengths. Importantly, those force fields will lead to similar predictions of physical and chemical properties despite their apparently large disagreement, while a machine-learned force field with a similar magnitude of error to one of those methods can not, in general, be expected to yield those properties as well and hence needs to be trained to a much higher accuracy.
One additional result that stands out is the relatively high RMSE of both the transfer learning and the baseline model on the stachyose dataset when compared to the other two test scenarios. However, two factors make this dataset particularly challenging. First of all, stachyose is a larger molecule than paracetamol and ethanol, which, in addition, contains many single sigma bonds that allow for

rotational degrees of freedom along the bond axis. This results in a very large configuration space for stachyose molecules, even relative to their size. The second factor that makes this benchmark more challenging for the transfer learning model is that, unlike in the ethanol case, the higher accuracy dataset was not composed of configurations generated from an MD trajectory of the lower accuracy method but instead from a trajectory at DFT-level accuracy. As a result, the distribution of configurations in the DFT dataset will be different from the one from the DFTB dataset.
Lastly, one important observation we made is that the transfer learning approach converges much faster than when training from scratch. While state-of-the-art models can take days to train from scratch, training and validation losses converged within minutes on the transfer learning tasks. The only reason we let the sampling algorithm run for as long as described in Appendix B is to make sure that no pathological overfitting takes place.

B Methodological Details

This section contains the methodological details that were omitted from the main manuscript to avoid breaking the flow of reading.

B.1 Details of the Base Models

The foundation models in this work operate by first mapping the input $x = \{(\boldsymbol{r}_1, z_1), ..., (\boldsymbol{r}_n, z_n)\}$ and optionally the lattice vectors $\boldsymbol{L_1}, \boldsymbol{L_2}, \boldsymbol{L_3}$ to latent variables $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$ that are invariant under distance-preserving transformations of the atomic coordinates $\boldsymbol{r}_i$. From those invariant atomic features, atomic energy contributions $\hat{E}_1, ..., \hat{E}_n$ are then calculated and summed up into a total potential energy prediction $\hat{E} = \sum_i E_i$. NequIP and MACE then calculate the forces acting on the atoms as the negative gradients of the potential energy $\hat{\boldsymbol{F}}_i = -\nabla_{\boldsymbol{r}_i} \hat{E}$ via automatic differentiation libraries, while the Equiformerv2 model calculates the forces directly from a set of equivariant atomic feature vectors. To apply the Bayesian neural network framework to these models, the architectures were modified slightly by adding layers that compute standard deviations $\sigma_1, ..., \sigma_n$ for the forces from the invariant features $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_n\}$. Further, an energy standard deviation $\sigma_{\hat{E}}$ is introduced and the distribution over the energy and forces is modeled as

$$E, \boldsymbol{F}_1, ..., \boldsymbol{F}_n | (\boldsymbol{r}_1, z_1), ..., (\boldsymbol{r}_n, z_n), \boldsymbol{\theta} \sim N(\hat{E}, \sigma_{\hat{E}}^2) \Pi_{i=1}^n N(\hat{\boldsymbol{F}}_i, \sigma_i^2 I),$$

where N denotes a normal distribution and I is the identity matrix.
For the Equiformev2 and MACE models, the predictions are additionally conditioned on the lattice vectors $\boldsymbol{L_1}, \boldsymbol{L_2}, \boldsymbol{L_3}$. For the MACE model, the stress tensor is also predicted. For this, the predicted distribution is modified to

$$E, \boldsymbol{F}_1, ..., \boldsymbol{F}_n, \boldsymbol{S} | (\boldsymbol{r}_1, z_1), ..., (\boldsymbol{r}_n, z_n), \boldsymbol{L_1}, \boldsymbol{L_2}, \boldsymbol{L_3}, \boldsymbol{\theta} \sim N(\hat{E}, \sigma_{\hat{E}}^2) N(\hat{\boldsymbol{S}}, \sigma_{\hat{\boldsymbol{S}}}^2 I) \Pi_{i=1}^n N(\hat{\boldsymbol{F}}_i, \sigma_i^2 I),$$

where $\hat{\boldsymbol{S}}$ is a vector containing the predicted components of the stress tensor and $\sigma_{\hat{\boldsymbol{S}}}$ is a small fixed standard deviation that is set to $0.1/16$ kcal/molÅ^3. For the prior of the additional parameters from the added layers, the means were set to zero but the same standard deviation as for the other parameters was used.

B.2 The EquiformerV2-Based Neural Network Architecture

The overall architecture of the neural network derived from the EquiformerV2 model is summarized in Figure 4. The publicly available 31 million parameter EquiformerV2 model pre-trained on the entire OC20 dataset, including the MD data, was chosen as a base model for transfer learning. Because this model is too large to train from scratch on such a relatively small training dataset, a smaller EquiformerV2 model was chosen for the baseline model. The configuration for that smaller model can be found on the first author's GitHub page. Because the projection layers were not included in the pre-training, their means were set to zero in the transfer learning prior. Both energies and force labels were used for training.

B.3 The NequIP-Based Neural Network Architecture

For the base neural network architecture, a NequIP model with four interaction blocks, a latent dimension of 64 and even and odd parity features up to and including angular momentum number l=2 was used.
The standard deviations of the forces σ_i are predicted by a three-layer MLP with input dimension 64, latent dimensions 32 and 16 and output dimension 1.
SiLU activation functions are used for the latent layers and the output activation function is the exponential function.
A base with 5 interaction blocks was used.
Because there aren't a lot of labeled examples for the potential energies, a fixed energy standard deviation was used during the on-the-fly experiments, which was set to 10 percent of the target accuracy.
For the experiments in Appendix A, no energies were included in the training and train only the force labels were trained on.

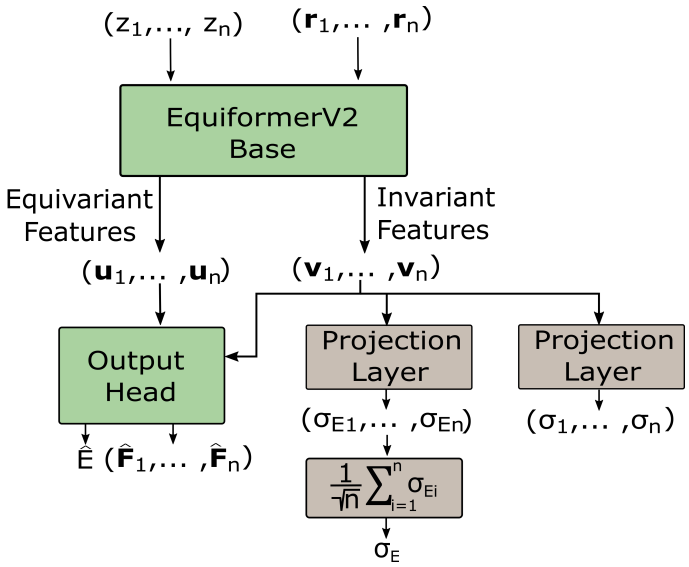

Fig. 4 The neural network architecture derived from the EquiformerV2 model used in the transfer learning task of potential energies. The green modules correspond to the modules of the EquiformerV2 architecture and the grey ones are additions to model the data probabilistically. The projection layers are linear layers followed by an exponential activation function.

B.4 The MACE-Based Neural Network Architecture

The modifications to the MACE model follow the same pattern as for the NequIP model.
The output of the mace-mp medium model contains feature vectors for each atom. 256 components of each feature vector are rotation invariant. From these features, the standard deviations of the forces σ_i are predicted by a three-layer MLP with input dimension 256, latent dimensions 64 and 16 and output dimension 1.
Again use a fixed energy standard deviation set to 10 percent of the target accuracy was used. For all experiments, a fixed stress standard deviation of $0.1/16$ kcal/molÅ^3 was used.

B.5 Generating Samples from the Posterior

For sampling the Bayesian posterior, we use the SGHMC algorithm[5] with the adaptive mass term introduced by us in a previous work[4].

B.5.0.1 Sampling for the Transfer Learning Experiments:

For the ethanol and paracetamol test cases, the step size γ is exponentially decreased from 10^{-2} and $0.3 \cdot 10^{-2}$ to 10^{-5} during the first 10^6 steps for the baseline model and transfer learning model, respectively. At the end of this phase, the first model is sampled. Afterward, a cyclical learning rate schedule is used:

$$\gamma_i = \frac{\gamma_0}{2}\left(\cos\left(\pi + \frac{i \cdot \pi}{K}\right) + 1\right)$$

with $\gamma_0 = 0.001$ and cycle length $K = 50000$
to sample the subsequent models from the same Markov chain at the end of each cycle.
The same procedure is also utilized for the baseline model on the stachyose test case. However, the initial convergence phase is shortened to $0.5 \cdot 10^6$ steps for the transfer learning model, as the other two test cases had revealed a quicker convergence for the transfer learning models. For the paracetamol and ethanol cases, a batch size of 30 is used and for the stachyose case, it is set as 15. Analogous to the on-the-fly learning experiments, an energy offset was calculated for the training, validation and test data, of the equiformer model so that for the first sample in the training data, the energy equals the one predicted by the pre-trained model. For the surface-adsorbate transfer learning task, a batch size of 20 was used for the baseline model and for the transfer learning task, it is set as 15. Furthermore, the same sampling procedure is employed. For the baseline and transfer learning model, the step size γ is exponentially decreased from 10^{-4} and $\cdot 10^{-5}$ to 10^{-7} and 10^{-8} respectively during the initial convergence phase. This phase was $0.5 \cdot 10^6$ steps long for the baseline model and 10^5 steps long for the transfer learning model, respectively. Then again, a cyclical sampling procedure is employed to generate the other samples with $\gamma_0 = 0.0001$ and cycle length $K = 50000$ After the first 90 percent of the initial convergence phase, the mass term is kept constant to ensure close convergence to the posterior.

B.5.0.2 Sampling for the on-the-fly Finetuning Experiments:

To sample the posterior for the on-the-fly fine-tuning experiments, 8 separate Markov Chains were run, one for each Monte Carlo sample. After each added training sample, the SGHMC algorithm with the adaptive mass term is run for 2000 steps to update the Monte

Carlo samples. A batch size of 5 and a learning rate schedule

$$\gamma_i = \frac{\gamma_0}{2}\left(\cos\left(\frac{i \cdot \pi}{2000}\right) + 1\right),$$

was used. For the MACE model, $\gamma_0 = 0.001$ was used and for the NequIP model, it was set to $\gamma_0 = 0.00003$. To improve the speed of convergence, priority sampling was used to increase the likelihood of sampling the newly added training sample at each iteration. The sampling probability was increased so that, on average, the newly added sample is contained once in each minibatch. Each sample in the minibatch estimator of the log-likelihood was weighted by the likelihood ratio of a uniform sampling procedure and the sampling probabilities used to ensure the minibatch estimator is unbiased.

B.6 Details on the Simulations

B.6.0.1 The Ethanol On-The-Fly Simulation: A Langevin thermostat with 0.5 fs time steps and a friction term of $0.01 fs^{-1}$ of the Atomic Simulation Environment (ASE) library was used to drive the dynamics. The DFT calculations were done with the Vienna Ab initio Simulation Package (VASP) using a cubic 30 Å simulation cell. A plane wave energy cutoff of 800 eV with a convergence criterion of $1e-6$ eV and the B3LYP exchange-correlation functional[6–9] was used.

B.6.0.2 The $LaMnO_3$ On-The-Fly Simulation: The NPT thermostat of the Atomic Simulation Environment (ASE) library with a 0.5 fs time step, a ttime of 100 fs, a pfactor of $160GPa \cdot 0.1 \cdot 75^2 fs^2$ and an external pressure of 1 bar was used to drive the dynamics. A 2x2x2 supercell containing 40 atoms in total was simulated. The DFT calculations were done with the Vienna Ab initio Simulation Package (VASP). A plane wave energy cutoff of 500 eV with a convergence criterion of $1e-4$ eV and the PBE exchange-correlation functional[10] and a 4x4x4 gamma centered Monkhorst-pack grid was used. A Hubbard term of 3.9 was used for the Mn atoms.

B.6.0.3 The Proton Diffusion On-The-Fly Simulations: The Langevin thermostat with a friction term of 0.5 of the Atomic Simulation Environment (ASE) library was used to drive the dynamics for 1 ps to set the temperature. Afterwards, The velocity Verlet integrator was used to continue the simulation for another 30 ps. One initial training run was done for each of the two fine-tuned models. Afterwards, 5 production runs were done to investigate the diffusivity of the protons. 1.5 fs time steps were used for all simulations. A $CaZrS_3$ supercell containing 160 atoms and two additional protons was simulated. The DFT calculations were done with the Vienna Ab initio Simulation Package (VASP). A plane wave energy cutoff of 510 eV with a convergence criterion of $1e-4$ eV and the PBE exchange-correlation functional[10] and a 2x2x2 gamma centered Monkhorst-pack grid was used.

B.7 Pretraining the Models

B.7.0.1 Pretraining for the Transfer Learning Experiments:

To pre-train a model, it was converged to a local maximum of the log-posterior on the pre-training dataset with a Gaussian mean field prior $p(\boldsymbol{\theta}) \sim N(\mathbf{0}, I)$. Almost the same sampling algorithm and hyperparameters were used as in the sampling of the posterior of the corresponding baseline model. The only differences are that the injected noise is downscaled by a factor of 0.1 and only the first model is sampled. The injected noise was not set to zero, because we found that a small amount of injected noise actually speeds up convergence, especially at the beginning of the optimization.

B.7.0.2 Pretraining the NequIP Model on the Spice Dataset for the Ethanol On-The-Fly experiment:

The NequIP model was trained at a batch size of 25 at a fixed learning rate of 3e-5 with the Adam optimizer. The loss function $L = (1/200)MSE(Energy) + MSE(Forces)$ was used, where the Mean Square Error (MSE) refers to the batch mean. We keep a validation set of 70 structures from the SPICE dataset, which are not included in the training dataset. Every 20000 training steps, the energy MSE on the validation set were evaluated. The training was stopped after the energy validation loss hadn't improved for 3 epochs. The final model used is the one with the lowest validation loss during the training run.

B.8 Training and Evaluating the Deep Ensemble

To generate the deep ensemble, 8 stochastic NequIP models were trained from scratch with different random initializations of the neural network parameters. The models are trained with the AMSGrad optimizer at a batch size of 30 with an initial learning rate of 0.01, which is decayed to 10^{-5} over the course of $5 \cdot 10^5$ training steps. Every 1000 training steps, the model's RMSE is evaluated on a validation set of size 10. The parameter set with the best RMSE during the optimization procedure for each weight initialization is used to make predictions on the test set. We again fit a normal distribution to the predictions of the ensemble and recalibrate those uncertainties on the validation set when evaluating the MLLs.

B.9 Calculation of Densities During Inference

To smooth the predicted distribution of several Monte Carlo samples or ensemble models, the final distribution was smoothed by fitting a normal distribution to the predicted means and variances. The total variance of several Monte Carlo samples or ensemble models for

force components was calculated as

$$\sigma_{F_i}^2 = Variance\left(\hat{F}_{i,j}\right) + \frac{1}{k}\sum_{j=1}^{k}\sigma_{F_{i,j}}^2,$$

where j enumerates the predicted standard deviations of the individual Monte Carlo samples/ ensemble models, $\hat{F}_{i,j}$ is the predicted expectation value for the i-th force component of that particular model and $\sigma_{F_{i,j}}$ the corresponding standard deviation. The variance is calculated over the Monte Carlo samples/ ensemble models. The mean of the predicted distribution was simply calculated as $\hat{F}_i = \frac{1}{k}\sum_{j=1}^{k}\hat{F}_{i,j}$.
The calculation of the final energy and stress distribution was done completely analogously.

B.10 Construction of the Estimator for the Relationship between Error and Predicted Standard Deviation from Figure ??.

To construct this estimator, the pairs of predicted standard deviations and observed errors $\{(\sigma_1, e_1), ..., (\sigma_{1800}, e_{1800})\}$ were ordered by the magnitude of the predicted standard deviation. A Gaussian filter with a sigma value of 200 was then applied to both the list of ordered variances $[\sigma^2_{sorted,1}, ..., \sigma^2_{sorted,1800}]$ and squared errors $[e^2_{sorted,1}, ..., e^2_{sorted,1800}]$ resulting in the smoothed arrays $[\sigma^2_{smoothed,1}, ..., \sigma^2_{smoothed,1800}]$ and $[e^2_{smoothed,1}, ..., e^2_{smoothed,1800}]$. Finally, the root of the smoothed predicted variances over the root of the smoothed squared errors was plotted to generate the figure.

B.11 The Bayesian Calibration Estimator

Given a dataset of empirical observations of (independent) errors $E = \{e_1, ..., e_n\}$ and predicted uncertainties $\Sigma = \{\sigma_1, ..., \sigma_n\}$, the error e^* on a new sample with predicted standard deviation σ^* is given in closed form as the students t-distribution

$$p(e^*|\sigma^*, E, \Sigma) = \int p(e^*|\sigma^*, \lambda)p(\lambda|E, \Sigma)d\lambda$$

$$= \frac{1}{\int p(E, \Sigma|\lambda)p(\lambda)d\lambda} \cdot \int p(e^*|\sigma^*, \lambda)p(E, \Sigma|\lambda)p(\lambda)d\lambda$$

$$= \frac{\Gamma(a + \frac{n+1}{2})}{\sqrt{2\pi\sigma^{*2}}\Gamma(a + \frac{n}{2})} \cdot \frac{(b + \frac{1}{2}n \cdot M_n)^{a+\frac{n}{2}}}{(b + \frac{1}{2}n \cdot M_n + \frac{1}{2}\frac{e^{*2}}{\sigma^{*2}})^{a+\frac{n+1}{2}}},$$

where $M_n = \frac{1}{n}\sum_{i=1}^{n}\frac{e_i^2}{\sigma_i^2}$. By integrating this density from $e^* = -K$ to $e^* = K$, the result:

$$p(|e^*| < K|\sigma^*, E, \Sigma) = \frac{2K\Gamma(a + \frac{n+1}{2})}{\sqrt{2\pi\sigma^{*2}}\Gamma(a + \frac{n}{2})\sqrt{b + \frac{1}{2}n \cdot M_n}}$$

$$\times Hyp2F1\left(\frac{1}{2}, a + \frac{n+1}{2}; \frac{3}{2}, -\frac{K^2}{\sigma^{*2}(2b + n \cdot M_n)}\right),$$

can be derived. Further, with the identities $\frac{\Gamma(x+\frac{1}{2})}{\Gamma(x)\sqrt{x}} \to 1$ and $(1 + \frac{c}{x})^x \to e^c$ for $x \to \infty$ it can be verified, that for $n \to \infty$ the predicted error distribution becomes

$$p(e^*|\sigma^*, E, \Sigma) \sim \mathcal{N}\left(0, \sigma^{*2}M_n\right).$$

C The Datasets

C.1 The Ethanol Transfer Learning Datasets

To pre-train the model, 5000 randomly sampled configurations from the MD17 ethanol dataset are used. This dataset consists of over 500000 configurations generated from a molecular dynamics trajectory calculated at DFT level accuracy. The training and test datasets of ethanol at CCSD(T) level accuracy introduced by Bogojeski et al. [3] were used for the transfer learning task. The last 10 configurations of the training set were used as validation data. The actual training data consisted of the first $m \in \mathbb{N}$ configurations of the training dataset for varying values of m.

C.2 The Paracetamol Transfer Learning Datasets

The pretraining dataset consists of randomly sampled configurations from the aspirin, benzene, malonaldehyde, toluene, salicylic acid, naphthalene, ethanol, uracil and azobenzene from the MD17 dataset, as well as the AT-AT DNA base pair, stachyose, Ac-Ala3-NHMe, and docosahexaenoic acid datasets from the MD22 dataset. The first 100000 configurations from each MD17 dataset and all configurations from the MD22 datasets were used to form a pool of configurations from which 100000 are randomly drawn as the pretraining dataset. For the actual training set $m \in \mathbb{N}$, configurations are randomly sampled from the MD17 paracetamol dataset for varying values of m. 10 additional configurations are randomly sampled as a validation set. The rest of the 106490 configurations are used as a test set.

C.3 The Stachyose Transfer Learning Datasets

The pretraining dataset was generated from a long molecular dynamics trajectory of a stachyose molecule in DFTB+ [11]. The initial geometry was generated from a structural relaxation with a convergence criterion of $10^{-3} H/\text{Å}$ for the maximal force component. The MD trajectory was simulated at 1 femtosecond time steps with a Nose Hoover thermostat [12] at 600 Kelvin with a coupling strength of 3200 cm^{-1}. The simulation ran for 10^6 time steps using the velocity Verlet driver with one configuration sampled every ten time steps, yielding a dataset of 100000 configurations. For both the geometry optimization as well as the MD simulation, a Hamiltonian with self-consistent charges [13] and third-order corrections [14] was used in correspondence with the 3ob-3-1 Slater Coster files [15]. For all atoms, s- and p-orbitals were used in the Hamiltonian.

For the actual training set $m \in \mathbb{N}$, configurations are randomly sampled from the first 10000 configurations of the MD22 stachyose dataset for varying values of m. 10 additional configurations are randomly sampled from the configurations 10100 to 10900 as a validation set. Configurations 11000 up to 27000 are used as a test set.

C.4 The Surface-Adsorbate Dataset

For the energy transfer learning dataset of the surface adsorbate system, the NbSiAs surface - COH adsorbate dataset from the OC20-Dense dataset was chosen.
8000 configurations were randomly sampled as possible training configurations. For a training dataset of size $n \in \mathbb{N}$, the first n configurations from that subset were used as a training set. Further, twenty randomly sampled configurations were used as a validation set to recalibrate the uncertainties. The rest of the configurations were used as the test set.

D Runtime Estimates on Different Hardware Configurations for the $CaZrS_3$ - Proton System

A single 30-ps proton diffusion simulation in $CaZrS_3$ would take around 3 months on the two 36-core Intel Platinum 8360Y processors that were used to do the DFT interventions in VASP 5.4.4, while it took less than a day to finetune the 15 kcal/mol threshold model and around a week for the 5 kcal/mol threshold model during the initial 30-ps training runs. The time for the 5 production runs with each model was negligible. Increasing the CPU resources to eight processors will reduce the time of the simulation in VASP 5.4.4 to around one month and also reduce the time for the on-the-fly simulations to less than half of the previous values, since the DFT interventions were the computational bottleneck. Increasing CPU resources even more will start to result in diminishing returns, as the MPI communications overhead will start to become the limiting factor.
GPU nodes containing 8 L40S GPUs were used for the experiments and updating all Monte Carlo samples on a new training data point takes only a few minutes on that hardware. To test the impact of constrained GPU resources, the experiment for the largest system, the 162-atom $CaZrS_3$-proton system, was rerun using only two A100 GPUs. On that hardware, it takes around 20 minutes to update the model. However, it should be noted that DFT calculations for systems of that size will also be quite time-intensive and on the two Intel Platinum 8360Y processors that were used, they took around 24-25 minutes.

DFT calculations involving two 36-core Intel Platinum 8360Y processors were conducted on a single compute node with two CPU sockets (four NUMA domains, 36 logical cores and 64 GB RAM per domain). For calculations with eight 36-core Intel Platinum 8360Y processors, VASP was run on four compute nodes. VASP was run using the full nodes, with MPI ranks and OpenMP threads distributed across CPUs. No explicit NUMA binding or memory placement was applied. Memory allocation followed the system's default NUMA policy. VASP was compiled with Intel compilers (icx, icpx, ifx) and Intel MPI (mpiicx, mpiicpx, mpiifx). Linear algebra routines used the Intel MKL library. The VASP binary used was vasp_std (CPU variant). No custom compilation options beyond the default module build were applied.

Neural network optimization was performed with torch 2.6 and mace-torch 0.3.0 using CUDA 12.4.1 without cuEquivariance acceleration.

Notes and references

1 Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017. doi: 10.1126/sciadv.1603015.

2 Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T. Unke, Adil Kabylda, Huziel E. Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023. doi: 10.1126/sciadv.adf0873. URL `https://www.science.org/doi/abs/10.1126/sciadv.adf0873`.

3 Mihail Bogojeski, Leslie Vogt-Maranto, Mark E. Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature Communications*, 11, 2019.

4 Tim Rensmeyer, Ben Craig, Denis Kramer, and Oliver Niggemann. High accuracy uncertainty-aware interatomic force modeling with equivariant bayesian neural networks. *Digital Discovery*, 3:2356–2366, 2024. doi: 10.1039/D4DD00183D. URL `http://dx.doi.org/10.1039/D4DD00183D`.

5 Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

6 Axel D. Becke. Density-functional thermochemistry. iii. the role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, 04 1993. ISSN 0021-9606. doi: 10.1063/1.464913. URL https://doi.org/10.1063/1.464913.

7 Chengteh Lee, Weitao Yang, and Robert G. Parr. Development of the colle-salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B*, 37:785–789, Jan 1988. doi: 10.1103/PhysRevB.37.785. URL https://link.aps.org/doi/10.1103/PhysRevB.37.785.

8 P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch. Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *The Journal of Physical Chemistry*, 98(45):11623–11627, 1994. doi: 10.1021/j100096a001. URL https://doi.org/10.1021/j100096a001.

9 S. H. Vosko, L. Wilk, and M. Nusair. Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis. *Canadian Journal of Physics*, 58(8):1200–1211, August 1980. doi: 10.1139/p80-159.

10 John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996. doi: 10.1103/PhysRevLett.77.3865. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.3865.

11 B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshaye, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim. DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. *The Journal of Chemical Physics*, 152(12):124101, 2020. ISSN 0021-9606. doi: 10.1063/1.5143190. URL https://doi.org/10.1063/1.5143190.

12 Glenn J. Martyna, Mark E. Tuckerman, Douglas J. Tobias, and Michael L. Klein. Explicit reversible integrators for extended systems dynamics. *Molecular Physics*, 87(5):1117–1157, 1996. doi: 10.1080/00268979600100761. URL https://doi.org/10.1080/00268979600100761.

13 M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haugk, Th. Frauenheim, S. Suhai, and G. Seifert. Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties. *Phys. Rev. B*, 58:7260–7268, 1998. doi: 10.1103/PhysRevB.58.7260. URL https://link.aps.org/doi/10.1103/PhysRevB.58.7260.

14 Michael Gaus, Qiang Cui, and Marcus Elstner. Dftb3: Extension of the self-consistent-charge density-functional tight-binding method (scc-dftb). *Journal of Chemical Theory and Computation*, 7(4):931–948, 2011. doi: 10.1021/ct100684s. URL https://doi.org/10.1021/ct100684s.

15 Michael Gaus, Xiya Lu, Marcus Elstner, and Qiang Cui. Parameterization of dftb3/3ob for sulfur and phosphorus for chemical and biological applications. *Journal of Chemical Theory and Computation*, 10(4):1518–1537, 2014. doi: 10.1021/ct401002w. URL https://doi.org/10.1021/ct401002w. PMID: 24803865.