PHYSICAL REVIEW B **100**, 014105 (2019)

Editors' Suggestion

On-the-fly machine learning force field generation: Application to melting points

Ryosuke Jinnouchi,[1,2] Ferenc Karsai,[3] and Georg Kresse[1]
[1] *University of Vienna, Department of Physics, Sensengasse 8/16, Vienna, Austria*
[2] *Toyota Central R&D Labs., Inc., 41-1 Yokomichi, Nagakute, Aichi 480-1192, Japan*
[3] *VASP Software GmbH, Sensengasse 8, Vienna, Austria*

(Received 25 April 2019; revised manuscript received 1 July 2019; published 17 July 2019)

An efficient and robust on-the-fly machine learning force field method is developed and integrated into an electronic-structure code. This method realizes automatic generation of machine learning force fields on the basis of Bayesian inference during molecular dynamics simulations, where the first-principles calculations are only executed, when new configurations out of already sampled datasets appear. The developed method is applied to the calculation of melting points of Al, Si, Ge, Sn and MgO. The applications indicate that more than 99% of the first-principles calculations are bypassed during the force field generation. This allows the machine to quickly construct first-principles datasets over wide phase spaces. Furthermore, with the help of the generated machine learning force fields, simulations are accelerated by a factor of thousand compared with first-principles calculations. Accuracies of the melting points calculated by the force fields are examined by thermodynamic perturbation theory, and the examination indicates that the machine learning force fields can quantitatively reproduce the first-principles melting points.

DOI: 10.1103/PhysRevB.100.014105

I. INTRODUCTION

The quantitative prediction of first-order phase transitions of real materials from first-principles (FP) calculations is a long-standing issue in condensed matter physics. The phase transition point is simply located at a temperature-pressure point where Gibbs free energies of two phases become identical. Its prediction is, however, quite challenging. Direct simulations using molecular dynamics (MD) and Monte Carlo methods are not computationally tractable because of the long timescale nature of the phase transitions. Several approaches were proposed in order to solve this timescale issue. One is an indirect approach on the basis of the thermodynamic integration method [1–3]. In this approach, the Gibbs free energy of a single phase is calculated by integrating a chemical potential derivative along a reversible thermodynamic path that connects from a simple statistical models to a realistic interacting model. Another one is a direct approach, where the coexistence point of the solid-liquid interface is directly explored by MD [4–9]. There is an alternative approach [10] that transforms this out-of-equilibrium direct simulation to an equilibrium simulation by introducing a bias potential pinning the system in an interfacial state. In this third approach, the coexistence point is determined by using the free energy difference obtained from the mean force of the bias potential. However, all these methods need significant computational resources.

Recently developed machine learning force field (MLFF) techniques [11–15] have the potential to solve this problem. In those techniques, the potential energy of the system is described as a function of structural descriptors that map the $3N$-dimensional structural information onto a lower-dimensional descriptor space, and the function is optimized to reproduce the FP data. High flexibility of the descriptors and this function allow for an accurate reproduction of the FP data, and the simulations are orders of magnitude faster using the generated force fields than the FP simulations.

However, applications of machine learning (ML) approaches are still limited to few simple materials. The difficulties are mainly in the force field generation process. Carefully selected reference datasets and a tremendous amount of FP calculations on typically 2000–12 000 structures [11,15–19] are needed in order to train force fields. Furthermore, the data selection and parameter optimization are complex and involve a quite large number of trial and error steps.

On-the-fly force field generation [20–24] has the potential to overcome these limitations. In this method, energy, forces, and/or the stress tensor as well as their uncertainties are computed by Bayesian inference or relevant statistical methods during an MD simulation. If the uncertainties are judged to be small, the computed energy, forces, and the stress tensor are used to integrate the equations of motions. If the uncertainties are judged to be large, FP calculations are executed in order to obtain new data that are then used to refine the force field. This error estimation and judgment step can realize efficient explorations of wide phase spaces and systematic data selection.

In this paper, we present an efficient and robust algorithm that can be applied to liquid-solid phase transitions of a wide variety of materials. In Sec. II, theories and equations used in our on-the-fly algorithm are presented. In Sec. III, method and parameters used in the simulations of the phase transitions are described. In Sec. IV, the calculated melting points are presented before we finally conclude the paper in Sec. V.

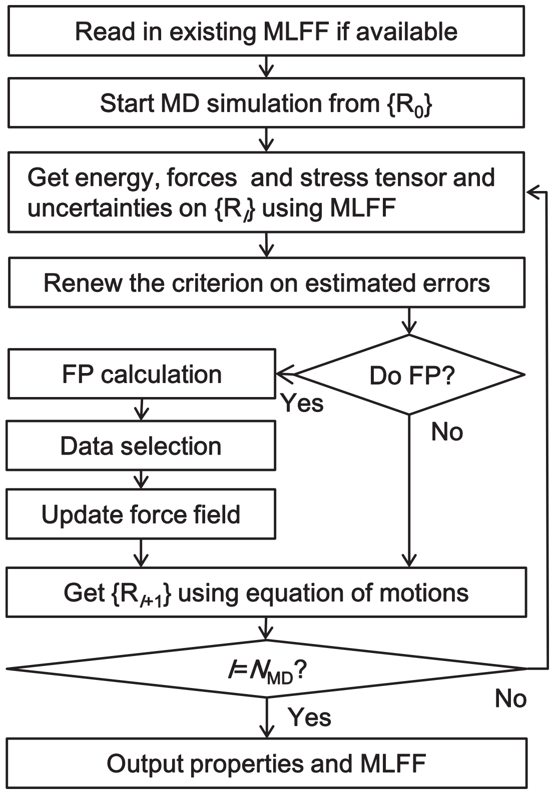

FIG. 1. Flowchart of our on-the-fly machine learning force field generation scheme.

II. METHOD: ON-THE-FLY FORCE FIELD GENERATION

In this section, after describing an outline of our on-the-fly MLFF generation scheme, the necessary methodologies composing this scheme are presented. For a concise presentation of the methodologies, we define *structure datasets* and *local configurations*. A single *structure dataset* consists of the Bravais lattice, the atomic positions, the total energy, the forces and the stress tensor for one specific structure calculated by the FP method. We will label these datasets using the superscript α. For each atom in the structure, a *local configuration* around this atom can be determined. This local configuration is mapped onto a set of descriptors describing the local environment around each atom as will be explained later. Local structures and the central atom in a local structure are labeled using indices i or i_B. Several structure datasets and local configurations are selected, and the ML force field is fitted to those. The selected datasets and configurations are referred to as *reference structure datasets* and *local reference configurations*, respectively.

A. Outline of the on-the-fly force field generation

Figure 1 shows the flowchart of our on-the-fly force field generation scheme. In our scheme, a force field is generated during MD simulations, as outlined below.

(1) The machine predicts the energy, forces, stress tensor, and their uncertainties on a given structure using the yet available force field.

(2) The machine decides whether to execute the FP calculation or not. The decision is done on the basis of the uncertainty in the prediction and a history of previous samplings. If the machine decides not to execute the FP calculations, the algorithm skips to step 5. Otherwise, it continues with step 3.

(3) The FP calculation is executed on the given structure, and the obtained structure dataset is stored as a candidate for a new reference structure dataset.

(4) If the number of the newly collected structures reaches a certain threshold, or if the uncertainty in the prediction becomes too large, the machine updates the set of reference structure datasets and local reference configurations and generates a new force field.

(5) Atomic positions and velocities are updated. If the machine judges that the force field is unreliable, the FP energy, forces and stress tensor are used. Otherwise those provided by the force field are used. Afterwards the machine returns to step 1 until the end of the MD simulation is reached ($I = N_{\mathrm{MD}}$ in Fig. 1).

This scheme needs several key methodologies: an accurate description of the potential energy surface, an optimization of the parameters in the force field, evaluation of the uncertainty, setting of the threshold for the uncertainty, sparsification, and data selections. All these ingredients were implemented within the Vienna *ab initio* simulation package (VASP) [25,26]. Their details are explained in the following sections.

B. Descriptor

Our description of the potential energy surface is similar to that adopted in the Gaussian approximation potential (GAP) [12] with the smooth overlap of atomic positions (SOAP) [27] as a similarity measure. Several new features are, however, introduced to make the on-the-fly force field generation process more efficient and robust. In order to explain them in a concise manner, we formulate the energy and descriptor in this section. Here, we show for simplicity the equations only for single element systems, however, the extension to multi-element systems is straightforward.

In the method presented in this work, the potential energy U of a structure with N_a atoms is approximated as a summation of local energies U_i as

$$U = \sum_{i=1}^{N_a} U_i. \tag{1}$$

Each local energy U_i is assumed to be fully determined by the local environment around atom i. To represent the local environment, the distribution of other atoms around the atom i is an obvious starting point. This distribution is represented by the probability density ρ_i to find another atom j at the position $\mathbf{r}$ around the atom i within a radius R_{cut}. It is defined as

$$\rho_i(\mathbf{r}) = \sum_{j=1}^{N_a} f_{\mathrm{cut}}(r_{ij}) g(\mathbf{r} - \mathbf{r}_{ij}), \tag{2}$$

where f_{cut} is a cutoff function that smoothly removes the information outside the radius R_{cut}. The position vector of the atom i is denoted by $\mathbf{r}_i$, and $r_{ij} = |\mathbf{r}_{ij}| = |\mathbf{r}_j - \mathbf{r}_i|$ is the distance between two atoms, and $g(\mathbf{r})$ is the delta function $\delta(\mathbf{r})$. In SOAP, the delta function is replaced by a normalized

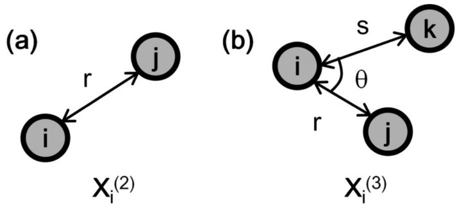

FIG. 2. (a) Radial and (b) angular descriptors.

Gaussian function as

$$g(\mathbf{r}) = \frac{1}{\sqrt{2\sigma_{\text{atom}}\pi}}\exp\left(-\frac{|\mathbf{r}|^2}{2\sigma_{\text{atom}}^2}\right). \tag{3}$$

The local energies U_i are functionals of the density ρ_i, $U_i{=}F[\rho_i(\mathbf{r})]$. So the simplest numerical approach to implement this procedure would be to develop $\rho_i(\mathbf{r})$ into a finite basis set and express F as a function of the coefficients. The drawback of this is that F would not possess rotational invariance. Therefore, it is expedient to introduce intermediate functions—usually called *descriptors*—that depend on $\rho_i(\mathbf{r})$. These intermediate functions should be invariant under rotations (as well as translations). The simplest rotationally invariant descriptor is the radial distribution function defined as

$$\rho_i^{(2)}(r) = \frac{1}{4\pi}\int \rho_i(r\hat{\mathbf{r}})d\hat{\mathbf{r}}, \tag{4}$$

which measures pairwise distances from the atom i within R_{cut} as schematically shown in Fig. 2(a). Here, $\hat{\mathbf{r}}$ denotes the unit vector of $\mathbf{r}$. This function, however, cannot accurately describe the potential energy surface because of the lack of angular information. Specifically two different probability densities ρ_i can yield an identical $\rho_i^{(2)}$, which would then yield the same local energy U_i. The necessary angular information can be incorporated by using the probability to find an atom j at a distance r from the ith atom and another atom k at a distance s from the ith atom along the angle $\angle kij = \theta$ as schematically shown in Fig. 2(b). Starting from $\rho_i(\mathbf{r})$, this probability can be determined as

$$\rho_i^{(3)}(r,s,\theta) = \iint \delta(\hat{\mathbf{r}}\cdot\hat{\mathbf{s}} - \cos\theta)\rho_i(r\hat{\mathbf{r}})\rho_i^{*}(s\hat{\mathbf{s}})d\hat{\mathbf{r}}d\hat{\mathbf{s}}. \tag{5}$$

This function, commonly referred to as angular distribution function, is equivalent to the power spectrum used in practical applications of the GAP [16,28–30]. In order to show this equivalence, ρ_i is expanded as

$$\rho_i(\mathbf{r}) = \sum_{l=1}^{L_{\max}}\sum_{m=-l}^{l}\sum_{n=1}^{N_R^l} c_{nlm}^i \chi_{nl}(r) Y_{lm}(\hat{\mathbf{r}}). \tag{6}$$

Here, $\{\chi_{nl}|n = 1,\ldots,N_R^l, l = 0,\ldots,L_{\max}\}$ denote radial basis functions that satisfy the following orthonormal relation

$$4\pi\int_0^\infty \chi_{nl}(r)\chi_{n'l}(r)r^2dr = \delta(n-n'). \tag{7}$$

Y_{lm} are the spherical harmonics. By using Eq. (6), Eqs. (4) and (5) can be rewritten as

$$\rho_i^{(2)}(r) = \frac{1}{\sqrt{4\pi}}\sum_{n=1}^{N_R^0} c_n^i \chi_{nl}(r), \tag{8}$$

$$c_n^i = c_{n00}^i, \tag{9}$$

$$\rho_i^{(3)}(r,s,\theta) = \sum_{l=1}^{L_{\max}}\sum_{n=1}^{N_R^l}\sum_{\nu=1}^{N_R^l}\sqrt{\frac{2l+1}{2}} \times p_{n\nu l}^i \chi_{nl}(r)\chi_{\nu l}(s)P_l(\cos\theta), \tag{10}$$

$$p_{n\nu l}^i = \sqrt{\frac{8\pi^2}{2l+1}}\sum_{m=-l}^{l} c_{nlm}^i c_{\nu lm}^{i*}, \tag{11}$$

where P_l is a Legendre polynomial of order l. Equation (11) is the same as the equation for the power spectrum described in Refs. [27,31]. Equations (10) and (11) indicate that $p_{n\nu l}^i$ corresponds to the expansion coefficients of $\rho_i^{(3)}$ with respect to the orthonormal radial and angular basis functions. Thus $p_{n\nu l}^i$ contains the same information as the angular distribution defined in Eq. (5).

C. Potential energy and Gaussian approximation potential

In our ML algorithm, we use the distributions $\rho_i^{(2)}$ and $\rho_i^{(3)}$ to parametrize the potential energy surface U. This means that U_i is described as a functional of $\rho_i^{(2)}$ and $\rho_i^{(3)}$,

$$U_i = F\big[\rho_i^{(2)}, \rho_i^{(3)}\big]. \tag{12}$$

Obviously, it is not generally a simple matter to find a suitable functional form, although neural networks and the moment tensor potentials (MTP) have been found to yield an excellent approximation for total energies [11,32,33]. In the present work, we adopt the Gaussian approximation potential as pioneered by Bartók and coworkers [12]. In this approach, a set of N_B local reference structures $\{\rho_{i_B}|i_B = 1,\ldots,N_B\}$ are chosen. These reference configurations are converted to a set of coefficients in the descriptor space $\{\mathbf{X}_{i_B}|i_B = 1,\ldots,N_B\}$ and the potential energy is approximated by fitting a set of coefficients $\{w_{i_B}|i_B = 1,\ldots,N_B\}$:

$$F\big[\rho_i^{(2)}, \rho_i^{(3)}\big] = \sum_{i_B=1}^{N_B} w_{i_B} K(\mathbf{X}_i, \mathbf{X}_{i_B}). \tag{13}$$

Here each vector $\mathbf{X}_i$ collects all coefficients c_n^i and $p_{n\nu l}^i$ for a specific local configuration $\rho_i(\mathbf{r})$ [Eqs. (9) and (11)]. The function K is supposed to measure the similarity between a local configuration of interest $\rho_i(\mathbf{r})$ and the reference configurations $\rho_{i_B}(\mathbf{r})$. It usually approaches unity if two configurations are similar and decays towards a small value if the two configurations are different.

In the present case, the following polynomial function is used:

$$K(\mathbf{X}_i, \mathbf{X}_{i_B}) = \beta^{(2)}\big(\mathbf{X}_i^{(2)}\cdot\mathbf{X}_{i_B}^{(2)}\big) + \beta^{(3)}\big(\hat{\mathbf{X}}_i^{(3)}\cdot\hat{\mathbf{X}}_{i_B}^{(3)}\big)^{\zeta^{(3)}}. \tag{14}$$

Here, $\mathbf{X}_i^{(2)}$ and $\mathbf{X}_i^{(3)}$ are the vectors containing c_n^i and $p_{n\nu l}^i$, respectively. The vector $\hat{\mathbf{X}}_i^{(3)}$ denotes a normalized vector of $\mathbf{X}_i^{(3)}$, $\beta^{(2)}$ and $\beta^{(3)}$ are weighting parameters, and $\zeta^{(3)}$ is a parameter to control the sharpness of the function K. Phenomenologically, the first term in Eq. (14) can be regarded as a pairwise linear interaction term, which is suited to describe long-range radial interactions, such as Coulomb and Lennard-Jones interactions. In contrast, the second term provides nonlinear many-body interaction terms as discussed by Glielmo *et al.* [34]. The latter term is known as the SOAP [27]. The name SOAP relates to the fact that the dot product $\mathbf{X}_i^{(3)} \cdot \mathbf{X}_{i_B}^{(3)}$ can be related to the Haar integral [35] of the square of an overlap between two probability distributions as

$$\mathbf{X}_i^{(3)} \cdot \mathbf{X}_{i_B}^{(3)} = \int \left|S_{ii_B}(\hat{R})\right|^2 d\hat{R}, \tag{15}$$

$$S_{ii_B} = \int \rho_i(\mathbf{r})\rho_{i_B}(\hat{R}\mathbf{r})d\mathbf{r}, \tag{16}$$

where $\hat{R}$ denotes the rotational operator, and the integral in Eq. (15) needs to be performed over all possible rotations defined by this operator. In this sense, SOAP measures the structural similarity between the structure i and the reference structure i_B as an overlap of two probability distributions. Similarly, the dot product $\mathbf{X}_i^{(2)} \cdot \mathbf{X}_{i_B}^{(2)}$ can be rewritten as an overlap integral between the radial distribution functions as

$$\mathbf{X}_i^{(2)} \cdot \mathbf{X}_{i_B}^{(2)} = 4\pi \int_0^{R_{\text{cut}}} \rho_i^{(2)}(r)\rho_{i_B}^{(2)}(r)r^2 dr. \tag{17}$$

In this study, in order to examine the efficiency of the on-the-fly scheme in a manner comparable to previous publications using the SOAP scheme, we only use the SOAP $K(\mathbf{X}_i, \mathbf{X}_{i_B})$ by setting $\beta^{(2)} = 0$ for all materials. An application including the radial descriptor is presented elsewhere [24].

Before ending this section, we briefly explain some key points where our implementation differs from the previous SOAP implementations. At every MD step, the expansion coefficients c_{nlm}^i and their partial derivatives with respect to the atomic positions need to be calculated in order to compute the potential energy U and its partial derivatives. In the SOAP [27], c_{nlm}^i can be analytically formulated using the Gaussian function of Eq. (3) as

$$c_{nlm}^i = \sum_{j=1}^{N_a} h_{nl}(r_{ij})Y_{lm}^*(\hat{\mathbf{r}}_{ij}), \tag{18}$$

$$h_{nl}(r) = \frac{4\pi}{\left(\sqrt{2\sigma_{\text{atom}}^2\pi}\right)^3} f_{\text{cut}}(r) \int_0^\infty \chi_{nl}(r') \times \exp\left(-\frac{r'^2 + r^2}{2\sigma_{\text{atom}}^2}\right)\iota_l\left(\frac{rr'}{\sigma_{\text{atom}}^2}\right)r'^2 dr', \tag{19}$$

where ι_l denotes the modified spherical Bessel function of the first kind. However, the calculation of h_{nl} and dh_{nl}/dr by Eq. (19) would be computationally rather demanding. In order to accelerate the calculations, we adopt a spline interpolation. In this method, $h_{nl}(r)$ is calculated on a radial mesh over r once at the beginning of the training or MD simulation. The calculated function is spline-interpolated, and the interpolated function is used to calculate the coefficients c_{nlm}^i and their derivatives later. This method allows us to avoid evaluating the integral in Eq. (19) at each MD step and accelerates the computation of the expansion coefficients c_{nlm}^i and their derivatives by roughly a factor of ten, and accelerates the entire calculation of the function K by a factor of two. Another difference to previous implementations is in the choice of the radial basis functions. In our method, normalized spherical Bessel functions $\chi_{nl} = j_l(q_{nl}r)$ are used as the radial basis functions, because their mutual orthogonality allows for a systematic improvement by simply increasing the number N_R^l of the radial functions as described in Appendix A. Finally, we use the cutoff function proposed by Behler and Parrinello [11] defined as

$$f_{\text{cut}}(r_{ij}) = \begin{cases} \frac{1}{2}\left[\cos\left(\pi\frac{r_{ij}}{R_{\text{cut}}}\right) + 1\right] & \text{if } r_{ij} \leqslant R_{\text{cut}} \\ 0, & \text{otherwise} \end{cases}. \tag{20}$$

This cutoff function weights atoms close to the central atom i more strongly. This radially scaled weight enables the descriptor to efficiently describe the structural differences that strongly influence the potential energy of atom i similarly to the radially scaled kernel proposed by Willat and co-workers [36]. In Table S1 in Ref. [37], all parameters described in Secs. II B and II C are tabulated.

D. Fitting of energy, forces, and stress tensor and their uncertainty

In order to determine the fitting parameters w_{i_B}, the energies, forces and stress tensor for a set of reference structural datasets labeled by a superscript $\alpha = 1, \ldots, N_{st}$ must be fitted. Here, N_{st} denotes the number of the reference structure datasets. Combining Eqs. (1), (12), and (13) yields for the total energy per atom the following equation that must be fulfilled in a least square sense:

$$\frac{U^\alpha}{N_a^\alpha} \overset{!}{=} \sum_{i=1}^{N_a^\alpha} \frac{U_i^\alpha}{N_a^\alpha} = \sum_{i_B=1}^{N_B} w_{i_B} \sum_{i=1}^{N_a^\alpha} \frac{K\left(\mathbf{X}_i^\alpha, \mathbf{X}_{i_B}\right)}{N_a^\alpha} \quad \forall\alpha = 1, \ldots, N_{st}. \tag{21}$$

Here, U_i^α is the local energy of atom i in the structure α, and $\mathbf{X}_i^\alpha$ is the vector of coefficients in the descriptor space for atom i in structure α, and U^α is the actual FP energy. In practice, we simultaneously fit the energy per atom, forces and the stress tensor for reference structures α, for which FP calculations have yet been performed (see below for details). The previous equation indicates that the total potential energy U^α is a linear function of the coefficients w_{i_B}. It is also straightforward to see that the forces and the stress tensor components are described as linear functions of the coefficients w_{i_B}. These linear equations can be collected into a matrix-vector form as

$$\mathbf{y}^\alpha \overset{!}{=} \boldsymbol{\phi}^\alpha \mathbf{w} \quad \forall\alpha. \tag{22}$$

Here, $\{\mathbf{y}^\alpha | \alpha = 1, \ldots, N_{st}\}$ denotes column vectors containing the dimensionless FP potential energy per atom in the first line, the forces and the components of the stress tensor in the subsequent lines for a single structure α in the reference structure dataset, in total

$$m^\alpha = 1 + 3N_a^\alpha + 6 \tag{23}$$

components for N_a^α atoms. The entries are made dimensionless by dividing their values by the standard deviations of the FP energies per atom, forces and stress tensors in the reference structure datasets. The column vector $\mathbf{w}$ is comprised of the N_B coefficients w_{i_B}, and $\boldsymbol{\phi}^\alpha$ is a $m^\alpha \times N_B$ matrix. The first line of the matrix is made up by $\sum_i K(\mathbf{X}_i^\alpha, \mathbf{X}_{i_B})/N_a^\alpha$, the second line is made up by the derivative of the energy with respect to the first atomic coordinate in the structure α and so on.

After the fitting, the energies and forces of a new structure with the descriptor $\mathbf{X}_i$ can be efficiently obtained calculating

$$\mathbf{y} = \boldsymbol{\phi}\mathbf{w}, \tag{24}$$

where $\boldsymbol{\phi}$ comprises $\sum_i K(\mathbf{X}_i, \mathbf{X}_{i_B})$ in the first row, and the partial derivatives of the function K with respect to the coordinates in the structure in the subsequent rows.

In the conventional schemes, FP calculations are carried out on a wide variety of structures in advance, local configurations are chosen from each structure in the dataset, and the coefficients $\mathbf{w}$ are optimized to optimally reproduce the reference structure datasets. In our scheme, the FP data generation, selection, and parameter optimization are carried out on the fly during the MD simulations. A key component that makes this algorithm extremely efficient is the evaluation of the uncertainty in the prediction. This is used to decide whether the FP calculations are necessary or not at step 2. In our scheme, both the optimization of $\mathbf{w}$ and the uncertainties are estimated by a Bayesian linear-regression method [38].

The Bayesian linear regression assumes the presence of a set of the coefficients that exactly reproduce the exact energy, forces and stress tensor without any numerical noises. Then, this method determines the probability to find this exact set of coefficients at $\mathbf{w}$ on the basis of the observation of a limited number of FP datasets $\{\mathbf{y}^\alpha | \alpha = 1, \dots, N_{st}\}$ containing numerical noises. In practice, the FP data usually carry comparatively small errors, however, the finite cutoff R_{cut} implies that the model can never exactly describe the FP data. In other words, as the atoms outside the cutoff radius move, the local energies and forces should change, but since the model assumes that the local energy depends only on the position of the atoms inside the cutoff sphere, residual errors are introduced as shown by Ref. [29]. We assume that these errors can be modeled by the presence of noise with a Gaussian distribution in the FP data. This assumption also implies some errors in the model parameters $\mathbf{w}$. The so-called posterior distribution $p(\mathbf{w}|\mathbf{Y})$ is determined as a Gaussian distribution written as (see necessary assumptions in Appendix B):

$$p(\mathbf{w}|\mathbf{Y}) = \mathcal{N}(\bar{\mathbf{w}}, \boldsymbol{\Sigma}), \tag{25}$$

$$\bar{\mathbf{w}} = \frac{1}{\sigma_v^2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{Y}, \tag{26}$$

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma_w^2} \mathbf{I} + \frac{1}{\sigma_v^2} \boldsymbol{\Phi}^T \boldsymbol{\Phi}. \tag{27}$$

Here, $\mathbf{Y}$ is a supervector with size $M = \sum_\alpha m^\alpha$ [compare Eq. (23)] collecting *all* FP energies per atom, forces, and stress tensors $\{\mathbf{y}^\alpha | \alpha = 1, \dots, N_{st}\}$ in the reference structure datasets. Similarly, the $M \times N_B$ design matrix $\boldsymbol{\Phi}$ is a collection of all matrices $\boldsymbol{\phi}^\alpha$ on all reference structure datasets, and $\mathbf{I}$ denotes the unit matrix. The symbols σ_v^2 and σ_w^2 denote parameters optimized to balance the accuracy and robustness of the evolving force field as explained later on. The symbol $\mathcal{N}(\bar{\mathbf{w}}, \boldsymbol{\Sigma})$ is a multidimensional normalized Gaussian centered at $\bar{\mathbf{w}}$ and defined as

$$\mathcal{N}(\bar{\mathbf{w}}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^{N_B} ||\boldsymbol{\Sigma}||}} \times \exp\left[-\frac{(\mathbf{w} - \bar{\mathbf{w}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \bar{\mathbf{w}})}{2} \right], \tag{28}$$

where $||\boldsymbol{\Sigma}||$ means the determinant of the matrix $\boldsymbol{\Sigma}$. The desired optimal coefficients are determined at the center of the Gaussian distribution $\mathbf{w} = \bar{\mathbf{w}}$, where the posterior probability is maximized. It is straightforward to show that the vector $\bar{\mathbf{w}}$ is identical to the vector obtained by the ridge regression, with the ratio of σ_v^2 to σ_w^2 being equivalent to the Tikhonov regularization parameter.

The uncertainty in the predictions is provided from the probability to find the exact FP energy per atom, forces and stress tensor at $\mathbf{y}$. This posterior distribution $p(\mathbf{y}|\mathbf{Y})$ is obtained from the posterior distribution $p(\mathbf{w}|\mathbf{Y})$ as (see also Appendix B)

$$p(\mathbf{y}|\mathbf{Y}) = \mathcal{N}(\boldsymbol{\phi}\bar{\mathbf{w}}, \boldsymbol{\sigma}), \tag{29}$$

$$\boldsymbol{\sigma} = \sigma_v^2 \mathbf{I} + \boldsymbol{\phi}^T \boldsymbol{\Sigma} \boldsymbol{\phi}. \tag{30}$$

The mean vector $\boldsymbol{\phi}\bar{\mathbf{w}}$ contains the results of the predictions on the dimensionless energy per atom, forces and stress tensor. The diagonal elements of the covariance matrix $\boldsymbol{\sigma}$ provide the variance of the marginal distribution of each dimensionless component. Note that two terms are present, where the first term usually dominates and quickly grows towards the mean difference between predicted and actual first-principles data. Since the term varies very slowly over time, it is not suitable to detect configurations that have not yet been visited and learned. However, the second term in Eq. (30), which corresponds to the uncertainty in the regression coefficient $\mathbf{w}$, can be used to determine configurations where additional training data are required, as discussed below. To estimate the error that the machine most likely makes, the square root of this dimensionless variance is multiplied with the standard deviation of the training data, which was used to generate the dimensionless training data $\mathbf{Y}$.

As in the ridge regression, the optimization of the parameters σ_v^2 and σ_w^2 is important to prevent overfitting. In our algorithm, they are optimized by the evidence approximation [39–41]. In this scheme, the parameters σ_v^2 and σ_w^2 are determined by maximizing the marginal likelihood function called evidence function. This evidence function corresponds to a probability that the regression model with specific parameters σ_v^2 and σ_w^2 provides the reference data $\mathbf{Y}$, and it is calculated as

$$p(\mathbf{Y}|\sigma_v^2, \sigma_w^2) = \left(\frac{1}{\sqrt{2\pi\sigma_v^2}} \right)^M \left(\frac{1}{\sqrt{2\pi\sigma_w^2}} \right)^{N_B} \times \int \exp[-E(\mathbf{w})] d\mathbf{w}, \tag{31}$$

$$E(\mathbf{w}) = \frac{1}{2\sigma_v^2} ||\boldsymbol{\Phi}\mathbf{w} - \mathbf{Y}||^2 + \frac{1}{2\sigma_w^2} ||\mathbf{w}||^2. \tag{32}$$

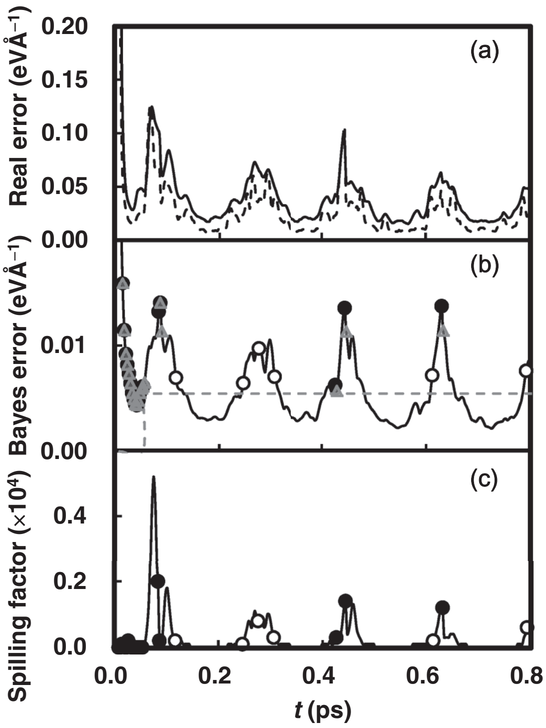

FIG. 3. Errors during a finite temperature simulation of rocksalt MgO containing 64 atoms at 300 K. (a) Actual difference between the FP forces and the forces calculated by the MLFF. Solid line shows the root mean square difference averaged over all atoms, and the dashed line shows a quarter of maximum difference on a single atom. (b) The maximum Bayesian error [the second term of Eq. (30)] of the force on a single atom. (c) The maximum spilling factor on a single atom. Open circles and black small dots indicate steps 3 and 4, respectively. Dark gray triangles indicate the Bayesian errors stored as $\sigma_{\max,I}$. These are used to determine the threshold for the Bayesian error $\epsilon_{\rm BE}$ as explained in Sec. II E. Gray dashed lines indicate the threshold for the Bayesian error.

Hence, the optimization over σ_v^2 and σ_w^2 can be regarded as the maximization of the probability to provide the correct answer $\mathbf{Y}$ at any regression coefficient $\mathbf{w}$. Details of the maximization method are documented in Appendix C.

In addition to the Bayesian error, we also calculate the spilling factor suggested by Miwa and Ohno [42]. The spilling factor is a measure of the density of local reference configurations i_B near the new local configuration $\mathbf{X}$ in the descriptor space, and its equation is formulated as

$$s = 1 - \frac{\sum_{i_B=1}^{N_B}\sum_{i'_B=1}^{N_B} K\big(\mathbf{X},\mathbf{X}_{i_B}\big)K^{-1}\big(\mathbf{X}_{i_B},\mathbf{X}_{i'_B}\big)K\big(\mathbf{X}_{i'_B},\mathbf{X}\big)}{K(\mathbf{X},\mathbf{X})}. \tag{33}$$

The equation was slightly modified from its original equation in order to make it applicable to a non-normalized similarity measure such as the one used by us, see Eq. (14). If the density of the reference configuration is great enough to provide complete overlap among the configurations, s approaches zero, otherwise s approaches unity.

Trends of these two estimated errors are illustrated in Fig. 3, where the real errors, Bayesian errors and spilling factors are shown as a function of the MD simulation time during on-the-fly force field generations for the MgO solid at 300 K. Note again that we use only the second term in Eq. (30) to determine regions where additional first-principles data are required. As already mentioned, the first term quickly grows towards the total average error, but varies only slowly in time and is fairly insensitive to the configurations. The second term alone is unfortunately one order of magnitude smaller than the real error for the forces, but it turns out to follow the real errors rather precisely. Ideally, the total predicted error (sum of both terms) should closely follow the real error, but that is not the case. The cause for this issue is not yet entirely clear. However, we speculate that the regression may not be accurate enough to quantitatively predict the uncertainty (i) because of difficulties in the simultaneous description of the prior distribution of the dimensionless energy, forces and stress tensor components by a single Gaussian function written as Eq. (B2), or (ii) because the evidence approximation fails to predict accurate parameters for σ_v^2 and σ_w^2 individually. Concerning (i), the error in the second term more accurately resembles the real error in the forces, if the training weights forces more strongly, as shown in Appendix D. However, weighting the forces more strongly does not entirely resolve the issue. Concerning (ii) we note again that the first term very quickly grows towards the average real error, so it essentially accommodates all the uncertainties, in turn reducing the second term. In practice the second term, although too small in magnitude, is able to robustly determine under-represented regions of the configuration space, which suffices for our purpose. So in practice the underestimation of the real error has little consequences, since we determine the thresholds automatically as described below.

Concerning the spilling factor we note the following issues. Although, the spilling factor also resembles the shape of the real error, it fails to reproduce the large uncertainties at the beginning of the training simulation. Furthermore, its absolute value is very system and materials dependent, e.g., 10^{-6}–10^{-5} for MgO solid and 10^{-3}–10^{-2} for MgO liquid. This strong dependence causes difficulties to set a criterion for the spilling factor, in particular, for simulation where we train on the liquid and solid. Furthermore, we found numerical instability in its computation, since the matrix K is not regularized and is potentially nonpositive definite unlike the covariance matrix $\mathbf{\Sigma}$ for the Bayesian error.

Details of the decision scheme at step 2 are explained in the following section. Here we briefly note that the criterion for the spilling factor is set to a relatively large value of 0.02 as in Ref. [38]. For the applications reported here, the threshold for the spilling factor is hardly ever passed. Only for liquid and interfacial MgO, the calculated spilling factors exceeded this criterion. However, even then only 2% of the total FP calculations are executed because of the spilling factor criterion. So in the present work, the Bayesian error criterion is more relevant.

During the training MD simulations, the total energy is not conserved when FP calculations are performed or when the force field is retrained. However, the discontinuous potential energy changes are small, and we did not observe any numerical instability in our training MD simulations, at least when using a Langevin thermostat [43] with a friction coefficient of

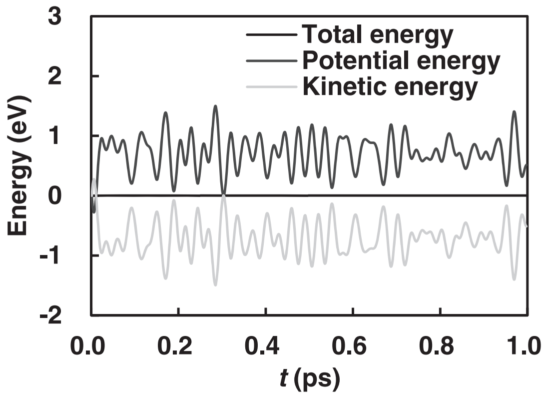

FIG. 4. Total (black), kinetic (light gray), and potential (dark gray) energies of solid MgO containing 216 atoms at 300 K during the first 1 ps MD simulation with the microcanonical ensemble using the generated force field. The energies were shifted to provide zero at the beginning.

$10\ \mathrm{ps}^{-1}$. When only the generated force field is used without any retraining, the total energy is conserved as demonstrated in Fig. 4, because the forces and stress tensor components are formulated consistently as the partial derivatives of the potential energy.

Before ending this section, we note the similarities and differences between our approach and the GAP [12]. Both methods adopt the Gaussian process and assume that the error in the data **Y** follows a Gaussian distribution. Differences are in the regression method. In our work, we use Bayesian linear regression to solve Eq. (13), in combination with a nonlinear map describing the similarity between two configurations with descriptors $\mathbf{X}_i$ and $\mathbf{X}_j$ [see Eq. (14)]. Contrary to this, GAP adopts the kernel ridge regression. Kernel ridge regression transforms the linear ridge regression to the kernel ridge regression [44] using a nonlinear kernel that closely resembles our map. In kernel ridge regression, the final equation for the optimal regression coefficients is given by the equation (for simplicity, we disregard derivative information, e.g., forces here)

$$\bar{\mathbf{w}} = \left(\mathbf{K}^T\mathbf{K} + \sigma_v^2/\sigma_w^2\mathbf{K}\right)^{-1}\mathbf{K}^T\mathbf{Y} = \left(\mathbf{K} + \sigma_v^2/\sigma_w^2\mathbf{I}\right)^{-1}\mathbf{Y}, \tag{34}$$

whereas in our case, the solution can be written as [substituting Eq. (27) into Eq. (26)]

$$\bar{\mathbf{w}} = \left(\mathbf{K}^T\mathbf{K} + \sigma_v^2/\sigma_w^2\mathbf{I}\right)^{-1}\mathbf{K}^T\mathbf{Y}. \tag{35}$$

In the limit of no regularization, these two equations are equivalent, however, they are clearly distinct after regularization (**I** or **K**). Bayesian regression essentially regularizes the squared problem ($\mathbf{K}^T\mathbf{Y} \overset{!}{=} \mathbf{K}^T\mathbf{K}\mathbf{w}$), whereas in kernel ridge regression the original problem ($\mathbf{Y} \overset{!}{=} \mathbf{K}\mathbf{w}$) is regularized. A deeper analysis of the advantages and disadvantages of both approaches is beyond the scope of the present work, although we note that the inclusion of forces and stress tensor components (multivariate regression) seems simpler and more efficient using Bayesian regression. Otherwise, we expect both approaches to posses roughly similar performance.

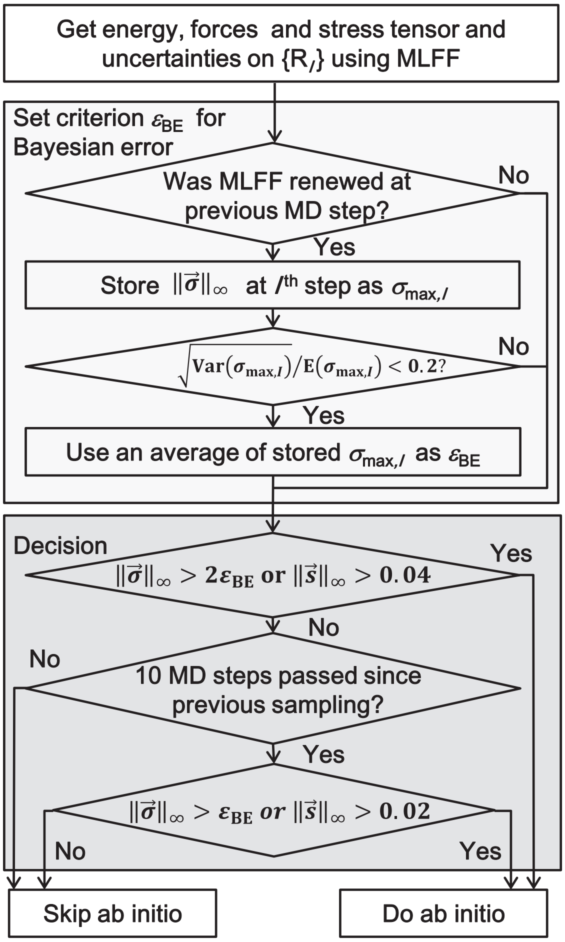

FIG. 5. Flowchart of the decision step whether to perform FP simulations or not. The symbols $\vec{\sigma}$ and $\vec{s}$ denote the vectors containing the Bayesian errors in the forces and spilling factors, respectively, for all atoms. $||\vec{x}||_\infty$ denotes the infinity norm (also called supremum norm) of the vector $\vec{x}$, ϵ_{BE} denotes the criterion for the Bayesian error, and $\mathrm{Var}(x)$ and $\mathrm{E}(x)$ refer to the variance and the average of the data x. At the beginning of every training simulation, the criterion ϵ_{BE} is set to zero.

E. Decision to perform FP calculation

The decision whether to perform a FP calculation in step 2 or not (see Sec. II A) is obviously an important one. Figure 5 shows the flowchart of our decision scheme. As shown in the dark gray square in this figure, the decision is done on the basis of the estimated errors and the history of the previous samplings. First, the machine examines the estimated errors. If the maximum Bayesian error in one of the forces or the spilling factor is larger than twice the chosen threshold described later, the machine performs FP calculations. This avoids instabilities in the MD simulation caused by very inaccurate forces. Next, the machine checks the previous data sampling step. If the current step is within 10 MD steps from the previous sampling step, the machine skips the FP calculations. This process avoids too dense sampling within a narrow phase space. If more than 10 MD steps have passed since the previous sampling, the machine examines the estimated errors. If the maximum Bayesian error in one of the forces or the spilling factor is larger than the chosen threshold, the machine performs the FP calculation, otherwise the FP calculation is skipped.

The threshold for the spilling factor is set to 0.02 following Ref. [42]. For the Bayesian error, which exhibits a descriptor- and materials-dependent nonzero value, the threshold ϵ_{BE} is automatically determined on the fly. The corresponding scheme is shown in the light gray square in Fig. 5. At the MD step I just after the refinement of the force field (shown as the gray triangles in Fig. 3), the machine stores the maximum value $\sigma_{\max,I}$ of the Bayesian errors of the forces predicted for the new structure I. Because this new structure does not significantly differ from the structure sampled at the training step, the calculated Bayesian errors are nearly identical to the Bayesian errors on the previously sampled structure. Hence, this maximum Bayesian error $\sigma_{\max,I}$ can be regarded as a measure of the lowest currently attainable Bayesian error and can provide a reasonable threshold for the future. In our algorithm, the threshold is updated to be the average of the last 10 $\sigma_{\max,I}$, if their relative standard deviation is smaller than 0.2 (empirically set). The dashed gray lines in Fig. 3(b) illustrate the criteria determined by this on-the-fly scheme during the training on the MgO solid as an example.

F. Sparsification and data selection

As previously explained, whenever the machine decides that for a specific structure insufficient information is stored in the machine, FP calculations are performed for that structure. To reduce the computational demands, the machine is not retrained after each FP calculation, but instead retraining is done typically after $n = 5$ FP calculations or when the estimated errors are twice larger than the determined criteria. This allows to block many of the computationally expensive steps in the training. When n FP calculations have been performed, the reference structure datasets and local reference configurations are selected, and the force field is refined. Figure 6 shows the flowchart corresponding to this data selection. As shown in the dark gray square, the data selection is done in a two step procedure. First, the machine selects those local configurations that exhibit Bayesian errors on forces and spilling factors that are larger than the threshold. Although this step is already a sparsification process, numerical instabilities sometimes occur because of the overcompleteness among the remaining local configurations. In order to avoid those numerical instabilities, another sparsification process is performed using a CUR algorithm [29,45]. In our implementation, the machine examines correlations between the local configurations and eigenvalues of the matrix K smaller than 10^{-10} using leverage scoring ω_i. For details we refer to Appendix E. Starting from the largest ω_i in descending order, N_{low} configurations are discarded, where N_{low} denotes the number of the eigenvalues smaller than 10^{-10}. Finally, as shown in the light gray square in Fig. 6, the machine discards those structure datasets that do not provide any local reference configurations to speed up the computations and to reduce the memory usage.

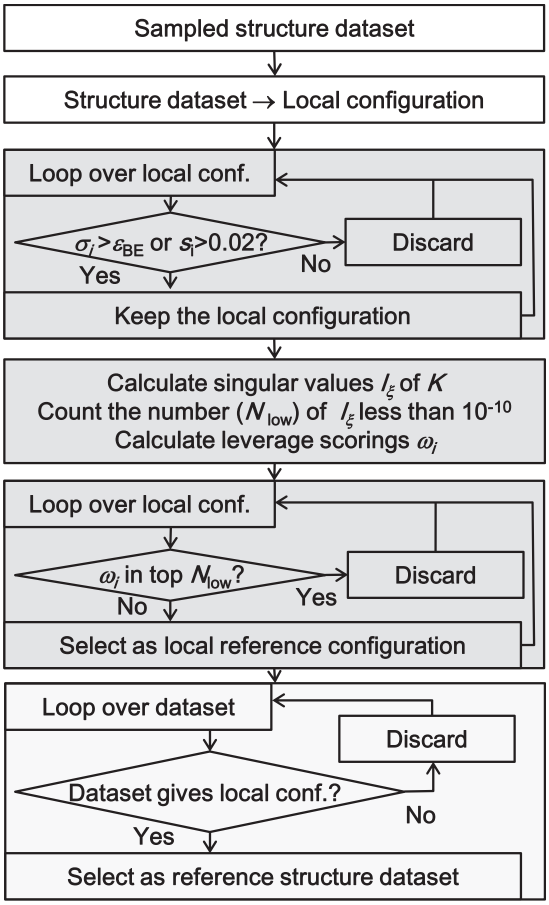

FIG. 6. Flowchart of the sparsification and data selection step. The symbol K denotes the matrix comprised of the elements $K(i, j)$ defined as Eq. (14). The leverage scoring ω_i is calculated by Eq. (E5) in Appendix E.

III. TECHNICAL DETAILS OF MELTING POINT CALCULATIONS

A. Training conditions

For each material, the local reference configurations and the datasets were collected during MD simulations on solid, liquid, and interface systems, and all of those collected data were used to optimize the regression coefficients. The chosen descriptors can fully distinguish the local environment for these three cases and allow for the simultaneous reproduction of the potential energies for all configurations. The Al solid and liquid were modeled by unit cells with 108 atoms, and the Al interface was modeled by a unit cell with 144 atoms. For the other materials, solids and liquids were modeled by unit cells with 64 atoms, and the interfaces were modeled by unit cells with 128 atoms. A $3\times3\times3$ **k**-point mesh was used for the Al solid and liquid, and a $3\times3\times2$ **k**-point mesh was used for the Al interface. For the other materials, a $2\times2\times2$ mesh was used for the solids and liquids, and a $2\times2\times1$ mesh was used for the interfaces. Plane-wave basis sets and the projector augmented wave (PAW) method were used in all FP calculations. The PAW atomic reference configuration was $2s^22p^4$ for O, $3s^23p^0$ for Mg, $3s^23p^1$ for Al, $3s^23p^2$ for Si, $4s^24p^2$ for Ge, and $5s^25p^2$ for Sn. The plane-wave cutoff energy was set to

325, 325, 225, 135, and 520 eV for Al, Si, Ge, Sn, and MgO, respectively. Training was performed for several functionals: the local density approximation (LDA) in the parametrization of Ceperly and Alder [46], the Perdew-Burke-Ernzerhof (PBE) functional [47], its variant for solids (PBEsol) [48] and the strongly constrained appropriately normed (SCAN) functional [49]. For each material, the local reference configurations and the datasets were collected during the MD simulations on solid, liquid and interfacial systems. For each condition and all phases, the MD simulation was executed for 100 ps. We opted for this simulation time after observing that the machine did not execute any FP calculations during the last 10 ps of the training simulations. We also confirmed that both, the Bayesian error and the spilling factor during the production runs, were comparable to the values calculated in the training simulations. The MD time step was set to 3 fs for all materials except for Ge and Sn, where the time step was set to 10 fs. Further details of the MD parameters and snapshots of the models are shown in Sec. S2 in Ref. [37].

B. Efficiency and accuracy

During the on-the-fly force field generation, more than 99% of the FP calculations were skipped reducing the computational time by a factor of more than 200. These accelerations allow the machine to efficiently collect reference configurations in a wide phase space. The details of the skipping ratio and the acceleration are summarized in Table S3 in Ref. [37].

The number of structures in the reference structure datasets is typically less than 500, and the number of local reference configurations is less than 1000 as shown in Table S4 in Ref. [37]. Both are much smaller than the reference configurations used in previous studies [11,15–19]. It should be mentioned though that we train our force fields essentially to the specific application, here liquids, solids, and interfaces.

By the efficient sampling of the local reference configurations, MD simulations by the force fields are noticeably accelerated. Table S5 in Ref. [37] tabulates the elapsed time per MD step by the force fields and FP calculations. The force fields accelerate the MD simulations by factors of 2000 to 5000. The factor will further grow with increasing system size because the computational cost of the MLFF calculation scales only linearly with the system size.

In addition to the significant acceleration of the computational speed, the adaptable reference datasets and flexible functional form allows for accurate predictions of the potential energy surfaces. The error analysis summarized in Fig. 7 indicates that the mean absolute errors (in parentheses root mean square errors) in energies, forces and stress tensors are 5.5 (6.2) meV atom^{-1}, 0.07 (0.09) eV Å^{-1}, and 0.18 (0.27) GPa, on average, respectively.

C. Interface pinning

The melting-point calculations are carried out using force fields and using the interface pinning method [10]. This method has shown to be able to accurately predict the melting temperature and pressure [10]. In the interface pinning method, an MD simulation with constant temperature and pressure [58,59] is carried out on a solid-liquid interface. During the MD simulation, a harmonic bias potential is added to the potential energy U in order to constrain the order parameter Q of the system to an intermediate order parameter a between the solid and the liquid phases as

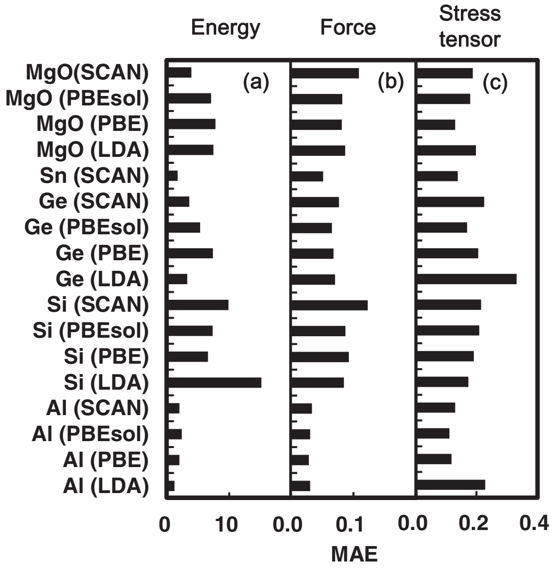

FIG. 7. Mean absolute errors in the energy per atom (meV atom^{-1}) (a), force (eV Å^{-1}) and stress tensor (GPa) predicted by the force fields on 200 configurations of solids and liquids at the melting points.

$$U' = U + \frac{\kappa}{2}(Q - a)^2, \tag{36}$$

where κ is a force constant. From the mean force that keeps the order parameter close to a, the difference in the chemical potential $\Delta\mu$ between two phases is calculated as

$$\Delta\mu = -\kappa(\langle Q\rangle' - a)\frac{\Delta Q}{N_a}, \tag{37}$$

where $\langle Q\rangle'$ is the order parameter of the interfacial system averaged over the biased MD simulation. ΔQ is the difference in the order parameter between the solid and liquid. The melting temperature is determined as the point where $\Delta\mu$ becomes zero by using Newton's root finding method. As order parameters, the collective density proposed in Ref. [10] is adopted. Further details of the used parameters and interfacial systems are summarized in Sec. S4 in Ref. [37].

Once the melting temperature has been calculated, one can obtain the entropy of fusion S_{ls} from the difference in the enthalpy between two phases at the melting temperature. Furthermore, the slope of the melting curve dT_m/dp can be calculated as S_{ls}/V_{ls} by the Clausius-Clapeyron relation, where V_{ls} denotes the volume difference between the liquid and solid. These thermodynamic properties were also evaluated and compared with previously reported values as well as experimental results.

Although the generated force fields are very precise, they necessarily deviate from the FP data as shown in Fig. 7. The effects of these errors on the melting points were also evaluated by thermodynamic perturbation theory. Details of

TABLE I. Melting temperatures (K) of Al, Si, Ge, Sn, and MgO. CORR and MLFF denote the results with and without the thermodynamic perturbation corrections, respectively. CORR-3^3 denotes the results for Si, Ge, and Sn with the thermodynamic perturbation using a $3\times3\times3$ **k**-point mesh. Values in the parentheses indicate the uncertainties estimated by the block averaging method described in Refs. [43,50]. Reference calculations with tight error tolerance and similar PAW potential are underlined.

	XC	MLFF	CORR	CORR-3^3	DFT	Exp		XC	MLFF	CORR	DFT	Exp
Si	LDA	1207(5)	1298(5)	1283(5)	1350(100)[a] 1300(50)[e] 1241(20)[f]	1685(2)[b]	Al	LDA	918(7)	909(8)	890(20)[c]	933.47[d]
	PBE	1409(6)	1431(9)	1450(9)	1449(10)[g]			PBE	871(8)	837(9)		
	PBEsol	1145(10)	1172(21)	1213(21)				PBEsol	986(5)	954(6)	985(30)[f]	
	SCAN	1786(7)	1825(7)	1833(7)	1842(10)[g]			SCAN	1017(12)	981(16)		
Ge	LDA	814(7)	814(9)	841(9)		1210.4[b]	MgO	LDA	3165(20)	3243(21)		3040(100)[h]
	PBE	843(5)	876(5)	893(5)				PBE	2652(20)	2698(23)	2747(34)[i]	3250(25)[j]
	PBEsol	758(8)	759(8)	792(8)				PBEsol	2916(19)	2981(20)		
	SCAN	1060(2)	1065(3)	1081(3)				SCAN	3079(23)	3072(25)	3032(54)[i]	
Sn	SCAN	468(11)	459(13)	459(13)		505[k]						

[a] Data from Ref. [1].

[b] Data from Ref. [51].

[c] Data from Ref. [2].

[d] Data from Ref. [52].

[e] Data from Ref. [3].

[f] Data from Ref. [10].

[g] Data from Ref. [53] using the same PAW and 3×3×3× **k** points.

[h] Data from Ref. [54].

[i] Data from Ref. [55] using $2p^63s^2$ PAWs for Mg.

[j] Data from Ref. [56].

[k] Data from Ref. [57].

the thermodynamic perturbation method are described in Ref. [15]. From both, the liquid and solid, trajectories obtained by 100 ps MD simulations with the MLFF at the calculated melting temperatures, 500–1000 structures are selected. FP calculations on the selected structures are performed, and the energy difference between the FP and ML potentials are used in thermodynamic perturbation theory. In these calculations, the same supercells and **k** points as for the training simulations are employed. For Si, Ge, and Sn, we also performed thermodynamic perturbation theory to $3\times3\times3$ **k** points in order to obtain more accurate results and, for Si, to allow for direct comparison with previous literature values [53].

IV. RESULTS: MELTING POINTS

Table I summarizes the melting points T_m of Al, Si, Ge, Sn, and MgO with and without the thermodynamic perturbation corrections. The entropy of fusion, volume change and slopes of the melting curves calculated by the force fields are summarized in Table II. The uncorrected melting temperatures of Al (LDA and PBEsol), Si (LDA, PBE, and SCAN), and MgO (PBE and SCAN) already agree well with the reported DFT results. The differences between the force field and DFT are comparable to those reported by previous studies on Si using carefully optimized GAP [13] and neural network potentials [14]. The thermodynamic perturbation corrections improve the agreement further. We observe that the perturbational corrections (difference between MLFF and CORR in Table I) are more strongly correlated with the errors in the total energy than with the errors in forces and the stress tensor, indicating that accurate predictions of the total energies are essential for the predictions of the melting points. For Si (LDA and PBE) and MgO (PBE and SCAN), the entropies of fusion, volume changes at the phase transition temperature and slopes of the melting curves also agree well with the recently published reference values [53,55]. These results also indicate that our on-the-fly scheme can efficiently generate force fields applicable to quantitative predictions of thermodynamic properties.

Comparison of the theoretical melting points with the experimental results indicates that LDA significantly underestimates the melting point of Si. PBE improves the calculated value, but the melting point is still too low compared to experiment. SCAN, while slightly overestimating the melting point, provides the best agreement with experiment. The worst agreement compared with experiment is obtained by PBEsol, which strongly underestimates the melting point. The trend among PBE, PBEsol, and SCAN for Ge is similar to that observed for Si, with the only exception that now even SCAN underestimates the melting point.

As opposed to Si and Ge, Sn crystallizes in the β-tin structure. For Sn, PBE, and PBEsol do not provide a stable solid within a reasonable temperature range compared to experiment; the melting point seems to be placed at way too low temperatures. Hence, the melting point of Sn has only be determined for SCAN, which still underestimates the melting temperature. As shown in Fig. 8, the observed trend among different functionals and materials can be reasonably well correlated to the energy difference between the α-tin (the cubic diamond structure) and the β-tin structures. A similar correlation was already observed in the melting point

TABLE II. Entropies of fusion, S_{ls} (k_B), volumetric changes, $\Delta V_m = V_l - V_s$ (Å^3 atom^{-1}), or its relative value to the volume of solid, V_s, and slopes of the melting curves dT_m/dp (K GPa^{-1}) for Al, Si, Ge, Sn, and MgO. Errors in our results are within ±0.03 (k_B) for S_{ls}, ±0.02 (Å^3 atom^{-1}) for ΔV_m, ±0.003 for $\Delta V_m/V_s$ and ±3 (K GPa^{-1}) for dT_m/dp. Recent reference calculations with tight error tolerance and similar PAW potential are underlined.

	Property	XC	MLFF	DFT	Exp		Property	XC	MLFF	DFT	Exp
Si	S_{ls}	LDA	3.18	3.0[a], 3.5[b], 3.5[c]	3.3[d]	Al	S_{ls}	LDA	1.31	1.36[e]	1.38[f]
		PBE	3.36	$\underline{3.3}$[g]	3.6[h]			PBE	1.33		
		PBEsol	3.13					PBEsol	1.27		
		SCAN	3.47	$\underline{3.3}$[g]				SCAN	1.25		
	$\Delta V_{ls}/V_s$	LDA	−0.151	−0.10[a], −0.142[b]	−0.119[i]		ΔV_{ls}	LDA	1.20	1.26[e]	1.24[e]
		PBE	−0.128	$\underline{-0.120}$[g]	−0.095[d]			PBE	1.11		
		PBEsol	−0.172					PBEsol	1.25		
		SCAN	−0.089	$\underline{-0.091}$[g]				SCAN	1.11		
	dT_m/dp	LDA	−69	−50[a], −58[b], −51[c]	−38[k]		dT_m/dp	LDA	67		65[j]
		PBE	−57	−55[g]				PBE	68		
		PBEsol	−81					PBEsol	63		
		SCAN	−38	−40[g]				SCAN	64		
Ge	S_{ls}	LDA	3.30		3.7[k]	MgO	S_{ls}	LDA	1.58		
		PBE	3.50					PBE	1.57	$\underline{1.62}$[l]	
		PBEsol	3.38					PBEsol	1.57		
		SCAN	3.53					SCAN	1.50	$\underline{1.70}$[l]	
	$\Delta V_m/V_s$	LDA	−0.124		−0.055[m]		$\Delta V_m/V_s$	LDA	0.254		
		PBE	−0.111					PBE	0.297	$\underline{0.305}$[l]	
		PBEsol	−0.125					PBEsol	0.267		
		SCAN	−0.091					SCAN	0.269	$\underline{0.291}$[l]	
	dT_m/dp	LDA	−63		−20[k]		dT_m/dp	LDA	123		
		PBE	−57		−38[m]			PBE	153	$\underline{153}$[l]	
		PBEsol	−63					PBEsol	137		
		SCAN	−44					SCAN	140	$\underline{134}$[l]	
Sn	S_{ls}	SCAN	1.81		1.7[h]						
	$\Delta V_m/V_s$	SCAN	0.039		0.023[h]						
	dT_m/dp	SCAN	45		27[n]						

[a] Data from Ref. [1].
[b] Data from Ref. [3].
[c] Data from Ref. [10].
[d] Data from Ref. [60].
[e] Data from Ref. [2].
[f] Data from Ref. [61].
[g] Data from Ref. [53].
[h] Data from Ref. [57].
[i] Data from Ref. [62].
[j] Data from Ref. [63].
[k] Data from Ref. [51].
[l] Data from Ref. [55].
[m] Data from Ref. [64].
[n] The melting curve of tin was calculated from the volume of β-tin at 453 K reported in Ref. [65], the volume change by the fusion written in Ref. [66], and the heat of fusion and melting temperature reported in Ref. [57].

studies of Si in previous publications [3,53], where the trends in the melting points among different functionals were well described by the trends in the energy differences. This is because the energy and structure of liquid Si and Ge can be qualitatively described by the sixfold coordinated structure observed in the β-tin structure. The results summarized in Fig. 8 indicate that this empirical rule is also roughly applicable to Ge. For Ge, the energy difference between the α-tin and β-tin structures is significantly smaller, and thus, its melting temperature is lower than that of Si. In the case of Sn, the α-tin structure is less stable than the β-tin structure, and melting is obviously from the β-tin structure itself. This implies that the energy difference between both structures might not have a direct relevance for the melting temperature, nevertheless, the linear relation between the melting temperature and energy difference still seems to apply approximately.

For Al, LDA, PBEsol, and SCAN closely reproduce the experimental melting point while PBE underestimates it. Similarly for MgO, LDA, PBEsol, and SCAN reproduce the experimental melting point well, while PBE again underestimates it.

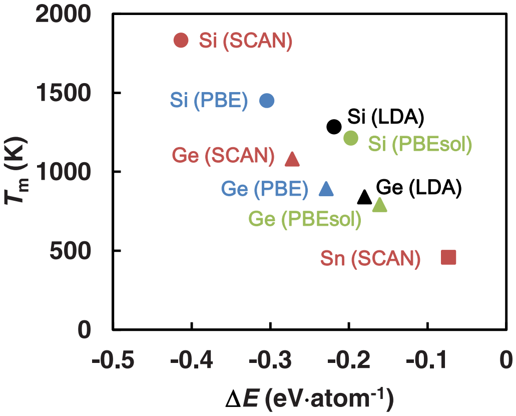

FIG. 8. Melting point T_m calculated by the interface pinning method using the machine learning force fields and the thermodynamic perturbation corrections vs the energy difference ΔE between the α- and β-tin structures obtained from DFT calculations at 0 K. ΔE is defined to be negative when the α-tin structure is more stable than the β-tin structure.

In summary, SCAN is judged to provide the most balanced accuracy within the tested functionals for thermodynamic properties of metallic, covalent and ionic materials. Semilocal functionals always yield lower melting temperatures, but whether LDA, PBEsol, or PBE performs better is not *a priori* clear. For the tetrahedrally coordinated semiconductors LDA and PBEsol are worse (obviously they underbind the diamond structure), whereas for densely packed materials such as Al, the melting temperature is increased towards the experiment. Since LDA and PBEsol increase the binding energy on average, an increase of the melting temperature for phase transitions where the local structure changes little is in line with what one would expect.

V. CONCLUSION

An on-the-fly MLFF generation method has been developed and integrated into an electronic structure code. In the developed method, the machine predicts not only the energy, forces and the stress tensor but also the uncertainty on the basis of Bayesian inference. Although the predicted error possesses quantitative discrepancies from the real error, the relative trend of the real error is reproduced by the Bayesian inference. This allows the machine to decide whether FP calculations are required or can be bypassed. This error estimation and decision scheme enhances the self-learning ability in the MLFF generation and dramatically reduces the need for human intervention and supervision. The developed method was applied to the calculation of melting points of Al, Si, Ge, Sn, and MgO. The application demonstrates that the on-the-fly method indeed enables an efficient creation of accurate MLFFs for metallic, covalent and ionic materials. The MD simulations are more than two orders of magnitude accelerated by the on-the-fly scheme even during learning. Furthermore, for large unit cells, the generated force fields are more than three orders of magnitude faster for MD simulations than FP calculations. This allowed us to to calculate the melting points of five materials using the interface pinning method in a fraction of the compute time that would have been required without ML. The melting temperatures predicted by the MLFFs already agree well with the FP results, but it is straightforward and involves only little overhead to employ thermodynamic perturbation theory and correct for the remaining errors. Our on-the-fly method is universally applicable to a wide variety of multi-element complex materials. We believe that this has the potential to become a new working paradigm in the materials science community.

Concerning melting temperatures, we observe that the SCAN functional is clearly a step forward in accurate predictions compared to experiments. SCAN consistently outperforms the semi-local functionals tested in the present work. A general trend between the semi-local functionals can not be made out. For Al, where the local structure remains 12-fold coordinated upon melting, the melting temperature increases from PBE over LDA to PBEsol towards experiment, in line with the increased cohesive energies predicted by PBEsol and LDA. For Si and Ge, however, the melting temperature decreases from PBE over LDA to PBEsol away from experiment. This trend is related to a destabilization of the cubic diamond structure by PBEsol. Clearly, melting temperatures are a tough test for the performance of density functionals, and only SCAN is reasonably satisfactory in this regard.

ACKNOWLEDGMENT

All authors gratefully thank Ryoji Asahi for his many suggestions on the application and use of machine-learning methods to materials sciences.

APPENDIX A: RADIAL BASIS FUNCTIONS

The number of the radial basis functions N_R^l is automatically determined such that a linear combination of the radial basis functions reproduces the radial functions $f_{lm}(r, r_{ij})$ with a predefined accuracy. The functions $f_{lm}(r, r_{ij})$ are obtained by expanding the broadened atomic distribution ρ_i [see Eqs. (3) and (2)] in products of spherical harmonics and the radial functions as

$$\rho_i(\mathbf{r}) = \sum_{l=0}^{L_{\max}} \sum_{m=-l}^{l} \sum_{j=1}^{N_a} f_{lm}(r, r_{ij}) Y_{lm}^*(\hat{\mathbf{r}}_{ij}) Y_{lm}(\hat{\mathbf{r}}), \tag{A1}$$

$$f_{lm}(r, r_{ij}) = \frac{4\pi}{\left(\sqrt{2\sigma_{\text{atom}}^2 \pi}\right)^3} f_{\text{cut}}(r_{ij}) \times \exp\left(-\frac{r^2 + r_{ij}^2}{2\sigma_{\text{atom}}^2}\right) \iota_l\left(\frac{r r_{ij}}{\sigma_{\text{atom}}^2}\right). \tag{A2}$$

To derive this equation, the following theorem has been used:

$$\exp\left(\frac{-|\mathbf{r} - \mathbf{r}_{ij}|^2}{2\sigma_{\text{atom}}^2}\right) = 4\pi \exp\left(-\frac{r^2 + r_{ij}^2}{2\sigma_{\text{atom}}^2}\right) \times \sum_{l=0}^{L_{\max}} \sum_{m=-l}^{l} \iota_l\left(\frac{r r_{ij}}{\sigma_{\text{atom}}^2}\right) \times Y_{lm}^*(\hat{\mathbf{r}}_{ij}) Y_{lm}(\hat{\mathbf{r}}). \tag{A3}$$

The radial part is expanded in a set of radial basis functions $\chi_{nl}(r) = j_l(q_{nl} r)$ as

$$\sum_{j=1}^{N_a} f_{lm}(r, r_{ij}) = \sum_{n=1}^{N_R^l} c_{nlm}^i \chi_{nl}(r). \tag{A4}$$

Here the parameters q_{nl} are set such that $j_l(q_{nl} R_{\text{cut}}) = 0$. The number of the radial basis functions N_R^l is determined to satisfy Eq. (A4) within a desired precision. In our implementation, in advance of the MD simulation, the radial function $f_{lm}(r, r_{ij})$ is calculated using Eq. (A2) on 100 radial grid points for r ranging from 0 to R_{cut} and r_{ij} ranging from 0.5 Å to R_{cut}. The number of radial basis functions N_R^l is determined to reproduce the original values of $f_{lm}(r, r_{ij})$ within an error of ± 0.02. This means that the width of the broadening σ_{atom} in Eq. (2) alone determines the number of basis functions.

APPENDIX B: ASSUMPTIONS NECESSARY FOR DERIVING THE POSTERIOR DISTRIBUTIONS

The formulation of the posterior distributions $p(\mathbf{w}|\mathbf{Y})$ and $p(\mathbf{y}|\mathbf{Y})$ starts from two assumptions.

(i) The FP data vector $\mathbf{y}^\alpha$ deviates from the model vector $\boldsymbol{\phi}^\alpha \mathbf{w}$. The distribution of the deviation is assumed to be described by a Gaussian function with a covariance matrix of $\sigma_v^2 \mathbf{I}$:

$$p(\mathbf{Y}|\mathbf{w}) = \mathcal{N}\big(\boldsymbol{\Phi}\mathbf{w}, \sigma_v^2 \mathbf{I}\big). \tag{B1}$$

$p(\mathbf{Y}|\mathbf{w})$ denotes the probability to observe the FP data $\mathbf{Y}$ after the determination of the coefficients $\mathbf{w}$.

(ii) The prior probability to find the vector $\mathbf{w}$ is assumed to be described by a Gaussian distribution with a mean vector at zero and a covariance matrix of $\sigma_w^2 \mathbf{I}$:

$$p(\mathbf{w}) = \mathcal{N}\big(\mathbf{0}, \sigma_w^2 \mathbf{I}\big). \tag{B2}$$

On the basis of these two assumptions and the Bayesian theorem, the posterior distribution $p(\mathbf{w}|\mathbf{Y})$ is derived as

$$p(\mathbf{w}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{w}) p(\mathbf{w})}{p(\mathbf{Y})}, \tag{B3}$$

$$p(\mathbf{Y}) = \int p(\mathbf{Y}|\mathbf{w}) p(\mathbf{w}) d\mathbf{w}. \tag{B4}$$

The equation can be converted to the Gaussian distribution written as Eq. (25) by the completing square method [38]. From this posterior distribution, another posterior distribution $p(\mathbf{y}|\mathbf{Y})$ is obtained as

$$p(\mathbf{y}|\mathbf{Y}) = \int p(\mathbf{y}|\mathbf{w}) p(\mathbf{w}|\mathbf{Y}) d\mathbf{w}. \tag{B5}$$

Similarly to $p(\mathbf{w}|\mathbf{Y})$, this distribution can be converted to a Gaussian distribution specified in Eq. (29) [38].

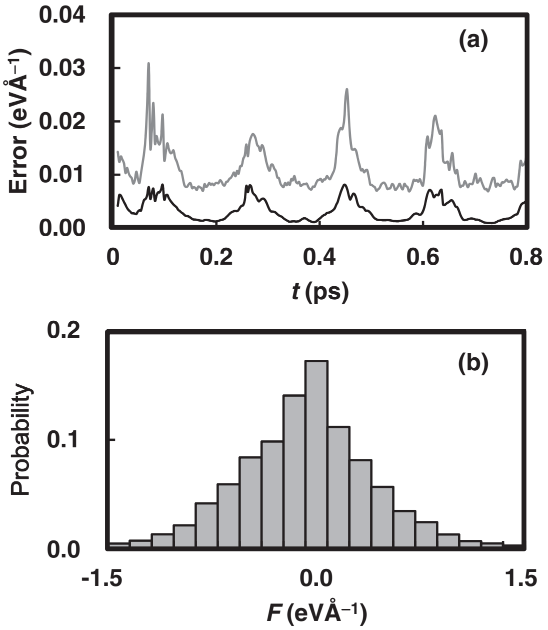

FIG. 9. (a) Real (root mean square error) (gray line) and Bayesian error (black line) on forces (a) and (b) histogram of training data on forces during the training mainly on force data.

APPENDIX C: MAXIMIZATION OF EVIDENCE FUNCTION

The maximization of the evidence function Eq. (31) with respect to the parameters σ_v^2 and σ_w^2 is carried out by simultaneously solving the following equations derived from $\partial p / \partial \sigma_v^2 = \partial p / \partial \sigma_w^2 = 0$ (see Ref. [38]):

$$\sigma_w^2 = \frac{|\bar{\mathbf{w}}|^2}{\gamma}, \tag{C1}$$

$$\sigma_v^2 = \frac{|\mathbf{Y} - \boldsymbol{\phi}\bar{\mathbf{w}}|^2}{M - \gamma}, \tag{C2}$$

$$\gamma = \sum_{k=1}^{N_B} \frac{\lambda_k}{\lambda_k + 1/\sigma_w^2}. \tag{C3}$$

Here, λ_k are the eigenvalues of the matrix $\boldsymbol{\Phi}^T \boldsymbol{\Phi} / \sigma_v^2$. As described in Ref. [41], if all eigenvalues are used in the actual computations of Eqs. (C1)–(C3), numerical instabilities happen because the nonregularized matrix $\boldsymbol{\Phi}^T \boldsymbol{\Phi} / \sigma_v^2$ can become nonpositive definite. In order to avoid this problem, eigenvalues smaller than 10^{-10} are excluded from the calculations.

APPENDIX D: TRAINING ON FORCES

Figure 9 shows the real error and the Bayesian error during the training on the MgO solid at 300 K. In the same figure, a histogram of the training data on forces collected during this short training simulation is shown. In this particular training simulation, the dimensionless training data of the energy and stress tensor components were less weighted than the training force data (10^{-6} times smaller weights). Thus, the training is executed almost only on the force data. The accuracy on the forces predicted by the MLFF is obviously improved, and the Bayesian error resembles the real error more quantitatively .

APPENDIX E: CUR ALGORITHM

In the following, we denote the element $K(i, j)$ in Eq. (13) as K_{ij} and the matrix comprised of K_{ij} for all candidates of the local reference configurations as $\mathbf{K}$ (both the column and row dimensions of $\mathbf{K}$ are equal to the number of the candidates). The formulation of the CUR algorithm starts from the diagonalization of the matrix $\mathbf{K}$.

$$\mathbf{U}^T\mathbf{K}\mathbf{U} = \mathbf{L} = \mathrm{diag}(l_1, \ldots, l_{N_B}), \tag{E1}$$

where $\mathbf{U}$ is the eigenvector matrix defined as

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{N_B}), \tag{E2}$$

$$\mathbf{u}_j^T = (u_{1j}, \ldots, u_{N_B j}), \tag{E3}$$

By using the notations in Eq. (E2), Eq. (E1) can be rewritten as follows:

$$\mathbf{k}_j = \sum_{\xi=1}^{N_B} (u_{j\xi} l_\xi)\mathbf{u}_\xi, \tag{E4}$$

where $\mathbf{k}_j$ denotes the jth column vector of the matrix $\mathbf{K}$. In the original CUR algorithm [45], the columns of the matrix $\mathbf{K}$ are maintained when they are strongly correlated to the eigenvectors $\mathbf{u}_\xi$ with large eigenvalues l_ξ. This algorithm was originally developed to efficiently select a few significant local reference configurations from many configurations. However, in our on-the-fly force field generation, we need an efficient algorithm that can select few insignificant configurations, because the number of configurations discarded by the sparsification is usually small. To this end, we have modified the algorithm. In our implementation, we dispose of those columns of $\mathbf{K}$ that are strongly correlated with the N_{low} eigenvectors $\mathbf{u}_\xi$ with the small eigenvalues l_ξ. The local configurations corresponding to those columns are disregarded. Similarly to the original CUR algorithm, the correlation is measured by the statistical leverage scoring determined for each column of $\mathbf{K}$ as

$$\omega_j = \frac{1}{N_{\mathrm{low}}} \sum_{\xi=1}^{N_B} \gamma_{\xi j}, \tag{E5}$$

$$\gamma_{\xi j} = \begin{cases} u_{j\xi}^2, & \text{if } l_\xi < 10^{-10} \\ 0, & \text{otherwise} \end{cases}. \tag{E6}$$

---

[1] O. Sugino and R. Car, Phys. Rev. Lett. **74**, 1823 (1995).
[2] G. A. de Wijs, G. Kresse, and M. J. Gillan, Phys. Rev. B **57**, 8223 (1998).
[3] D. Alfè and M. J. Gillan, Phys. Rev. B **68**, 205212 (2003).
[4] A. Ladd and L. Woodcock, Chem. Phys. Lett. **51**, 155 (1977).
[5] U. Landman, W. D. Luedtke, R. N. Barnett, C. L. Cleveland, M. W. Ribarsky, E. Arnold, S. Ramesh, H. Baumgart, A. Martinez, and B. Khan, Phys. Rev. Lett. **56**, 155 (1986).
[6] P. M. Agrawal, B. M. Rice, and D. L. Thompson, J. Chem. Phys. **119**, 9617 (2003).
[7] J. R. Morris and X. Song, J. Chem. Phys. **116**, 9352 (2002).
[8] J. J. Hoyt and M. Asta, Phys. Rev. B **65**, 214106 (2002).
[9] R. García Fernández, J. L. F. Abascal, and C. Vega, J. Chem. Phys. **124**, 144506 (2006).
[10] U. R. Pedersen, F. Hummel, G. Kresse, G. Kahl, and C. Dellago, Phys. Rev. B **88**, 094101 (2013).
[11] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[12] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[13] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, Phys. Rev. X **8**, 041048 (2018).
[14] L. Bonati and M. Parrinello, Phys. Rev. Lett. **121**, 265701 (2018).
[15] T. Morawietz, A. Singraber, C. Dellago, and J. Behler, Proc. Natl. Acad. Sci. USA **113**, 8368 (2016).
[16] V. L. Deringer, N. Bernstein, A. P. Bartók, M. J. Cliffe, R. N. Kerber, L. E. Marbella, C. P. Grey, S. R. Elliott, and G. Csányi, J. Phys. Chem. Lett. **9**, 2879 (2018).
[17] N. Artrith and A. Urban, Comput. Mater. Sci. **114**, 135 (2016).
[18] S. Chiriki, S. Jindal, and S. Bulusu, J. Chem. Phys. **146**, 084314 (2017).
[19] S. Faraji, S. A. Ghasemi, S. Rostami, R. Rasoulkhani, B. Schaefer, S. Goedecker, and M. Amsler, Phys. Rev. B **95**, 104105 (2017).
[20] Z. Li, J. R. Kermode, and A. De Vita, Phys. Rev. Lett. **114**, 096405 (2015).
[21] E. V. Podryabinkin and A. V. Shapeev, Comp. Mater. Sci. **140**, 171 (2017).
[22] K. Miwa and H. Ohno, Phys. Rev. Mater. **1**, 053801 (2017).
[23] T. L. Jacobsen, M. S. Jørgensen, and B. Hammer, Phys. Rev. Lett. **120**, 026102 (2018).
[24] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam, Phys. Rev. Lett. **122**, 225701 (2019).
[25] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[26] G. Kresse and J. Furthmüller, Comput. Mater. Sci. **6**, 15 (1996).
[27] A. P. Bartók, R. Kondor, and G. Csányi, Phys. Rev. B **87**, 184115 (2013).
[28] W. J. Szlachta, A. P. Bartók, and G. Csányi, Phys. Rev. B **90**, 104108 (2014).
[29] V. L. Deringer and G. Csányi, Phys. Rev. B **95**, 094203 (2017).
[30] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Sci. Adv. **3**, e1701816 (2017).
[31] A. P. Bartók, Ph.D. thesis, University of Cambridge, 2009.
[32] K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, J. Chem. Phys. **148**, 241727 (2018).
[33] K. Gubaev, E. V. Podryabinkin, G. L. Hart, and A. V. Shapeev, Comp. Mater. Sci. **156**, 148 (2019).
[34] A. Glielmo, C. Zeni, and A. De Vita, Phys. Rev. B **97**, 184307 (2018).
[35] L. Nachbin and L. Bechtolsheim, *The Haar Integral* (Van Nostrand, Princeton, NJ, 1965).
[36] M. J. Willatt, F. Musil, and M. Ceriotti, Phys. Chem. Chem. Phys. **20**, 29661 (2018).
[37] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.100.014105 for a detailed description of the applied computational techniques and settings.
[38] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, New York, 2006).
[39] S. F. Gull and J. Skilling, *Maximum Entropy Bayesian Methods,* Fundamental Theories of Physics (Springer, Dordrecht, 1989).
[40] D. J. C. MacKay, Neural Comput. **4**, 415 (1992).
[41] R. Jinnouchi and R. Asahi, J. Phys. Chem. Lett. **8**, 4279 (2017).

[42] K. Miwa and H. Ohno, Phys. Rev. B **94**, 184109 (2016).

[43] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, 2nd ed. (Oxford University Press, New York, 2017).

[44] S. Y. Kung, *Kernel Methods and Machine Learning* (Cambridge University Press, Princeton, NJ, 2014).

[45] M. W. Mahoney and P. Drineas, Proc. Natl. Acad. Sci. USA **106**, 697 (2009).

[46] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566 (1980).

[47] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[48] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke, Phys. Rev. Lett. **100**, 136406 (2008).

[49] J. Sun, A. Ruzsinszky, and J. P. Perdew, Phys. Rev. Lett. **115**, 036402 (2015).

[50] D. Frenkel and B. Smit, in *Understanding Molecular Simulation*, edited by D. Frenkel and B. Smit, 2nd ed. (Academic Press, San Diego, 2002), pp. 525–532.

[51] D. Bimberg, R. Blachnik, M. Cardona, P. J. Dean, T. Grave, G. Harbeke, K. Hübner, U. Kaufmann, W. Kress, O. Madelung *et al.*, *Physics of Group IV Elements and III-V Compounds/ Physik der Elemente der IV. Gruppe und der III-V Verbindungen* (Springer, Berlin, 1981).

[52] R. Lide, *CRC Handbook of Chemistry and Physics,* 74th ed. (CRC Press, Boca Raton, 1994).

[53] F. Dorner, Z. Sukurma, C. Dellago, and G. Kresse, Phys. Rev. Lett. **121**, 195701 (2018).

[54] A. Zerr and R. Boehler, Nature (London) **371**, 506 (1994).

[55] M. Rang and G. Kresse, Phys. Rev. B **99**, 184103 (2019).

[56] C. Ronchi and M. Sheindlin, J. Appl. Phys. **90**, 3325 (2001).

[57] I. Barin and O. Knacke, *Thermochemical Properties of Inorganic Substances* (Springer, Berlin, 1973).

[58] M. Parrinello and A. Rahman, Phys. Rev. Lett. **45**, 1196 (1980).

[59] M. Parrinello and A. Rahman, J. Appl. Phys. **52**, 7182 (1981).

[60] A. R. Ubbelohde, *The Molten State of Matter* (Wiley, New York, 1978).

[61] R. C. Wilhoit, J. Chao, and K. R. Hall, J. Phys. Chem. Ref. Data Suppl. **14**, 1 (1985).

[62] J. P. Gabathuler and S. Steeb, Z. Naturfors. A **34**, 1314 (1979).

[63] J. F. Cannon, J. Phys. Chem. Ref. Data **3**, 781 (1974).

[64] A. Jayaraman, W. Klement, and G. C. Kennedy, Phys. Rev. **130**, 540 (1963).

[65] E. A. Brandes, *Smihells Metals Reference Book*, 6th ed. (Butterworths, London, 1983).

[66] P. G. Harrison, *Chemistry of Tin* (Blackie, Glasgow, 1989).