JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

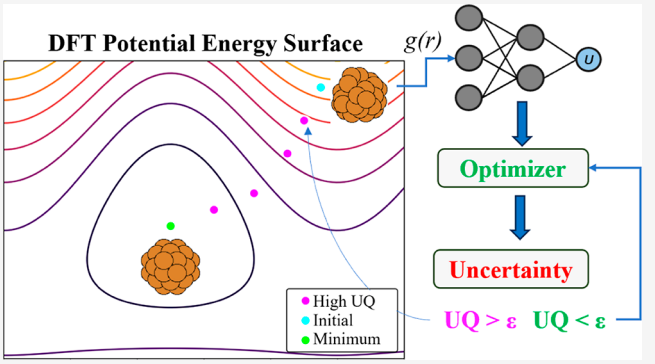
Uncertainty Based Machine Learning-DFT Hybrid Framework for Accelerating Geometry Optimization

Akksay Singh, Jiaqi Wang, Graeme Henkelman,* and Lei Li*

Cite This: J. Chem. Theory Comput. 2024, 20, 10022−10033 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Geometry optimization is an important tool used for computational simulations in the fields of chemistry, physics, and material science. Developing more efficient and reliable algorithms to reduce the number of force evaluations would lead to accelerated computational modeling and materials discovery. Here, we present a delta method-based neural network-density functional theory (DFT) hybrid optimizer to improve the computational efficiency of geometry optimization. Compared to previous active learning approaches, our algorithm adds two key features: a modified delta method incorporating force information to enhance efficiency in uncertainty estimation, and a quasi-Newton approach based upon a Hessian matrix calculated from the neural network; the later improving stability of optimization near critical points. We benchmarked our optimizer against commonly used optimization algorithms using systems including bulk metal, metal surface, metal hydride, and an oxide cluster. The results demonstrate that our optimizer effectively reduces the number of DFT force calls by 2−3 times in all test systems.



1. INTRODUCTION

Geometry optimization is a common task in computational chemical modeling. Optimization typically aims to find stable atomic configurations using algorithms including conjugate gradient descent or quasi-Newton approaches. From binding energy calculations to modeling reaction pathways, many physical properties can be determined from critical points on potential energy landscapes, describing atomic structures of interest. The bottleneck in these optimizations is the iterative need for quantum chemical calculations of energies and forces. Therefore, it is important to develop algorithms that increase optimization efficiency and reduce the number of force calls required in the optimization process.

Machine learning potentials (MLPs) are playing an increasingly prominent role in advancing computational physical sciences. A plethora of MLP approaches have been developed, including artificial feed forward neural networks (ANN),[1−13] Gaussian approximation potentials (GAPs),[14,15] Gaussian process regression (GPR),[16−20] graph neural networks (GNNs),[21−24] linear models[25−30] using spectral neighbor analysis (SNAP) or moment tensor potentials (MTP), among others. These MLPs have demonstrated accuracy comparable to ab initio approaches with reduced computational cost.

Utilizing MLPs as surrogate models of the potential energy surface (PES) to guide the optimization procedure has become an attractive approach to accelerate computational tasks such as transition state searches, and geometry optimization.[16,19,20,23,24,31−34] Peterson employed an ANN potential to accelerate saddle point searches with the nudged elastic band method.[34] Koistinen et al. used GPR to reduce the number of energy and force evaluation during transition state searches.[19,20] Ulissi et al. developed an active learning approach that leverages prior information from pretrained GNN models to accelerate structural relaxation.[23,24] These algorithms typically use an MLP model trained on-the-fly and switch to density functional theory (DFT) calculations when the uncertainty of the MLP model exceeds a threshold value. Following each switch, the new data point from the DFT calculation is then included in the training data to update the MLP model. Thus, a key aspect of these algorithms is uncertainty quantification, which ensures that model predictions are accurate and not unphysically extrapolating.

Currently, a variety of uncertainty quantification methods have been proposed, including but not limited to ensemble methods,[3,31,35,36] Bayesian neural networks (BNNs),[37,38] the delta method,[39−41] among others.[42] The ensemble method measures uncertainty as the standard deviation of the

Received: July 22, 2024
Revised: October 26, 2024
Accepted: October 29, 2024
Published: November 12, 2024

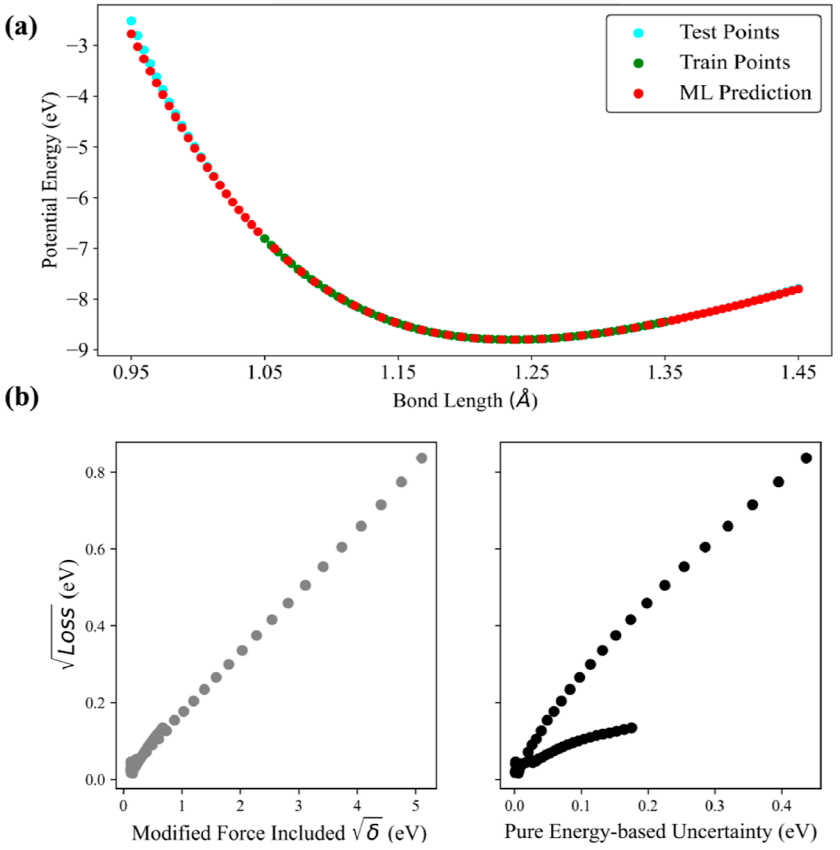


Figure 1. (a) Potential energy variation with the O−O bond length for the molecular O_2. Blue and red points represent training and testing data, respectively; (b) uncertainty estimates from our modified delta method (left) and the pure energy-based delta method (right). Square root values for the uncertainty from our method and loss are shown to ensure they have the same unit as the uncertainty obtained from the pure energy-based method. Note that the loss is positive so there are two branches corresponding to compression and stretching of the molecule. The training data set, testing data set, ML model parameters, and raw output files are given in the Supporting Information.

predictions from multiple simultaneously trained ML models. This method has been effective in identifying extrapolation regions when trained with large data sets. However, individual ensemble models often need to be large, for example, Peterson used two hidden layers of 50 neurons each in their neural network model. Additionally, a relatively large ensemble size ($\geq$10) is needed to ensure reasonable accuracy.[31,35] In practice, BNNs train parameters that minimize the mean squared errors between the model predictions and target variable, but also penalize large deviations of the model parameters from some prior assumed distribution. This training procedure allows for estimation of variance of the optimized model parameters, which then can be propagated to model predictions.

More recently, Kitchin and Zhan proposed a delta method for uncertainty quantification.[43] The delta method aligns model parameters obtained by minimizing the mean squared errors (MSE) with those obtained by maximizing the log likelihood of errors being Gaussian distributed around a mean of zero. In this approach, model uncertainty is derived from the gradient and Hessian matrix of the model. A significant advantage of the delta method is that it requires only one model, eliminating the need of additional models as in the ensemble method. However, this method has not yet been incorporated into active learning approaches. Integrating the delta method into ML-based optimization algorithms is a natural step forward.

In Kitchin's delta method, only energy information is included for uncertainty estimation. In practice, MLPs typically fit both energies and forces. Identifying extrapolation with force information can improve the uncertainty quantification. Another issue related to ML-based optimization algorithms is that the MLP can struggle to achieve sufficient accuracy in forces to reach standard convergence criteria near critical points (i.e., saddle points or local minima), defined here as a maximum atomic force of <0.05 eV/Å. However, optimizing precisely in low-force regions (also described as harmonic regions) is also crucial and often requires a considerable number of force evaluations. Further lowering of the convergence criterion also leads to convergence problems due to limited resolution of the ML model and errors in the predicted force.

In this work, we propose an uncertainty-based ML-DFT hybrid framework to accelerate geometry optimization. Following Kitchin's method, we employed a modified delta method incorporating both energy and force information to enhance uncertainty estimation. Our results on benchmark systems demonstrate that our algorithm exhibits significant improvement compared to commonly used geometry optimization algorithms. Furthermore, we use the Hessian matrix calculated from MLPs as a preconditioner in a quasi-Newton method to solve the convergence problem in harmonic (low force) regions near critical points. This strategy enables stable convergence of atomic systems to atomic forces below 0.01 eV/Å.

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

2. METHODS

2.1. Modified Delta Method as Uncertainty Measure in Active Learning. Inspired by Kitchin's work, we propose a modified delta method including both energy and force information for uncertainty evaluation. Based on the delta method, the uncertainty ($\delta(G,\hat{\theta})$) of a ML model with optimized parameters ($\hat{\theta}$) is calculated using the following equation, where G represents fingerprints of a training image.

$$\delta(G, \hat{\theta}) \approx \sqrt{\sigma \nabla_{\hat{\theta}} \mathrm{MLT}(G, \hat{\theta}) H^{+} \nabla_{\hat{\theta}} \mathrm{MLT}(G, \hat{\theta})} \tag{1}$$

Two modifications are made to Kitchin's approach to include force information. First, the Moore-Penrose pseudoinverse of the Hessian (H^+) of the true loss function (eq 2) is applied, rather than only the energy term.

$$\mathrm{Loss} = \sum_{k=1}^{M} \left[\alpha \left(\frac{E_k^{\mathrm{true}} - E_k^{\mathrm{ML}}}{N_k} \right)^2 + \frac{\beta}{3N_k} \sum_{l=1}^{3} \sum_{i=1}^{N_k} (F_{il}^{\mathrm{true}} - F_{il}^{\mathrm{ML}})^2 \right] \tag{2}$$

Here E_k^{ML} and E_k^{true} are the ML and true energy of image k, F_{il}^{ML} and F_{il}^{true} are the ML and true force of atom i along direction l, with l representing the x, y, or z direction in the Cartesian coordinate system, M is the number of training images, N_k is the number of atoms in image k, and α and β are energy and force coefficients, respectively. Note, in practice the eigenvalues of the loss Hessian may need to be shifted due to optimization issues; details are included in the Supporting Information.

Second, we redefine the machine learning term ($\mathrm{MLT}(G,\hat{\theta})$) of an image whose uncertainty is estimated as the sum of an energy term and a force term (eq 3). The derivative of these terms with respect to the model parameters $\hat{\theta}$, ($\nabla_{\hat{\theta}}\mathrm{MLT}(G, \hat{\theta})$) is substituted into eq 1 to obtain uncertainty of the prediction. Note that the uncertainty we calculated here has units of eV^2, the same as the loss function defined in eq 2. This is different from the uncertainty (in unit of eV) obtained using Kichin's method, as shown in ref 43.

$$\mathrm{MLT}(G, \hat{\theta}) = \alpha \left(\frac{E_k^{\mathrm{ML}}}{N_k} \right)^2 + \beta \cdot \left(\sum_{i=1}^{3} \sum_{i=1}^{N_k} \frac{(F_{il}^{\mathrm{ML}})^2}{3N_k} \right) \tag{3}$$

The key modification is that the Hessian of the loss function and the $\mathrm{MLT}(G,\hat{\theta})$ function used to propagate uncertainty both have force information. We compare our method with Kitchin's method for an O_2 molecule in a box, with results summarized in Figure 1. The training region is collected for the O−O bond length (R_{OO}) between 1.05 and 1.35 Å, whereas the test set includes points from 0.95 to 1.45 Å. For each point, the loss is calculated using eq 2. The purely energy-based uncertainty estimates have significantly different slopes for points in the compressed bond regions (R_{OO} < 1.05 Å) and stretched bond region (R_{OO} > 1.35 Å). This is because large forces in the compressed region significantly affects the loss value. The correlation inconsistency observed in the compressed and stretched regions poses a challenge in practical applications. Inclusion of force information in our method mitigates this problem. The calculated uncertainty for both regions exhibits a consistent correlation with true loss values.

It is important to mention that the correlations in Figure 1 do not have a slope near unity. This is a consequence of an unknown calibration of the covariance matrix, which is the inverse of the loss Hessian scaled by a calibration constant. In principle, this constant can be tuned if desired. We keep the calibration constant as the optimized loss value of the neural network, as suggested by Kitchin. This constant does not play a critical role in determining whether an image is uncertain. Instead, a relative ratio to the maximum uncertainty of the neural network predictions on any training point is used as the uncertainty threshold (δ_{th}) to decide if an image is sufficiently uncertain.[6,18]

$$\delta_{\mathrm{th}} = \varepsilon \cdot \max\,[\delta_i] \tag{4}$$

Here ε is a constant that controls the reliability of the prediction, and δ_i is the uncertainty from eq 2 on any image in the training set. This approach is based on the assumption that the uncertainty and loss values of the image are linearly correlated. If the uncertainty exceeds δ_{th}, defined in eq 4, that indicates the new image is beyond the training region. Consequently, the model needs to be retrained. Otherwise, the new image is considered close to the training region and the results from the model are reliable. In this work, the value of ε has been kept constant at 1.50 which exhibits best performance for the Au FCC (643) system (Figure S1a). In contrast, an energy-based uncertainty method required a much larger ε to achieve comparable performance (Figure S1b) due to the absence of force information.

2.2. Uncertainty Guided Active Learning Geometry Optimization Algorithm. The active learning framework for geometry optimization is illustrated in Figure 2. The optimizer consists of two main parts: uncertainty-guided ML optimization (denoted as ML + UQ optimizer) and ML-Hessian preconditioned refinement. The uncertainty-guided optimization employs the delta method for uncertainty estimation to systematically refine the MLP and achieve a soft convergence to 0.05 eV/Å. The process begins with a single-point DFT calculation of the system to obtain the energy and forces. Subsequently, an MLP is trained, and uncertainty threshold (δ_{th}) is calculated using eq 4. One geometry optimization step is then performed on the ML PES and the uncertainty (δ) of the MLP on the new geometry is calculated. If δ exceeds δ_{th}, the DFT forces and energy of the new geometry is calculated, and the MLP is refined using the new geometry data. This iterative process continues until the system has loosely converged with a DFT maximum atomic force ($F_{\mathrm{max}}^{\mathrm{DFT}}$) below 0.05 eV/Å. To mitigate model fitting errors, a tighter convergence criterion of 0.01 eV/Å is used for the optimization on the machine learning PES.

Once the optimizer has loosely converged, the ML Hessian of the potential energy is calculated and used to precondition the BFGS optimizer in the low force region to a tighter convergence to the critical point ($F_{\mathrm{max}}^{\mathrm{DFT}}$ < 0.01 eV/Å). Near the minimum, the ML models start to suffer from insufficient precision. In our hybrid method, we switch to accurate forces from DFT, but use the MLP to calculate the Hessian of the potential energy to precondition the BFGS algorithm to accelerate convergence in the harmonic region.

Two safety checks are included to consider possible failures of the MLP in describing the PES. If a maximum number of optimization steps on the surrogate model is reached (set to 100 by default) without the system reaching ($F_{\mathrm{max}}^{\mathrm{ML}}$) < 0.01 eV/Å, the last image is used to refine the ML model. This ensures

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

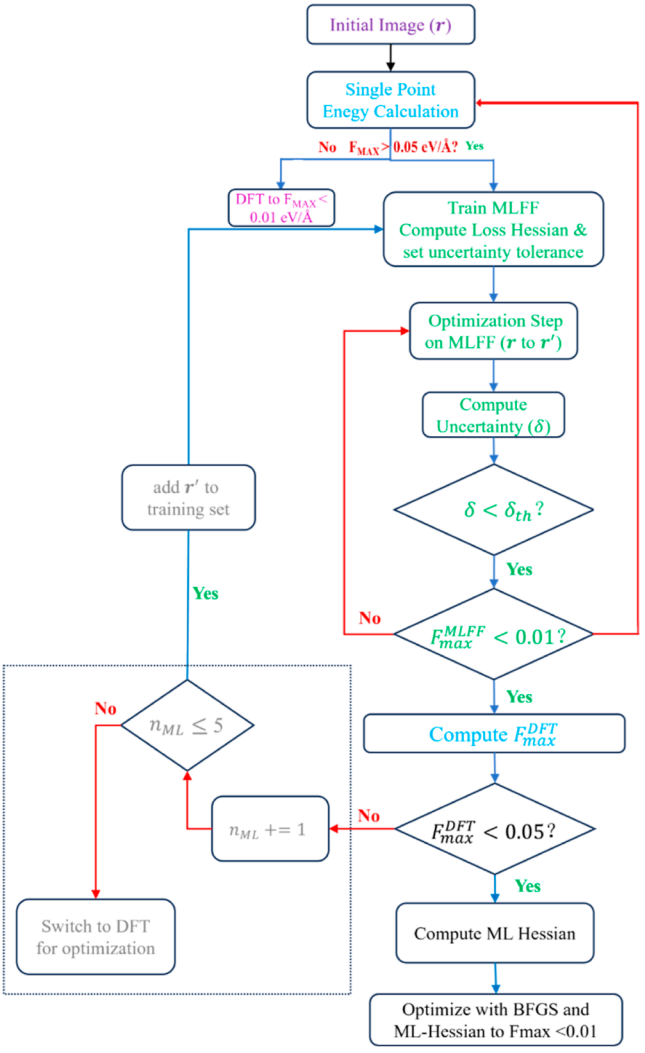


Figure 2. Uncertainty-method guided active learning geometry optimizer framework. The unit of force is eV/Å.

that the algorithm terminates, even if the surrogate model is not sufficiently smooth to reach a maximum atomic force of 0.01 eV/Å. Additionally, if the number of failures (n_{fail}) reaches a specific threshold value (n_{th} = 5 by default), the ML model is considered unreliable, and a switch to pure DFT is made (gray dashed line in Figure 2). Failure of the MLP is flagged when the ML maximum atomic force ($F_{\text{max}}^{\text{ML}}$) on a given geometry is below 0.01 eV/Å, yet $F_{\text{max}}^{\text{DFT}}$ on the same image is above 0.05 eV/Å. This safety check is for rare cases, such as when the ML input parameters are insufficient, or the MLP cannot approximate the true PES. This safety check was not triggered in any of the tests or examples presented here. The primary active learning framework presented here was sufficiently accurate to minimize to the DFT minimum at the tighter precision (0.01 eV/Å) for all systems tested.

2.3. Application Systems. Our algorithm was tested on five systems: an Rh FCC (211) surface, Au FCC (643) surface, bulk W BCC with a vacancy, $Pd_{13}H_2$ cluster, and a $Pd_{20}O_4$ cluster. The Au FCC (643) surface initial structure was obtained from Kitchin's work.[31] Initial structures for other systems were randomly displaced from the corresponding local minimum using displacements drawn from a normal distribution with a standard deviation of 0.1 Å. Upon displacement, each structure has a starting $F_{\text{max}}^{\text{DFT}}$ > 0.65 eV/Å.

DFT calculations were done with the Vienna Ab initio Simulation Package (VASP),[44,45] with the Perdew–Burke–Ernzerhof generalized gradient approximation exchange correlation funtional.[46] Input files and output results for each system are given in the Supporting Information.

2.4. Optimizer Hyperparameters. For our tests, we used a Behler Parinello neural network (BPNN), as implemented in the PyAMFF[5] package. A key challenge with neural network-based methods is the need to optimize the various hyperparameters for each system of interest. For user friendliness and generality, we have attempted to keep as many hyperparameters constant across systems as possible. The force coefficient (the relative weight of force and energy in the loss function) has been kept constant at a value of 1.40 Å^2. Each machine learning model was trained to a maximum of 2500 iterations steps or until the loss gradient of the neural network was below 0.0001 eV^2 for each parameter. The activation function used was tanh. The ML training was done with L-BFGS, as implemented in Scipy.[47] For BFGS preconditioning, the ML Hessian was modified to be positive definite (details provided in the Supporting Information). In general, the eigenvalue shift ratio used for all the systems in this work was 0.04, except for the $Pd_{20}O_4$ cluster, where we used 0.1. A table of the hyperparameters is available in the Supporting Information (Table S1).

The focus in this work has been on active learning for geometry optimization starting from a single configuration. When optimizing a structure for which similar minimizations have previously been done, sharing information over previously trained models can lead to further speed up, as shown by Kitchin and Ulissi.[23,24,31]

3. RESULTS AND DISCUSSION

3.1. ML + UQ Optimizer to Accelerate Geometry Optimization in the High Force Region. We initially tested the uncertainty-guided ML optimization algorithm on a Au FCC (643) surface and a $Pd_{13}H_2$ cluster. Our method relaxes the structure to a loose convergence criterion of 0.05 eV/Å. To ensure reproducibility, we conducted ten minimization runs for each system, all starting from the same initial configuration and identical neural network hyperparameters, but with random initializations of the neural network weights and biases.

Figure 3 shows the optimization progression for the Au FCC (643) surface and the $Pd_{13}H_2$ cluster, comparing the uncertainty guided ML optimizer with the L-BFGS[27] optimizer implemented in the VTST code[48] (denoted as VTST–LBFGS). Clearly, the uncertainty-guided ML optimizer outperforms the VTST–LBFGS optimizer. For Au FCC (643) and $Pd_{13}H_2$ the VTST–LBFGS optimizer takes 17 and 38 DFT steps respectively to converge to 0.05 eV/Å. Jumps in the force and energy are observed during this optimization procedure. In contrast, the ML optimizer shows a monotonic drop in the energy and forces and requires 7 and 18 DFT evaluations to achieve the convergence criterion (in one of the ten sample runs), respectively. This reduces the DFT force calls by 59% for the gold system, and 53% for the palladium hydrogen cluster.

We further compared the ML + UQ optimizer with other DFT optimizers, including the LBFGS,[49] quick-min (QM),[53] and conjugate gradient descent (CG)[52] optimizers as implemented in the VTST code,[48] BFGS,[49] MDMin, and

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022–10033

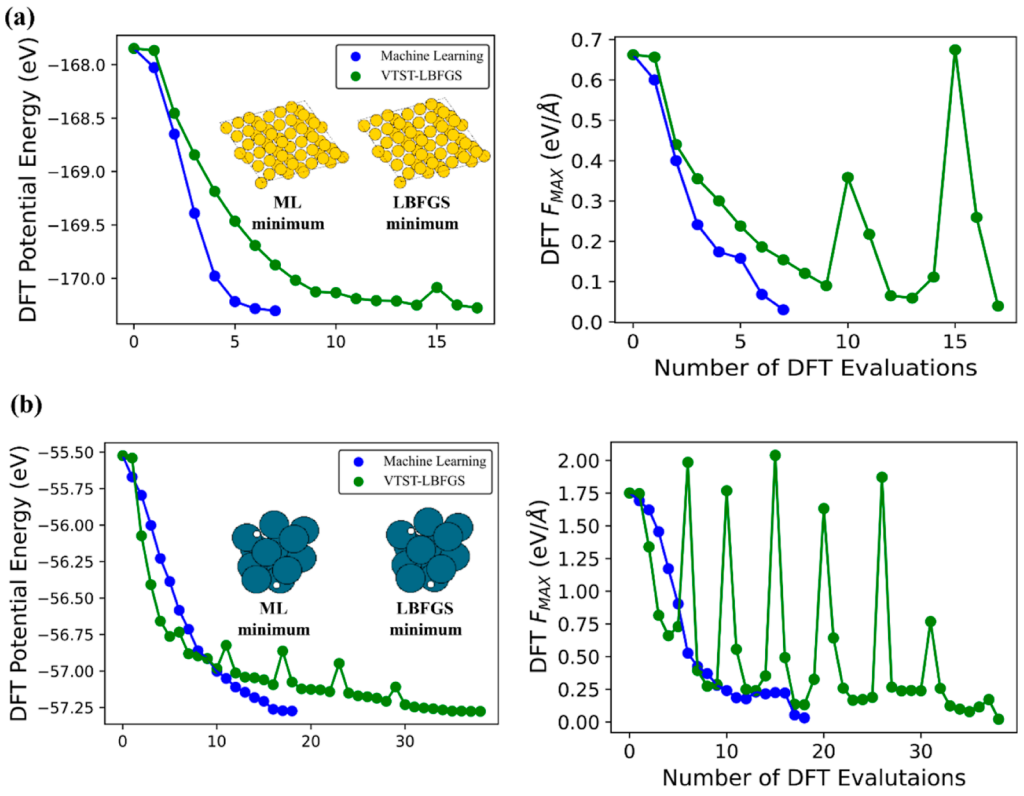


Figure 3. Variation of the potential energy (left) and maximum atomic force (F_{max}) (right) with the number of DFT evaluations during the optimization of the (a) Au FCC (643) and (b) $Pd_{13}H_2$ system using the ML + UQ and pure DFT using the L-BFGS optimization method implemented in the VTST code.[48] The inset images represent the corresponding minimum structures. Gold, dark blue and white balls are Au, Pd and H, respectively.

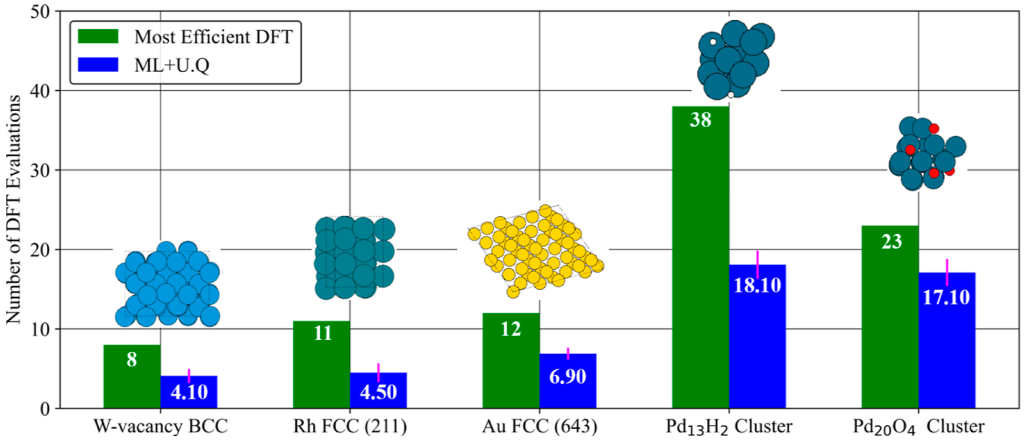


Figure 4. Comparison of number of DFT force calls between ML + UQ optimizer (averaged over 10 runs) and the pure DFT method in optimizing to $F_{max}^{DFT} < 0.05$ eV/Å, for the five systems tested. Light blue, turquoise, gold, dark blue, red and white circles are W, Rh, Au, Pd, H, and O respectively.

FIRE[50] methods as implemented in ASE,[51] and VASP's[44,45] damped-MD and RMM−DIIS implementations. The performance of our optimizer, in comparison to the most efficient pure DFT optimization method for the respective system, is summarized in Figure 4 (representative convergence plots for each system are shown in Figure S2).

The Rh FCC (211) surface, consisting of 36 atoms in a hexagonal arrangement, is relaxed by MDMin from ASE in 11 DFT force evaluations, while the ML + UQ optimizer takes an average of 4.5 DFT force calls. The 95-atom BCC bulk Tungsten system with a vacancy is considered a more challenging system due to the need to include second neighbors for an accurate description of the PES. The ML + UQ optimizer requires an average of only 4.1 DFT force calls to converge, around half the number of that required for RMM-DIIS, which is the most efficient DFT optimizer for this system. For the Au FCC (643) surface, RMM−DIIS requires 12 DFT calls, whereas the ML + UQ optimizer takes on average 6.9 DFT evaluations.

The presence of two distinct elements in the cluster systems leads to a wider range of bond frequencies and force constants, making optimization more challenging. For the $Pd_{13}H_2$ cluster,

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

VTST−LBFGS requires 38 DFT calls, whereas the ML + UQ method is able to converge in 18.1 single point calculations. For the $Pd_{20}O_4$ cluster, among the DFT optimizers employed, the ASE−BFGS takes least number of force calls and converges in 23 force calls, while our ML + UQ optimizer reduces the number of force calls to 17.1. Despite the higher number of evaluations compared to the periodic systems, the ML + UQ method significantly decreases the number of DFT calls required.

The performance of the ML optimizer can depend on the random initializations of neural network parameters, potentially affecting stability of the ML + UQ optimizer. However, based on 10 runs for each system, the variation in number of force calls is small (as shown by magenta lines in Figure 4). Therefore, the ML method with modified uncertainty quantification consistently provides reliable acceleration across all systems, effectively converging from the high force region to the harmonic region.

To gain deeper insight into how the algorithm works, we revisited the optimization procedure for the Rh(211) system given its relatively lower computational cost compared to the other systems. Figure 5 shows the variation in energy, maximum force error, uncertainty and uncertainty tolerance over optimization steps, respectively. Note that optimization on the ML surrogate model (denoted as the MLP loop) is counted in the optimization steps.

As shown in Figure 5c, the first three MLP refinements were triggered (marked as 1, 2 and 3) when the uncertainty exceeded the tolerance value (red line in Figure 5c). Although the potential energy from the NNP model aligned well with the DFT reference in the first two triggers (Figure 5a), the force error increased significantly, indicating poor reliability of the MLP. Upon each refinement, the force error consistently decreased and rose again as the structure moved away from the training region, leading to the next refinement (Figure 5b). Thus, each trigger of MLP refinement corresponds to the maximum force error in that MLP loop.

The fourth refinement of the MLP was triggered not due to uncertainty, but because the maximum number of optimization steps (set to 100) using the MLP was reached without achieving the machine learning converge criterion. The resulting DFT maximum atomic force (0.062 eV/Å) was higher than the DFT convergence criterion of 0.05 eV/Å, so this structure was considered uncertain, and was added to the training set for model refinement. Overall, our uncertainty quantification algorithm accurately identified points for model refinement. As optimization proceeded, the uncertainty tolerance decreased consistently, indicating that the model became increasingly accurate in describing the system's local PES.

3.2. ML-Hessian Preconditioned Optimizer for Geometry Refinement. We have shown that the ML + UQ optimizer efficiently accelerates convergence of the atomic systems to the harmonic region with atomic forces below 0.05 eV/Å. However, further relaxation of geometry to lower convergence criterion is challenging. In the harmonic regions where forces are small, localized models such as in the BPNN can lack the precision needed to accurately converge to the minimum. To mitigate this problem, we switch to a ML-Hessian preconditioned BFGS optimizer and conduct optimization using the true energy and forces of the system. Figure 6 shows the comparative performance (orange bars) of pure DFT optimizers and our ML-Hessian preconditioned

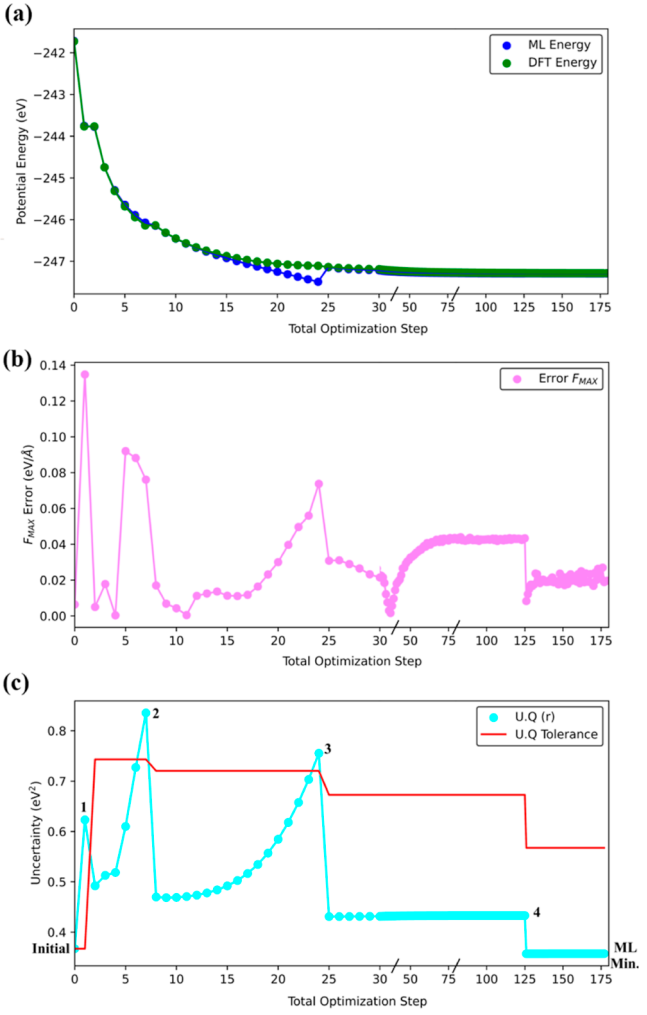


Figure 5. ML + UQ optimization of Rh FCC (211). (a) Variation of the ML and DFT potential energies, (b) error in the ML and DFT maximum atomic forces, and (c) the uncertainty of structures (cyan dots) and acquisition criterion (red line). Corresponding DFT energies and atomic forces of structures were obtained after the hybrid optimization was converged.

BFGS optimizer in converging from the harmonic region (F_{max}^{DFT} < 0.05 eV/Å) to the precise minimum (F_{max}^{DFT} < 0.01 eV/Å) for the Au FCC (643) and $Pd_{13}H_2$ cluster systems. This is an extension of Figure S3, where the DFT optimizers are not reset so that the memory from previous data points are retained for curvature approximating methods including BFGS or CG. The blue bars represent the number of DFT force evaluations required to converge the system to the harmonic region, in the same way as in Figure S3.

Figure 6 shows that the ML-Hessian preconditioned BFGS optimizer facilitates convergence for the Au FCC (643) surface and $Pd_{13}H_2$ systems to minima with a tighter criterion of 0.01 eV/Å. For the Au FCC (643) surface, it requires only 2 extra steps for convergence, much fewer than 11 steps needed for the RMM−DIIS optimizer, which is the most efficient DFT optimizer for this system. For $Pd_{13}H_2$, the optimization performance in the harmonic region is similar between VTST−LBFGS (the most efficient pure DFT method) and ML-Hessian preconditioned BFGS. But this is still significant because the ML Hessians are obtained from the ML force field trained on an average of 18.1 DFT points, whereas the LBFGS

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

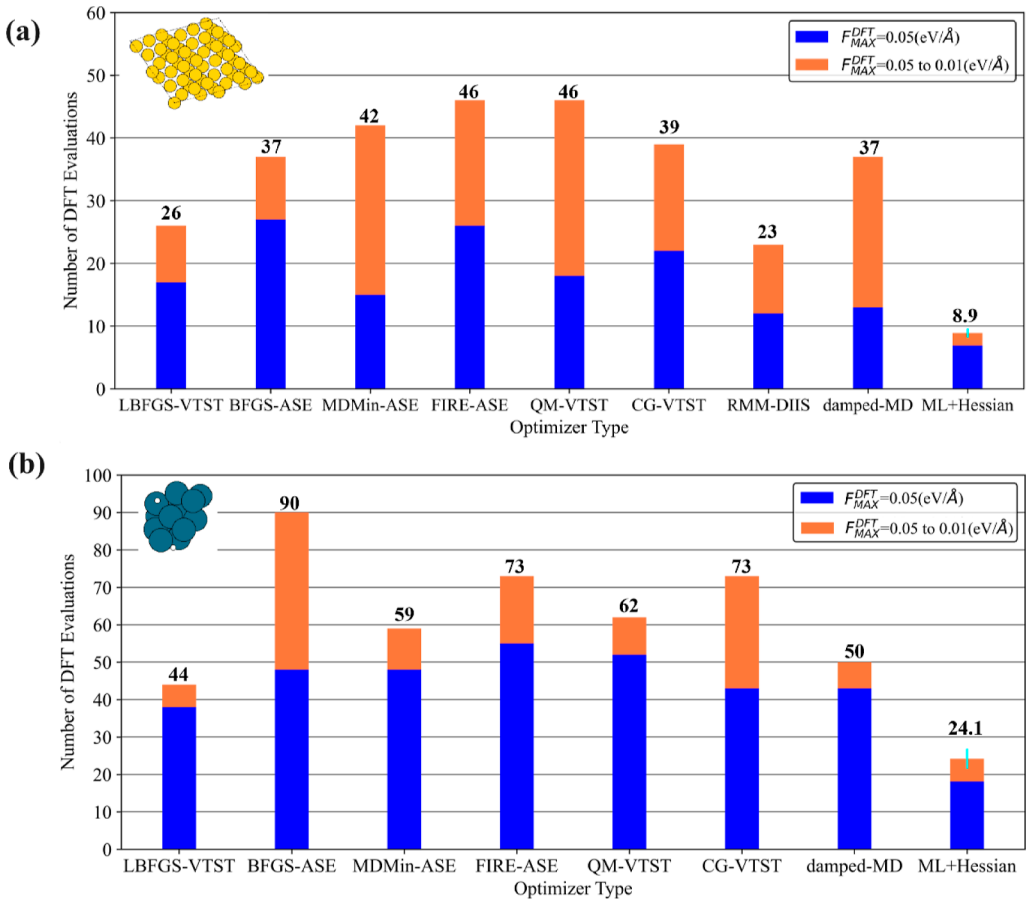


Figure 6. Comparison of number of DFT force evaluations between the ML-Hessian preconditioned optimizer (averages and sample standard deviations from 10 random runs) and other popularly used optimization methods for (a) the Au FCC (643) (b) $Pd_{13}H_2$ cluster systems. Gold, dark blue and white balls are Au, Pd and H, respectively. RMM−DIIS for $Pd_{13}H_2$ (not shown) does not converge in 200 DFT evaluations.

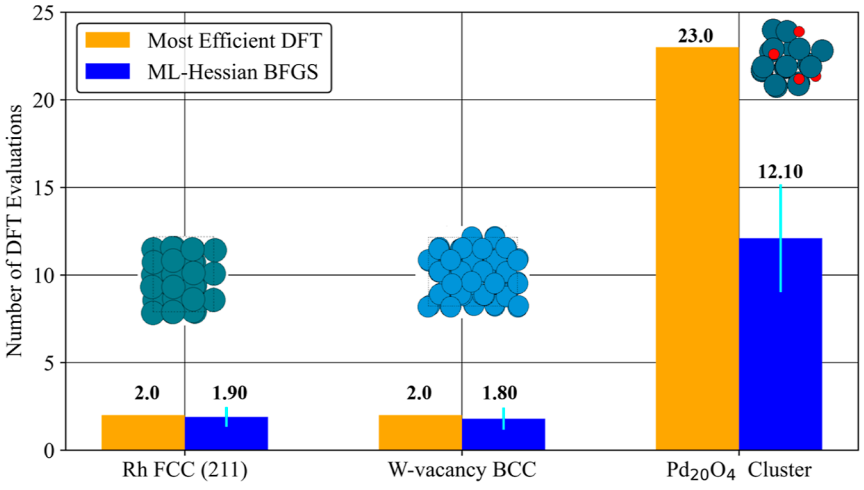


Figure 7. Comparison of the number of DFT calls needed for convergence from 0.05 to 0.01 eV/Å using the ML-Hessian preconditioned BFGS optimizer (averaged over 10 runs) and the most efficient pure DFT method for the Rh FCC (211), W-vacancy BCC and $Pd_{20}O_4$ cluster. Turquoise, light blue, dark blue and red circles are Rh, W, Pd and O, respectively.

Hessian has been approximated using 38 DFT points along the minimization trajectory.

We also continue the minimization for the Rh FCC (211), W-vacancy BCC and $Pd_{20}O_4$ cluster (Figure S4) to a tighter tolerance using the DFT and ML-Hessian preconditioned BFGS optimizer; the number of DFT calls is summarized in Figure 7. Using the most efficient pure DFT method, 2 extra steps are needed to converge Rh FCC (211) and W-vacancy BCC systems from 0.05 to 0.01 eV/Å. The ML-Hessian BFGS optimizer takes 1.9 steps for Rh FCC (211) and 1.8 steps for W-vacancy BCC on average over 10 runs. For the $Pd_{20}O_4$ cluster, for the overall most efficient DFT method (the FIRE method in Figure S4c) requires an additional 23 steps in the harmonic region to reach the precise minimum. Remarkably,

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

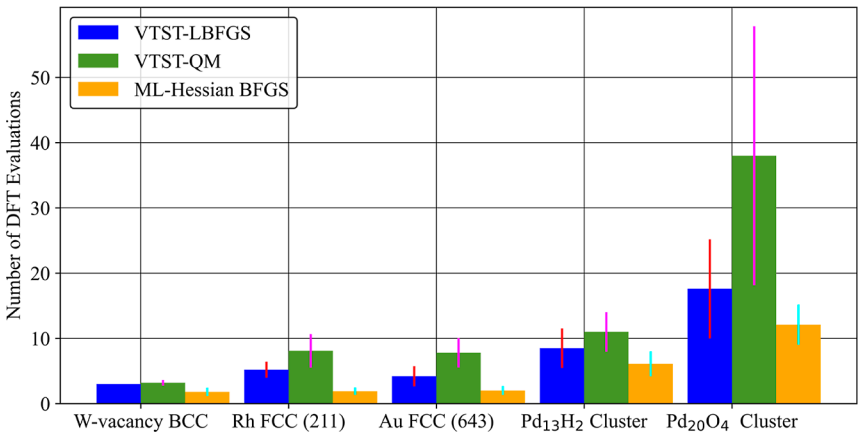


Figure 8. Average number DFT force evaluations (over 10 random runs) required to optimize to the minimum with F_{max}^{DFT} < 0.05 eV/Å obtained with the ML + UQ optimizer described in Part I, to F_{max}^{DFT} < 0.01 eV/Å, for each test system.

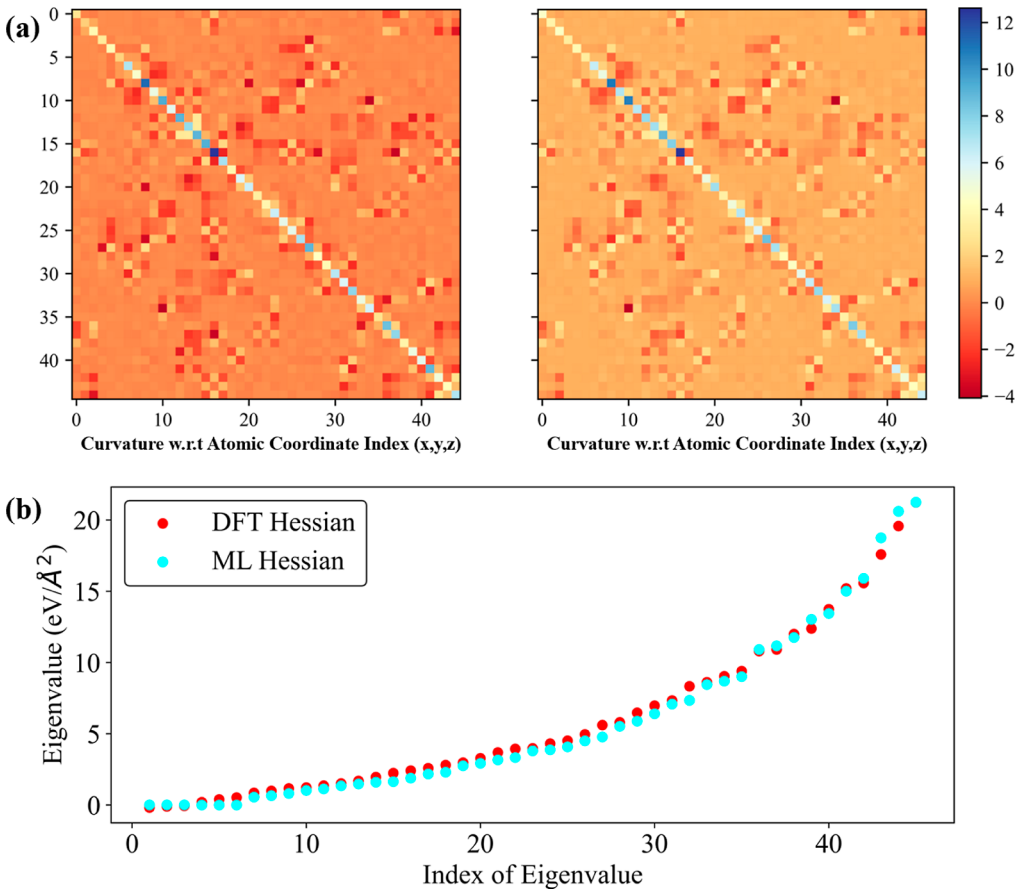


Figure 9. (a) Comparison of the DFT-Hessian (left) and ML-Hessian (right) from a $Pd_{13}H_2$ ML + UQ minimization to F_{max}^{DFT} < 0.05 eV/Å. Note that the first 6 rows correspond to H, and other 39 rows to Pd. (b) Spectrum of eigenvalues of the DFT and ML-Hessian.

with ML-Hessian as preconditioner, it takes only 12.1 steps on average over 10 runs. All ten runs for each system successfully minimized to the DFT-verified minima. More specifically, the maximum atomic displacement between each of the minima from our hybrid optimizer and the corresponding conjugate gradient minima was less than 0.05 Å, and the potential energy difference was less than 0.01 eV (see details in Figure S5). Overall, the BFGS optimizer with ML-Hessian as a preconditioner accelerates convergence of systems within the harmonic region.

The above tests in the harmonic region continue from previous individual simulations and start with different geometries, which can potentially affect performance. To further validate our results, we use the ML + UQ optimizer to converge the system to the harmonic region, and then switch from that geometry to one of the following methods: (i) a fresh VTST−LBFGS optimization, (ii) a Quick-Min optimization, or (iii) our ML-Hessian preconditioned BFGS optimization. This ensures all simulations start from the same geometry, eliminating performance dependence on the initial geometry.

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

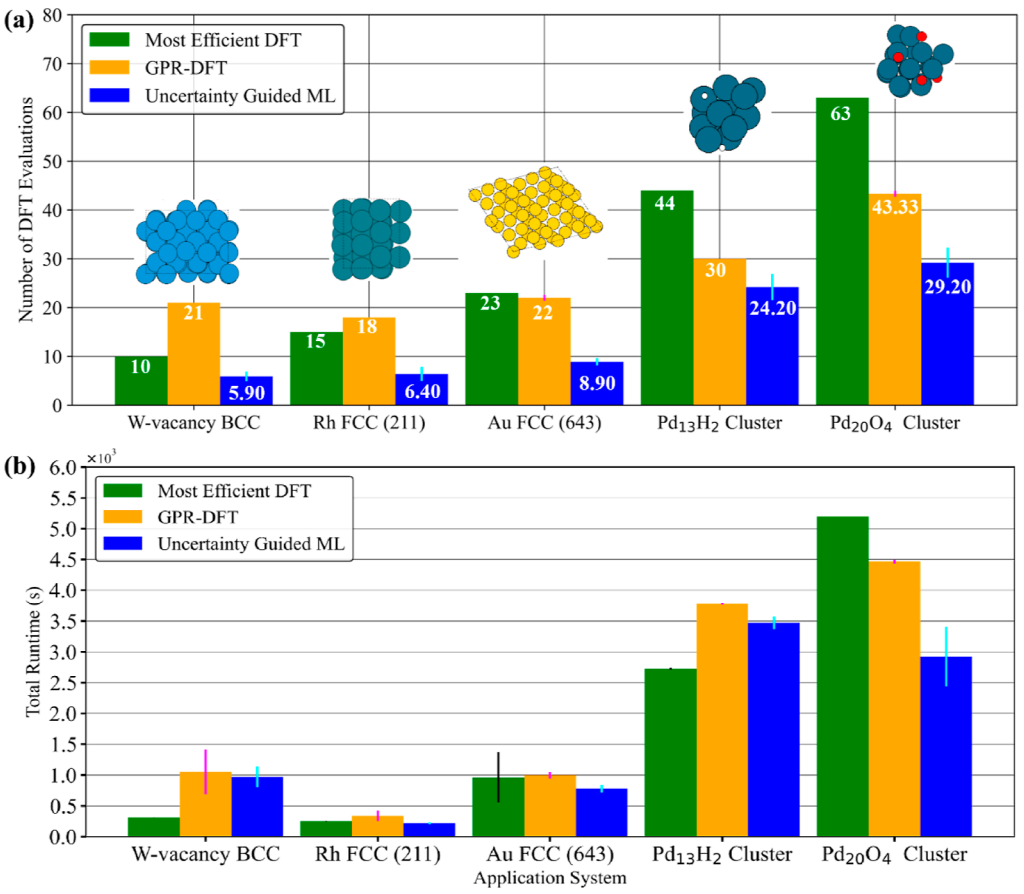


Figure 10. Comparison of (a) the number of DFT force calls and (b) total runtimes between the uncertainty guided ML optimizer (averaged over 10 runs for number of DFT evaluations, and 4 runs for total runtime), GPR-DFT[54] (averaged over 3 runs for both metrics), and the best pure DFT optimization method (averaged over 3 runs for runtimes) in optimizing to $F_{max}^{DFT} < 0.01$ eV/Å, for all systems. Surrogate model training and energy/force evaluations were performed serially on a single core, while DFT calculations were parallelized on 32 cores. For the $Pd_{20}O_4$ system, the electronic structure optimizer is switched to the conjugate gradient descent algorithm due to convergence issues of the Fast algorithm for the GPR−DFT optimizer. Likewise best performances of the DFT and our ML optimizer are reported. Light blue, turquoise, gold, dark blue, red and white circles are Pd, W, Au, O and H, respectively.

The VTST−LBFGS optimizer initializes the Hessian using a finite-difference steepest descent step and updates the Hessian with geometries encountered during optimization. Quick-Min is a force-based first order optimizer. The results are summarized in Figure 8. On average, the ML-Hessian BFGS approach is the most efficient, followed by the VTST−LBFGS optimizer. The first order Quick-Min method has the lowest performance due to a limited step size in the harmonic region caused by small atomic forces. Moreover, the ML-Hessian BFGS approach is more stable, as indicated by a smaller variation in the 10 runs (colored error bars in Figure 8).

We have demonstrated that our ML-Hessian preconditioned BFGS optimizer shows accelerated or similar quadratic convergence in harmonic regions compared to second order DFT methods, despite having access to fewer DFT configurations and the corresponding forces and energies. We attribute this performance to the ability of the ML potential to actively learn the true curvatures of the system with sufficient accuracy to help with the preconditioned quasi-Newton methods, despite having no explicit Hessian information included in the training process. Figure 9 compares the ML and DFT Hessian matrices, their eigenvalues, and the resulting frequencies for $Pd_{13}H_2$ upon convergence to the harmonic region. The ML Hessian matrix (right panel in Figure 9a) shows remarkable agreement with the DFT Hessian (left panel in Figure 9a). Eigenvalues and frequencies calculated from the ML force field also align well with those obtained from the reference DFT method. Similar agreement between ML and DFT Hessian information is observed for other systems (Figures S6−S9). Consequently, after ensuring positive definitiveness, the ML Hessians serve as good preconditioners for the BFGS optimizer, accelerating convergence in the harmonic regions.

3.3. Overall Performance of the Uncertainty Guided ML Algorithm. To summarize, the full (ML + UQ + H) method ensures accuracy of the MLP and convergence of the system to the harmonic region with $F_{max}^{DFT} < 0.05$ eV/Å and then uses the ML-Hessian preconditioned BFGS optimizer to further relax the geometry until $F_{max}^{DFT} < 0.01$ eV/Å. Figure 10 summarizes the total number of DFT force calls and total runtimes required for the method to optimize each system as compared to the most efficient benchmarked DFT optimizer and GPR-based active learning optimizer developed previously.[54] For the periodic systems, we have compared our hybrid optimizer with universal sparse preconditioners[55] (summarized in Table S2). The full ML optimizer rapidly converges all systems below F_{max}^{DFT} of 0.01 eV/Å, including simple metallic surfaces and multielement cluster systems. Compared to the most efficient DFT method for each system, the ML + UQ + H approach shows a 2−3 times DFT force call

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

saving and has small variation in the number of DFT force calls over 10 independent runs (Figure 10a). Moreover, our ML approach exhibit more consistent improvements in reducing the number of DFT calls compared to the GPR-based optimizer.

The ML + UQ + H optimizer reduces the number of DFT evaluations during structure optimization, but it shows limited improvement in total runtime (Figure 10b). A similar issue is observed with the GPR-based optimizer developed previously[54] (Figure 10b), which is less efficient than our ML + UQ + H optimizer. This is primarily due to inefficient preconditioning during the SCF calculations, stemming from significant structural differences between successive iterations. This issue is likely common to optimizers relying on ML surrogate models. Further method development, such as the efficient reuse of the electronic density and KS orbitals, needed to improve overall computational efficiency, will be investigated in future work. The UQ algorithm developed here is particularly valuable for efficient on-the-fly training of surrogate models in simulations requiring extensive DFT calculations, such as molecular dynamics simulations. Additionally, the ML-Hessian preconditioned BFGS provide a reliable acceleration in optimizing with on-the-fly optimizers near critical points.

For effective and robust application of the algorithm developed here, efforts have been made in this work to establish reliable default values for the hyperparameters associated with the ML-based optimization. These hyperparameters include neural-network structures, type of activation function, etc. Most of the hyperparameters are independent of the application system and have been kept constant (refer to section "Optimizer Hyperparameter" for details). In other systems, adjustment of these hyperparameters may be required, particularly those associated with the symmetry functions (fingerprints). For a better understanding of hyperparameter selection more benchmarking is required.

4. CONCLUSION

In this work, we developed an uncertainty-guided ML-DFT hybrid geometry optimizer that provides reliable acceleration in both the high- and low-force regions. Improving upon previously reported active learning approaches, we implemented an atomistic force-informed delta method for uncertainty estimation. This method demonstrates an efficient and easily applicable single model-based approach, thus eliminating the need for multiple models. Guided by this modified delta method, the ML model can be effectively refined and accelerate the convergence of atomic systems from high-force region to the low-force region (the harmonic region). Applying the ML Hessian as a preconditioner for the BFGS optimizer ensures the convergence of atomic systems to precise minima from the harmonic region, which is hard to achieve with a traditional active-learning approach.

We benchmarked our hybrid optimizer against eight commonly used optimizers and a GPR-accelerated active learning method for minimizing five different systems. The hybrid optimizer converges to the minimum for each system and offers a stable 2−3 times reduction in DFT force evaluations compared to the system-specific most efficient DFT optimizer. Our results suggest that further advancements in efficiently reusing electronic density and Kohn−Sham orbitals are needed to enhance overall computational efficiency.

■ ASSOCIATED CONTENT

Data Availability Statement

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.4c00953.

Additional details about fingerprint parameters and VASP input files for O_2, Rh FCC (211), W-vacancy BCC, Au FCC (643), the $Pd_{13}H_2$ and $Pd_{20}O_4$ clusters, convergence plots for Rh FCC (211), W-vacancy BCC, and $Pd_{20}O_4$, performance comparison of our developed ML-DFT hybrid optimizer with other pure DFT-based optimizers for Rh FCC (211), W-vacancy BCC and the $Pd_{20}O_4$ cluster, comparison of the DFT- and ML-Hessian for Rh FCC (211), W-vacancy BCC, Au FCC (643) and the $Pd_{20}O_4$ cluster (PDF)
(ZIP)

■ AUTHOR INFORMATION

Corresponding Authors

Graeme Henkelman − *Department of Chemistry, The University of Texas at Austin, Austin, Texas 78712, United States; Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712, United States;* orcid.org/0000-0002-0336-7153; Email: henkelman@utexas.edu

Lei Li − *Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China;* orcid.org/0000-0003-2882-2447; Email: lil33@sustech.edu.cn

Authors

Akksay Singh − *Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China; Department of Chemistry, The University of Texas at Austin, Austin, Texas 78712, United States; Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712, United States;* orcid.org/0000-0001-8593-0482

Jiaqi Wang − *Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.4c00953

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work is supported by Training Program of the Major Research Plan of the National Natural Science Foundation of China (92270103) and the National Key R&D Program of China (2022YFA1503102), Work in Austin was supported by the National Science Foundation (CHE-2102317) and the Welch Foundation (F-1841). The theoretical calculations are supported by Center for Computational Science and Engineering of Southern University of Science and Technology and the Texas Advanced Computing Center. A.S. would like to acknowledge Sung H. Jung and Ilgar Bagishov for insightful discussions regarding active learning and Eboni Williams and Caroline Leung for providing feedback on figures.

REFERENCES

(1) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(2) Behler, J. Constructing High-Dimensional Neural Network Potentials: A Tutorial Review. *Int. J. Quantum Chem.* **2015**, *115*, 1032−1050.

(3) Artrith, N.; Behler, J. High-Dimensional Neural Network Potentials for Metal Surfaces: A Prototype Study for Copper. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2012**, *85*, 045439.

(4) Khorshidi, A.; Peterson, A. A. Amp: A Modular Approach to Machine Learning in Atomistic Simulations. *Comput. Phys. Commun.* **2016**, *207*, 310−324.

(5) Li, L.; Ciufo, R. A.; Lee, J.; Zhou, C.; Lin, B.; Cho, J.; Katyal, N.; Henkelman, G. Atom-Centered Machine-Learning Force Field Package. *Comput. Phys. Commun.* **2023**, *292*, 108883.

(6) Pun, G. P. P.; Batra, R.; Ramprasad, R.; Mishin, Y. Physically Informed Artificial Neural Networks for Atomistic Modeling of Materials. *Nat. Commun.* **2019**, *10*, 2339.

(7) Artrith, N.; Urban, A. An Implementation of Artificial Neural-Network Potentials for Atomistic Materials Simulations: Performance for TiO2. *Comput. Mater. Sci.* **2016**, *114*, 135−150.

(8) 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*, 241722.

(9) Zeng, J.; Zhang, D.; Lu, D.; Mo, P.; Li, Z.; Chen, Y.; Rynik, M.; Huang, L.; Li, Z.; Shi, S.; Wang, Y.; Ye, H.; Tuo, P.; Yang, J.; Ding, Y.; Li, Y.; Tisi, D.; Zeng, Q.; Bao, H.; Xia, Y.; Huang, J.; Muraoka, K.; Wang, Y.; Chang, J.; Yuan, F.; Bore, S. L.; Cai, C.; Lin, Y.-C.; Wang, B.; Xu, J.; Zhu, J.; Luo, C.; Zhang, Y.; Goodall, R. E. A.; Liang, W.; Singh, A. K.; Yao, S.; Zhang, J.; Wentzcovitch, R. M.; Han, J.; Liu, J.; Jia, W.; York, D. M.; Weinan, E.; Car, R.; Zhang, L.; Wang, H. DeePMD-Kit v2: A Software Package for Deep Potential Models. *J. Chem. Phys.* **2023**, *159*, 054801.

(10) Shuaibi, M.; Hu, Y.; Lei, X.; Comer, B. M.; Adams, M.; Paras, J.; Chen, R. Q.; Musa, E.; Musielewicz, J.; Peterson, A. A.; Medford, A. J.; Ulissi, Z. AmpTorch: A Python Package for Scalable Fingerprint-Based Neural Network Training on Multi-Element Systems with Integrated Uncertainty Quantification. *J. Open Source Softw.* **2023**, *8*, 5035.

(11) Liu, M.; Kitchin, J. R. SingleNN: Modified Behler−Parrinello Neural Network with Shared Weights for Atomistic Simulations with Transferability. *J. Phys. Chem. C* **2020**, *124*, 17811−17818.

(12) Li, R.; Zhou, C.; Singh, A.; Pei, Y.; Henkelman, G.; Li, L. Local-Environment-Guided Selection of Atomic Structures for the Development of Machine-Learning Potentials. *J. Chem. Phys.* **2024**, *160*, 074109.

(13) Imbalzano, G.; Anelli, A.; Giofré, D.; Klees, S.; Behler, J.; Ceriotti, M. Automatic Selection of Atomic Fingerprints and Reference Configurations for Machine-Learning Potentials. *J. Chem. Phys.* **2018**, *148*, 241730.

(14) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, *104* (13), 136403.

(15) Bartók, A. P.; Csányi, G. Gaussian Approximation Potentials: A Brief Tutorial Introduction. *Int. J. Quantum Chem.* **2015**, *115*, 1051−1057.

(16) Vandermause, J.; Torrisi, S. B.; Batzner, S.; Xie, Y.; Sun, L.; Kolpak, A. M.; Kozinsky, B. On-the-Fly Active Learning of Interpretable Bayesian Force Fields for Atomistic Rare Events. *npj Comput. Mater.* **2020**, *6*, 20.

(17) Hajibabaei, A.; Kim, K. S. Universal Machine Learning Interatomic Potentials: Surveying Solid Electrolytes. *J. Phys. Chem. Lett.* **2021**, *12*, 8115−8120.

(18) Hajibabaei, A.; Myung, C. W.; Kim, K. S. Sparse Gaussian Process Potentials: Application to Lithium Diffusivity in Superionic Conducting Solid Electrolytes. *Phys. Rev. B* **2021**, *103*, 214102.

(19) Koistinen, O.-P.; Ásgeirsson, V.; Vehtari, A.; Jónsson, H. Minimum Mode Saddle Point Searches Using Gaussian Process Regression with Inverse-Distance Covariance Function. *J. Chem. Theory Comput.* **2020**, *16*, 499−509.

(20) Koistinen, O.-P.; Dagbjartsdóttir, F. B.; Ásgeirsson, V.; Vehtari, A.; Jónsson, H. Nudged Elastic Band Calculations Accelerated with Gaussian Process Regression. *J. Chem. Phys.* **2017**, *147*, 152720.

(21) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.* **2019**, *31*, 3564−3572.

(22) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*, 145301.

(23) Musielewicz, J.; Wang, X.; Tian, T.; Ulissi, Z. W. FINETUNA: Fine-Tuning Accelerated Molecular Simulations. *Mach. Learn. Sci. Technol.* **2022**, *3*, 03LT01.

(24) Wang, X.; Musielewicz, J.; Tran, R.; Ethirajan, S. K.; Fu, X.; Mera, H.; Kitchin, J. R.; Kurchin, R.; Ulissi, Z. W. Generalization of Graph-Based Active Learning Relaxation Strategies Across Materials. *Mach. Learn. Sci. Technol.* **2024**, *5*, 025018.

(25) Drautz, R. Atomic Cluster Expansion for Accurate and Transferable Interatomic Potentials. *Phys. Rev. B* **2019**, *99*, 014104.

(26) Dusson, G.; Bachmayr, M.; Csányi, G.; Drautz, R.; Etter, S.; van der Oord, C.; Ortner, C. Atomic Cluster Expansion: Completeness, Efficiency and Stability. *J. Comput. Phys.* **2022**, *454*, 110946.

(27) Xie, S. R.; Rupp, M.; Hennig, R. G. Ultra-Fast Interpretable Machine-Learning Potentials. *npj Comput. Mater.* **2023**, *9*, 162.

(28) Thompson, A. P.; Swiler, L. P.; Trott, C. R.; Foiles, S. M.; Tucker, G. J. Spectral Neighbor Analysis Method for Automated Generation of Quantum-Accurate Interatomic Potentials. *J. Comput. Phys.* **2015**, *285*, 316−330.

(29) Briganti, V.; Lunghi, A. Efficient Generation of Stable Linear Machine-Learning Force Fields with Uncertainty-Aware Active Learning. *Mach. Learn. Sci. Technol.* **2023**, *4*, 035005.

(30) Meziere, J. A.; Luo, Y.; Xia, Y.; Béland, L. K.; Daymond, M. R.; Hart, G. L. W. Accelerating Training of MLIPs through Small-Cell Training. *J. Mater. Res.* **2023**, *38*, 5095−5105.

(31) Yang, Y.; Jiménez-Negrón, O. A.; Kitchin, J. R. Machine-Learning Accelerated Geometry Optimization in Molecular Simulation. *J. Chem. Phys.* **2021**, *154*, 234704.

(32) Torres, J. A. G.; Jennings, P. C.; Hansen, M. H.; Boes, J. R.; Bligaard, T. Low-Scaling Algorithm for Nudged Elastic Band Calculations Using a Surrogate Machine Learning Model. *Phys. Rev. Lett.* **2019**, *122*, 156001.

(33) Gu, S.; Wang, H.; Zhou, X. Active Learning for Saddle Point Calculation. *J. Sci. Comput.* **2022**, *93*, 78.

(34) Peterson, A. A. Acceleration of Saddle-Point Searches with Machine Learning. *J. Chem. Phys.* **2016**, *145*, 074106.

(35) Peterson, A. A.; Christensen, R.; Khorshidi, A. Addressing Uncertainty in Atomistic Machine Learning. *Phys. Chem. Chem. Phys.* **2017**, *19*, 10978−10985.

(36) Schran, C.; Brezina, K.; Marsalek, O. Committee Neural Network Potentials Control Generalization Errors and Enable Active Learning. *J. Chem. Phys.* **2020**, *153*, 104105.

(37) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. Methods for comparing uncertainty quantifications for material property predictions. *Mach. Learn. Sci. Technol.* **2020**, *1*, 025006.

(38) Zhou, Y.; Yang, B. Uncertainty Quantification of Predicting Stable Structures for High-Entropy Alloys Using Bayesian Neural Networks. *J. Energy Chem.* **2023**, *81*, 118−124.

(39) Nilsen, G. K.; Munthe-Kaas, A. Z.; Skaug, H. J.; Brun, M. Epistemic Uncertainty Quantification in Deep Learning Classification by the Delta Method. *Neural Network.* **2022**, *145*, 164−176.

(40) Gawlikowski, J.; Tassi, N.; Ali, M.; Lee, J.-S.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; Shahzad, M.; Yang, W.; Bamler, R.; Zhu, X. X. A Survey of Uncertainty in Deep Neural Networks. *Artif. Intell. Rev.* **2023**, *56*, 1513−1589.

(41) Kallus, N.; McInerney, J. The Implicit Delta Method. *NeurIPS* **2022**, *35*, 37471−37483.

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033

(42) Wen, M.; Tadmor, E. B. Uncertainty Quantification in Molecular Simulations with Dropout Neural Network Potentials. *npj Comput. Mater.* **2020**, *6*, 124.
(43) Zhan, N.; Kitchin, J. R. Uncertainty Quantification in Machine Learning and Nonlinear Least Squares Regression Models. *AIChE J.* **2021**, *68*, No. e17516.
(44) Kresse, G.; Hafner, J. Ab Initio Molecular Dynamics for Liquid Metals. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1993**, *47* (1), 558−561.
(45) Kresse, G.; Furthmüller, J. Efficiency of Ab-Initio Total Energy Calculations for Metals and Semiconductors Using a Plane-Wave Basis Set. *Comput. Mater. Sci.* **1996**, *6*, 15−50.
(46) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(47) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, I.·.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261−272.
(48) Sheppard, D.; Terrell, R.; Henkelman, G. Optimization Methods for Finding Minimum Energy Paths. *J. Chem. Phys.* **2008**, *128*, 134106.
(49) Byrd, R. H.; Lu, P.; Nocedal, J.; Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190−1208.
(50) Bitzek, E.; Koskinen, P.; Gähler, F.; Moseler, M.; Gumbsch, P. Structural Relaxation Made Simple. *Phys. Rev. Lett.* **2006**, *97*, 170201.
(51) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The Atomic Simulation Environment-a Python Library for Working with Atoms. *J. Phys.: Condens. Matter* **2017**, *29*, 273002.
(52) Hestenes, M. R.; Steifel, E. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409.
(53) Jónsson, H.; Mills, G.; Jacobsen, K. W. In *Classical and Quantum Dynamics in Condensed Phase Simulations*; Berne, B. J., Ciccotti, G., Coker, D. F., Eds.; World Scientific: Singapore, 1998; pp 385−404.
(54) del Río, E. G.; Jacobsen, K. W. J. J.; Jacobsen, K. Local Bayesian optimizer for atomic structures. *Phys. Rev. B* **2019**, *100*, 104103.
(55) Packwood, D.; Kermode, J.; Mones, L.; Bernstein, N.; Woolley, J.; Gould, N.; Ortner, C.; Csányi, G. A universal preconditioner for simulating condensed phase materials. *J. Chem. Phys.* **2016**, *144* (16), 164109.

https://doi.org/10.1021/acs.jctc.4c00953
J. Chem. Theory Comput. 2024, 20, 10022−10033