

Neural networks for local structure detection in polymorphic systems

Philipp Geiger and Christoph Dellago

Citation: *The Journal of Chemical Physics* **139**, 164105 (2013); doi: 10.1063/1.4825111

View online: <http://dx.doi.org/10.1063/1.4825111>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/139/16?ver=pdfcov>

Published by the [AIP Publishing](#)

Articles you may be interested in

[Atom-centered symmetry functions for constructing high-dimensional neural network potentials](#)

J. Chem. Phys. **134**, 074106 (2011); 10.1063/1.3553717

[Accurate prediction of higher-level electronic structure energies for large databases using neural networks, Hartree–Fock energies, and small subsets of the database](#)

J. Chem. Phys. **131**, 124127 (2009); 10.1063/1.3231686

[Structure of coexisting liquid phases of supercooled water: Analogy with ice polymorphs](#)

J. Chem. Phys. **126**, 241103 (2007); 10.1063/1.2753145

[Determination of elastic properties of a film-substrate system by using the neural networks](#)

Appl. Phys. Lett. **85**, 6161 (2004); 10.1063/1.1841472

[An electronic system for simulation of neural networks with a micro-second real time constraint](#)

AIP Conf. Proc. **583**, 76 (2001); 10.1063/1.1405266



AIP | Journal of
Applied Physics

Journal of Applied Physics is pleased to
announce **André Anders** as its new Editor-in-Chief

Neural networks for local structure detection in polymorphic systems

Philipp Geiger and Christoph Dellago^{a)}

Faculty of Physics, University of Vienna, Boltzmannngasse 5, 1090 Vienna, Austria

(Received 12 July 2013; accepted 26 September 2013; published online 23 October 2013)

The accurate identification and classification of local ordered and disordered structures is an important task in atomistic computer simulations. Here, we demonstrate that properly trained artificial neural networks can be used for this purpose. Based on a neural network approach recently developed for the calculation of energies and forces, the proposed method recognizes local atomic arrangements from a set of symmetry functions that characterize the environment around a given atom. The algorithm is simple and flexible and it does not rely on the definition of a reference frame. Using the Lennard-Jones system as well as liquid water and ice as illustrative examples, we show that the neural networks developed here detect amorphous and crystalline structures with high accuracy even in the case of complex atomic arrangements, for which conventional structure detection approaches are unreliable. © 2013 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4825111>]

I. INTRODUCTION

Atomistic simulations carried out on high performance computers provide a wealth of detailed information on condensed matter processes. While visualization of these processes using computer graphics can yield important insights and stimulate our imagination, a true understanding of the underlying physical mechanisms requires a quantitative and automated analysis of the generated data. Such an analysis often involves the detection of particular atomistic structures based on their local environment. In the simulation of the crystallization of a supercooled liquid, for instance, it is necessary to differentiate between atoms that are part of the growing crystal and atoms that belong to the liquid. Similarly, in studying the microscopic mechanism of structural phase transitions it is crucial to be able to tell apart different crystal structures on a local level. Also when one investigates the structure and dynamics of defects in solids it is necessary to recognize particular atomic arrangements and to follow their motion in time. The ability to distinguish and classify local atomistic structures is not only important to analyze the output of computer simulation *a posteriori*, but also to steer computer simulations towards interesting regions of configuration space using biasing schemes such as umbrella sampling^{1,2} or metadynamics³ and to define long-lived states in path sampling simulations.^{4,5} Furthermore, automatic local structure detection schemes are also useful for the analysis of experimental data obtained with high resolution imaging techniques.⁶

Computational methods for structure recognition are usually formulated in terms of order parameters that, ideally, satisfy several important criteria: (a) For a given phase, the order parameter should include all configurations belonging to that phase and exclude all others (avoiding mis-assignments is particularly important, if the order parameter is used to drive a transition); (b) structures should be assigned accurately not only in a perfect crystal, but also in the presence of thermal

fluctuations or mechanical distortions; (c) the order parameter should be local with a well defined and controllable sensitivity region; (d) the order parameter should also be local in time, i.e., no time averaging should be required; (e) the order parameter should be able to distinguish all stable and metastable phases of a material over a wide range of conditions across the phase diagram; (f) the method should recognize defect structures, for instance arising from imperfect crystallization, and structures forming at surfaces and interfaces; (g) the order parameter should be invariant with respect to global rotations and translations as well as to permutations of identical atoms; (h) the method should be flexible such that it can be easily adapted to work for other substances or unexpected phases; (i) finally, the method should be computationally inexpensive and easy to implement.

During the past decades several computational approaches have been developed for automatic structure detection, which fulfill some but not all of the criteria specified above. In the common neighbor analysis (CNA) method, originally developed to distinguish emerging structures in freezing Lennard-Jones clusters⁷ and later used to detect crystalline regions as well as defects in various systems,^{8,9} local structures of individual atoms are assigned based on an analysis of the environment shared by neighboring atoms. In the centro-symmetric deviation method (CSD), introduced to study the nucleation and motion of dislocations in centro-symmetric materials such as fcc-metals,¹⁰ the local distortion caused by defects is assessed by computing a parameter sensitive to the angle formed by opposite bond vectors. A similar, but more general approach is followed in the bond angle distribution method (BAD), in which the distribution of angles formed by nearest neighbor bonds is used as characteristic feature to distinguish different crystal structures.¹¹ The most versatile (and currently most popular) class of methods for local structure recognition are, however, based on the Steinhardt local bond order parameters.^{12–15} In this approach, explained in more detail later in the paper, the particular symmetries of local structures are picked up by combinations of

^{a)}Electronic mail: christoph.dellago@univie.ac.at

spherical harmonics that are made invariant against rotations, translations, and permutations of identical particles and serve as characteristic structural fingerprints for different ordered and disordered structures.

While the methods mentioned in the previous paragraph have been successfully applied to a variety of problems, they often yield unreliable results particularly for complex open structures and in the presence of elastic deformations and thermal fluctuations. For instance, it is notoriously difficult to distinguish local configurations of liquid water and of the various forms of crystalline and amorphous ice.¹⁶ The ability to do that, however, is needed to study the nucleation of ice in supercooled water. As discussed by Brukhno *et al.*,¹⁷ standard Steinhardt bond order parameters are ineffective in detecting hexagonal and cubic ice because oxygen atoms with different tetrahedral hydrogen bonding patterns occur. To resolve this issue, these authors have introduced the maximum projection method, which is capable of distinguishing between liquid water and hexagonal and cubic ice, albeit at the price of introducing a preferred direction and thus breaking rotational invariance. As a remedy to this situation, Reinhardt *et al.*¹⁸ suggested a parameter based on particular Steinhardt bond order parameters and used it to drive nucleation of ice in supercooled water. In these simulations it proved necessary to explicitly remove unphysical chain structures that were forced on the system by a bias on the order parameter. Another order parameter to distinguish liquid water from hexagonal and cubic ice was developed by Chau and Hardwick^{19,20} based on detecting local tetrahedral structures. Recently, Sanz *et al.* demonstrated²¹ that the averaged bond order parameters \bar{q}_4 and \bar{q}_6 suggested by Lechner and Dellago¹⁵ can be used to distinguish liquid water, hexagonal ice Ih, and cubic ice Ic from each other. None of the order parameters developed so far for water and ice, however, are capable of discriminating between all phases of ice and liquid water.

Here, we treat the assignment of local structures as a pattern recognition problem that can be addressed with machine learning algorithms. We show how an appropriately trained neural network can be used to accurately detect local ordered structures and demonstrate the practical applicability of the algorithm by distinguishing the different phases of the Lennard-Jones system and of water in a broad range of conditions. Our approach is inspired by the neural network based method for representing potential energy surfaces for condensed matter systems recently developed by Behler and Parrinello.^{22,23} In this method, the total energy of the system is written as a sum of local contributions, each of which is determined by applying a neural network to a set of numbers characterizing the local environment of individual atoms. The structural fingerprints used as input for the neural network are provided by a collection of symmetry functions that are sensitive to distances and angles in a small region around the central atom and are furthermore invariant with respect to rotations, translations and permutations. In our neural network for structure recognition, we use similar symmetry functions but they are sensitive only to the positions of the atoms in a small and well defined region around a given atom. When properly trained using configurations with known structures, the neural network succeeds in recognizing structures even

in polymorphic systems with rich phase diagrams, for which conventional structure assignment methods fail. The method proposed here is flexible and can easily be adapted to a variety of substances.

The remainder of this article is organized as follows. In Sec. II, we introduce the method and explain in detail how the neural network is constructed and trained. The neural network is then applied to two test cases in Sec. III. We first use it to distinguish the various structures in the Lennard-Jones system and compare the performance of the neural network with that of Steinhardt bond order parameters. After that, we apply the neural network to the more challenging problem of distinguishing various phases of liquid water and ice over wide ranges of pressures and temperatures. Using the neural network, we compute the free energy for the nucleation of an ice crystal in supercooled water. A discussion of the strengths and limitations of the method is provided in Sec. IV.

II. NEURAL NETWORK FOR STRUCTURE RECOGNITION

Our approach for assigning local structures is based on the neural network method for energy and force calculations recently put forward by Behler and Parrinello.^{22,23} In this method, which provides the accuracy of density functional theory at a fraction of its cost, the total energy of the system is expressed as a sum of contributions from local regions centered around individual atoms. Each region is characterized by a set of generalized coordinates, called symmetry functions, which are functions of the atomic coordinates and serve as input for a feed-forward neural network. Based on these structural fingerprints, the neural network then predicts the local energy contribution of each atom, essentially by interpolation. Here, we adopt this approach and use the information encoded in the symmetry functions to detect local atomic structures. In Secs. II A–II D, we first describe the neural network, then discuss the definition of the symmetry functions, which are designed to be sensitive to the local environment of a given atom, and finally explain how the neural network is trained.

A. Feed-forward neural network

A neural network can be viewed as a complex nonlinear (scalar- or vector-valued) function that depends on a set of input variables and a possibly very large set of parameters, which can be tuned to obtain the desired behavior. Inspired by systems of biological neurons such as the brain, the basic concept of a neural network is depicted in Fig. 1. A feed-forward network such as the one used in our work consists of three parts: The input neurons, one or more calculation layers, and the output nodes. The input interface receives the external information and corresponds to the part of a biological neural network in which sensory stimuli are converted into signals. The information is then passed to the calculation layers, where the input signals are weighted and processed. The last part of the neural network is the output interface, where the final data processing step occurs and the information is converted into the desired output form. The size and

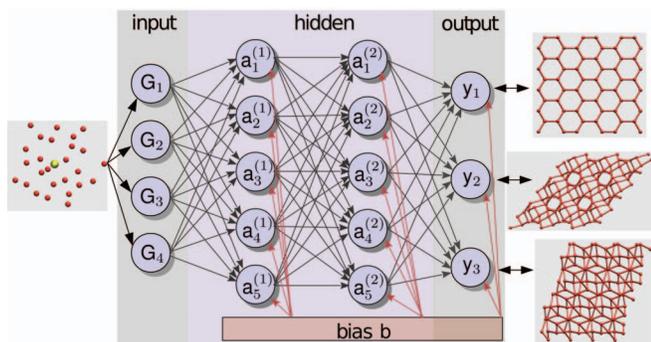


FIG. 1. Four layer feed-forward neural network with $I \times H \times H \times O$ topology for local structure identification. Here, I , H , and O refer to the number of neurons in the input layer, the hidden layers and the output layer, respectively. For a given configuration r , depicted schematically on the left hand side with the central atom shown in yellow, a set of symmetry functions $\{G_l(r)\}$ is computed. These symmetry functions serve as input for the first layer of the neural network, from which the data are passed to the hidden layers. Each neuron, or node, in the calculation layers takes the values of the neurons of the previous layer, as indicated by the black arrows, and performs a weighted sum over them. The weights, assigned during the training phase, define the relative importance of the connections between neuron pairs. After the sum is formed, a bias is added to the value of each neuron as indicated by the red arrows. Also the bias values are determined during the training of the network. For each node, the result is then fed into an activation function, which determines how active a neuron is, i.e., how much it contributes to the output. Finally, the output layer converts the processed information into a vector in which each component signals the occurrence of one of the possible structures.

topology of a neural network is specified using the notation $I \times H \times H \times O$, where I , H , and O denote the number of nodes in the input, hidden, and output layers, respectively.

Based on this network topology, the structure recognition works as follows. Consider a given system configuration $r = \{r_1, r_2, \dots, r_N\}$ consisting of the Cartesian coordinates of all N atoms. For simplicity we assume that there is only one atom species in the system, but we emphasize that the method can easily be generalized to more atom species. To determine the local structure around atom i one first computes a set of I symmetry functions $\{G_l^{[i]}(r)\}$, which depend on the Cartesian coordinates of the atoms in the vicinity of atom i . The symmetry functions, explained in detail below, are designed to characterize the local environment of a given atom. The symmetry functions are then turned over to the first layer, where for each of the neurons of this layer a weighted sum, $\sum_l w_{ml}^{(1)} G_l$, over the input (i.e., the symmetry functions) is computed. Here, the synaptic weight $w_{ml}^{(1)}$ connects the input neuron l with the neuron m of the first hidden layer. Each sum is then shifted by a bias (or threshold) $b_m^{(1)}$ and then passed through an activation function $\varphi(x)$, thus yielding the output of the first hidden layer of neurons,

$$a_m^{(1)} = \varphi \left(\sum_{l=1}^I w_{ml}^{(1)} G_l + b_m^{(1)} \right). \quad (1)$$

The output of the first hidden layer is then propagated to the second hidden layer and processed in a similar way using the weights $w_{nm}^{(2)}$ and biases $b_n^{(2)}$,

$$a_n^{(2)} = \varphi \left(\sum_{m=1}^H w_{nm}^{(2)} a_m^{(1)} + b_n^{(2)} \right). \quad (2)$$

Finally, the output of the neural network, given in form of the vector $y = \{y_1, y_2, \dots, y_O\}$ of O components, is obtained by carrying out another set of weighted sums, shifting them by the biases $b_p^{(3)}$ and feeding the results into the activation function $\tilde{\varphi}$,

$$y_p = \tilde{\varphi} \left(\sum_{n=1}^H w_{pn}^{(3)} a_n^{(2)} + b_p^{(3)} \right). \quad (3)$$

While we use the hyperbolic tangent as activation function for all neurons in the hidden layer, $\varphi(x) = \tanh(x)$, the identity function $\tilde{\varphi}(x) = x$ is employed as an activation function for the output neurons. This choice of activation functions was found to work well in the case of the neural networks for energy computation.²⁴ The output vector y has one component for each of the O possible structures the network is trained to recognize. If the network detects structure i , component y_i is large, while all other components are small. The complete action of the neural network on the symmetry functions G_l can be written as

$$y_p = \tilde{\varphi} \left(\sum_n w_{pn}^{(3)} \varphi \left(\sum_m w_{nm}^{(2)} \varphi \left(\sum_l w_{ml}^{(1)} G_l + b_m^{(1)} \right) + b_n^{(2)} \right) + b_p^{(3)} \right). \quad (4)$$

Due to the form of the activation function $\varphi(x)$, the output values y_p of the neural network depend on the input G_l in a complex nonlinear way. The large number of weights and biases provides the neural network with the flexibility required to reproduce with high accuracy, a given functional dependence of the output from the input. In the training phase, carried out as explained below, the weights and biases are adjusted such that the neural network correctly classifies the training set consisting of configurations with known structures.

B. Symmetry functions

The information about the spatial arrangement of the atoms within a certain cutoff of a given atom i is encoded in the so-called symmetry functions $\{G_l^{[i]}(r)\}$, which depend on the Cartesian coordinates of the atoms in this region. The functions, which are constructed to be invariant with respect to rotations, translations, and permutations of identical atoms, need to be defined carefully such that they carry sufficient details to reliably detect and classify structures. By tuning the parameters of these functions, they can be made sensitive to the various symmetries the environment of atom i can have. The following types of symmetry-functions have been proven sufficient for accurate structure detection in the two test cases studied in this work:

$$G_1^{[i]}(r) = \sum_{j \neq i} f_c(R_{ij}), \quad (5)$$

$$G_2^{[i]}(r) = \sum_{j \neq i} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}), \quad (6)$$

$$G_3^{[i]}(r) = \sum_{j \neq i} \cos(\kappa R_{ij}) f_c(R_{ij}), \quad (7)$$

$$G_4^{[i]}(r) = \frac{1}{2^\zeta} \sum_{j,k \neq i} (1 + \lambda \cos \theta_{ijk})^\zeta \times e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}), \quad (8)$$

$$G_5^{[i]}(r) = \frac{1}{2^\zeta} \sum_{j,k \neq i} (1 + \lambda \cos \theta_{ijk})^\zeta \times e^{-\eta(R_{ij}^2 + R_{ik}^2)} f_c(R_{ij}) f_c(R_{ik}), \quad (9)$$

$$G_6^{[i]}(r) = \frac{1}{2^\zeta} \sum_{j,k \neq i} (1 + \lambda \cos \theta_{ijk})^\zeta f_a(R_{ij}) f_a(R_{ik}), \quad (10)$$

$$G_7^{[i]}(r) = \frac{1}{2} \sum_{j,k \neq i} \sin[\eta(\theta_{ijk} - \alpha)] f_c(R_{ij}) f_c(R_{ik}), \quad (11)$$

$$G_8^{[i]}(r) = \frac{1}{2} \sum_{j,k \neq i} \sin[\eta(\theta_{ijk} - \alpha)] f_b(R_{ij}) f_b(R_{ik}). \quad (12)$$

Here, the summation indices run over all neighboring atoms within a certain cutoff of the central atom i . As illustrated in Fig. 2, R_{ij} is the distance from particle i to j and θ_{ijk} is the angle spanned by the triplet of atoms i , j , and k . The quantities R_s , α , η , λ , and κ are tunable parameters, which need to be chosen carefully in order to reflect the symmetries characterizing the possible local structures. While symmetry functions $G_1^{[i]}(r)$ to $G_5^{[i]}(r)$ were adopted from Ref. 25, where they were introduced for energy calculations, we developed functions $G_6^{[i]}(r)$ to $G_8^{[i]}(r)$ specifically for structure recognition.

Since one usually wants to detect structures in the immediate surrounding of a given atom, it is important that the

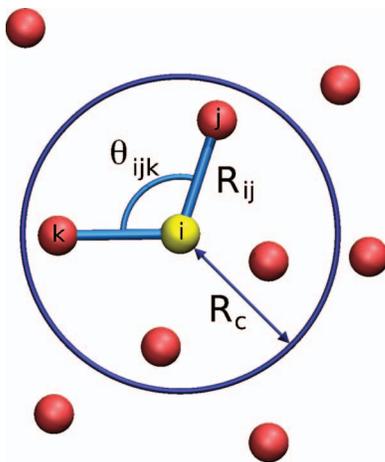


FIG. 2. The symmetry functions $G_i^{[i]}(r)$ around atom i , shown in yellow, are constructed using combinations of atomic distances R_{ij} and angles θ_{ijk} . Only atoms within a certain cutoff R_c contribute to the symmetry functions of the region centered around atom i .

symmetry functions are only sensitive to atomic positions in a small and well defined region. We, therefore, use cutoff functions that are sharper than the soft cosine-cutoff used in neural networks for energy calculations.²² The Fermi cutoff function

$$f_c(R) = \begin{cases} [1 + \exp\{\alpha_c(R - R_c + \epsilon_c)\}]^{-1} & \text{if } R < R_c, \\ 0 & \text{else,} \end{cases} \quad (13)$$

is close to unity for small arguments and then decreases from 1 to 0 in an interval of width $1/\alpha_c$ centered at $R_c - \epsilon_c$. In the limit of large values of α_c the cutoff function $f_c(R)$ turns into a step function with the step located at $R_c - \epsilon_c$. In addition, we defined the cutoff functions

$$f_a(R) = \begin{cases} \cos^2[\eta(R - \mu)] & \text{if } \mu - \frac{\pi}{2\eta} < R < \mu + \frac{\pi}{2\eta}, \\ 0 & \text{else,} \end{cases} \quad (14)$$

and

$$f_b(R) = \begin{cases} \cos^2[v(R - a_1)] & \text{if } a_1 - \frac{\pi}{2v} < R < a_1, \\ 1 & \text{if } a_1 < R < a_r, \\ \cos^2[v(R - a_r)] & \text{if } a_r < R < a_r + \frac{\pi}{2v}, \\ 0 & \text{else,} \end{cases} \quad (15)$$

which are used in the angular symmetry functions $G_6^{[i]}(r)$ and $G_8^{[i]}(r)$. Here, η , μ , v , a_1 , and a_r are free parameters that can be adjusted to make the cutoff functions sensitive only to specific distance ranges. Some symmetry functions with typical parameters are shown in Fig. 3.

For the neural network to work accurately, the parameters of the symmetry functions need to be selected such that the symmetry functions carry a maximum amount of structural information about the environment of a given atom. To select appropriate parameters while keeping the number of symmetry functions low, we first compute the distribution of many symmetry function candidates for the various phases of interest. Then, for each symmetry function, we determine the overlap between the distributions obtained for the different phases and finally select the symmetry functions which best differentiate between the phases, i.e., those corresponding to small overlaps. The symmetry functions are also scaled to the best working range of the neurons, which is from -1 to 1 for the activation functions used in our work.

From the set of symmetry functions determined in this way it is possible to eliminate redundant information by carrying out a sensitivity analysis of the neural network.^{26–28} For this purpose, one determines the derivative of the output of the network with respect to its input. A derivative close to zero then indicates that the corresponding input node has little effect on the output and can be eliminated without compromising the accuracy of the network. Typically, we remove input nodes for which the derivatives for all output nodes are smaller than 0.1% to 1% of the largest value. After the removal of the redundant symmetry functions the neural network needs to be retrained. Using this approach it is possible to reduce the number of symmetry functions required to fully characterize local structures to about 30–40.

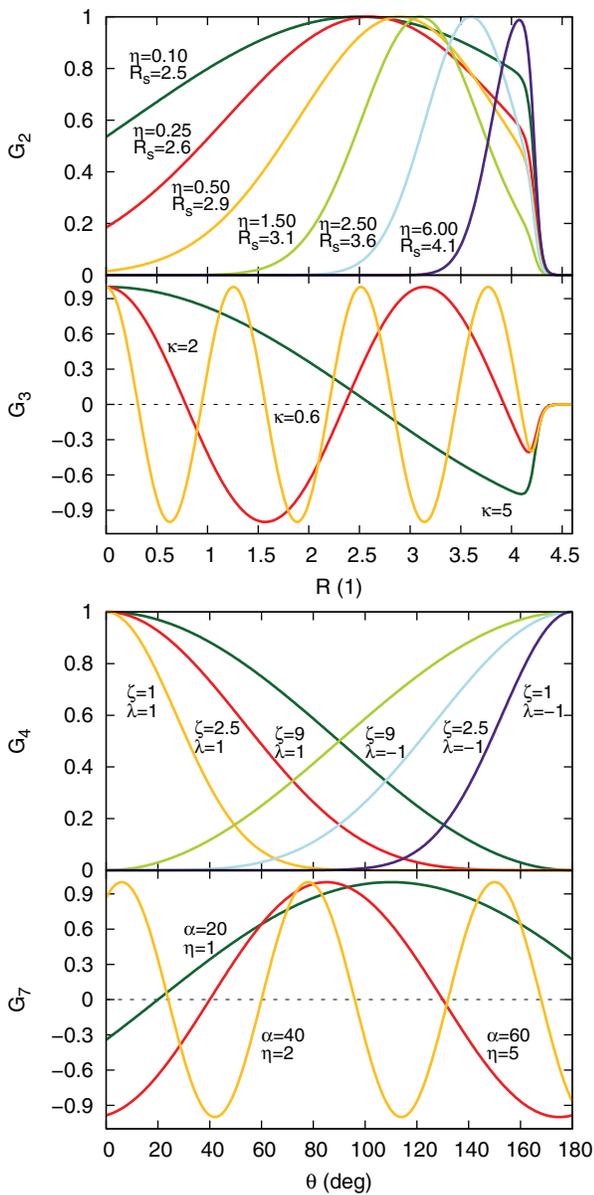


FIG. 3. Various symmetry functions with typical parameter values. The two top panels show the distance dependence of the terms in the two purely radial symmetry functions G_2 and G_3 . The two bottom panels show the angular parts of G_4 and G_7 . The cutoff function parameters $\epsilon_c = 0.2$, $\alpha_c = 30$, and $R_c = 4.43352$ were used.

C. Training the neural network

Once the symmetry functions are defined, the neural network needs to be trained to recognize and distinguish the desired structures. During the training procedure, the weights and biases are tuned based on the information contained in a training set, which consists of a large number of atomic configurations with known structure. Thus, each entry of the training set \mathcal{T} consists of a configuration r and the corresponding structure vector \tilde{y} ,

$$\mathcal{T} = \{r^{(\tau)}, \tilde{y}^{(\tau)}\}, \quad (16)$$

where the superscript τ numbers the entries in the training set. The structure vector $\tilde{y}^{(\tau)}$, which corresponds to the output vector of the neural network, encodes the information about the

type of structure of configuration $r^{(\tau)}$. If configuration $r^{(\tau)}$ is of structure type p , the component $\tilde{y}_p^{(\tau)}$ of the structure vector is assigned the value $\tilde{y}_p^{(\tau)} = +0.9$ and all other components p' are assigned the value $\tilde{y}_{p'}^{(\tau)} = -0.9$.

The training set can be prepared by running straightforward molecular dynamics (MD) simulations for the various phases and extracting local atomic environments from configurations sampled along the MD trajectories. Assuming that during a particular MD run the system remains in the phase from which the simulation was started, all atomic arrangements taken from that run can be assigned the same structure. Since in the course of the MD run the system can, in principle, turn into a different phase and change its structure, it is important to verify whether the original structure still exists at the end of the simulation. This can be done by visual inspection or by the calculation of some global order parameter for the entire system. In the unlikely case that a transition to a different phase has occurred, configurations from this simulation should not be included in the training set. To make sure that the training set is sufficiently diverse, training configurations for a particular phase should be sampled from several MD simulations carried out at different pressures and temperatures. Also, it has proven to be important to avoid correlations between entries in the training set. Thus, local environments extracted from the same trajectory should be significantly separated in time. Typically, the training set comprises tens of thousands of entries. Training sets of this size are necessary to guarantee that most thermal fluctuations typical for each phase are included in the training set.

Based on the information contained in the training set \mathcal{T} , the network is then trained such that for the configurations in the training set the output vectors y of the neural network best match the corresponding structure vectors \tilde{y} . The training procedure can be viewed as an optimization problem, in which the weights $w_{nm}^{(k)}$ and the biases $b_n^{(k)}$ are tuned to minimize the deviation of the predicted structures from the known structures. The target function for the minimization can be expressed as the mean square deviation

$$\chi^2 = \frac{1}{N_{\text{set}} O} \sum_{\tau=1}^{N_{\text{set}}} \sum_{p=1}^O [y_p(r^{(\tau)}) - \tilde{y}_p^{(\tau)}]^2, \quad (17)$$

where the sum over the index τ runs over all N_{set} entries of the training set and $y(r^{(\tau)})$ is the output of the neural network evaluated for structure $r^{(\tau)}$.

We have tested several machine learning methods to minimize χ^2 including back-propagation²⁹ and a nonlinear least squares algorithm.³⁰ We found, as Behler and Parrinello did,^{22,23} that the extended Kalman filter (EKF)^{31,32} converges quickly and does not easily get trapped in local minima. This learning method originates from signal processing theory, where it is used to filter out noise of signals, and has been widely and successfully used for neural network training.^{33–35} The main idea of the extended Kalman filter is to introduce an error covariance matrix, which is then improved iteratively. The iteration is stopped when the mean square deviation χ^2 is sufficiently converged. Finally, the predictive quality of the neural network is tested by computing the mean square deviation between predicted and true structure vectors for an

independent test set, typically of size similar to that of the training set. For a detailed discussion of the Kalman filter see Refs. 31 and 32.

D. Implementation details

In implementing the extended Kalman filtering method, several precautions must be taken to obtain an accurate and robust neural network for structure prediction. The Kalman filter learning technique is an online learning scheme, implying that each basic iteration step, in which one entry of the training set is processed, leads to an update of the entire network state (weights and biases). After the learning procedure has cycled through all entries of the training set, i.e., after one *epoch*, and before the next iteration starts, the order of the entries of the training set is randomly permuted to avoid correlations. The predictive quality of the network can be improved by a modification of the training scheme called multistreaming.³² In this technique, the entries of the training set are bundled into batches consisting of several entries each. The Kalman filter is then applied to entire batches rather than to individual training set entries. While this method improves the outcome of the training of the Kalman filter, it also increases the computational effort required for the training procedure.

It is well known that the extended Kalman filter method suffers from numerical rounding errors, which may lead to a rapid divergence at some point in the learning procedure. Two common approaches exist to overcome this instability.³² The first method updates the square root of the error covariance matrix rather than the covariance matrix itself, producing a very robust iteration prescription. Here, we follow a much simpler approach and introduce some artificial noise in form of a small constant that is added to all diagonal elements of the error covariance matrix at each iteration. This not only improves the stability of the algorithm, but also prevents the networks from getting trapped in local minima of the target function.

To accelerate the training procedure in the starting stage, the initial values of the weights are not assigned at random but are pre-conditioned following an extension of the method proposed in Ref. 36 for neural networks with two hidden layers.³⁷

III. APPLICATIONS

In this section, we demonstrate the application of the neural network for structure recognition using the Lennard-Jones system and water/ice as illustrative examples.

A. Lennard-Jones system

For the Lennard-Jones system conventional order parameters based on spherical harmonics^{12–15} provide an accurate way to distinguish between the liquid and the various crystalline phases. We first apply the neural network approach to this system for comparison. All quantities are given in reduced Lennard-Jones units, i.e., distances are given in units of σ , energies in units of ϵ and time in units of $\sqrt{m\sigma^2/\epsilon}$, where ϵ

and σ are the Lennard-Jones parameters and m is the particle mass.

1. Training set

Depending on temperature and pressure, the Lennard-Jones system exists in the liquid state or in one of several crystalline states including the hexagonal close packed structure (hcp), the face centered cubic structure (fcc), the body centered cubic structure (bcc), and a recently discovered distorted bcc-structure (I-43d).³⁸ In order to obtain a neural network that reliably distinguishes between these phases in all regions of the phase diagram where they are at least metastable, a training set including all typical local configurations and the fluctuations around them is required. We have generated such a training set by carrying out molecular dynamics simulations for various densities and temperatures both in the *NVT* and *NPT* ensembles with periodic boundary conditions and a cutoff of 2.6 for the Lennard-Jones pair interaction. The dynamics of the system was followed with time-reversible integrators using a Nosé-Hoover chain thermostat^{39,40} and a time step of $\Delta t = 5 \times 10^{-3}$. In the simulations in the *NPT* ensemble, we only allowed the rectangular simulation box to fluctuate isotropically, i.e., all box vectors are scaled equally keeping their angle fixed. The system sizes were chosen such that a perfect crystal of the desired structure fits in the simulation box. In particular, these particle numbers are $N_{\text{hcp}} = 1440$, $N_{\text{fcc}} = 1372$, $N_{\text{bcc}} = 1458$, $N_{\text{I-43d}} = 2000$, and $N_{\text{liq}} = 1450$. For each phase, we computed roughly 20 trajectories of length $\tau = 1.5 \times 10^3$ at different pressures and temperatures. To obtain an accurate neural network it has proven particularly important to train the network with information obtained for a broad range of pressures/densities. Here, we use configurations obtained at $T = 0.15$ and $T = 0.92$ and pressures ranging from $P = -5$ to $P = 15$.

From these molecular dynamics simulations we picked 30 000 local configuration for each phase and to avoid correlations between them we used only every 100th MD step. From each configuration we extracted only a few local environments to further reduce correlations. Each local environment is then assigned the structure of the simulated phase. The assigned structure is encoded in the structure vector as described in Sec. II. If, for instance, the assigned structure is hcp, the corresponding structure vector is given by $\tilde{y} = \{0.9, -0.9, -0.9, -0.9, -0.9\}$, and other structures correspond to a different positive component. Note that the value of 0.9 rather than 1.0 is selected, because the activation functions of the neural network are normalized to return values in the interval from -1 to 1 . In total, the training set includes 150 000 local configurations with respective structure vectors. This particular way of assigning the structure type to the local configurations implicitly assumes that during the entire length of the molecular dynamics simulation there is no phase change, not even locally or transiently. We have verified, both by visual inspection and by monitoring the energy and other quantities, that indeed no phase transitions take place in our simulations. From the same MD simulations we also extracted a test set of 22 000 configurations in total.

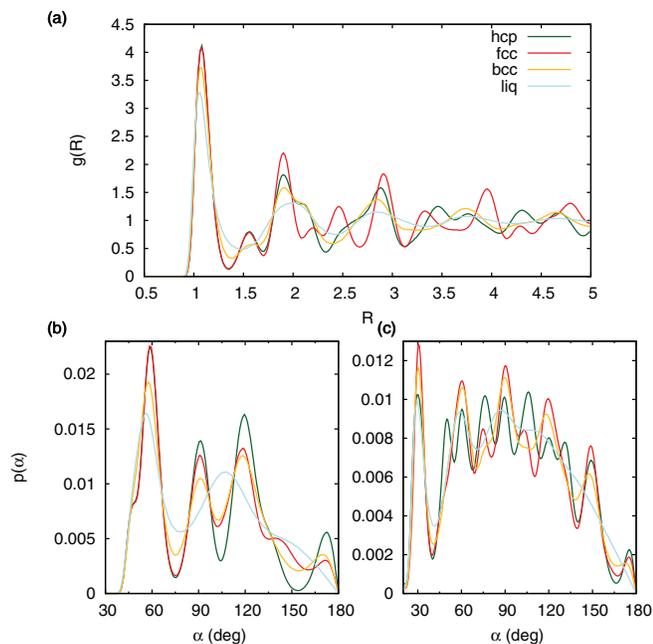


FIG. 4. Several spatial distribution functions for different phases of the Lennard-Jones system. (a) Radial distribution functions $g(R)$, (b) angular distribution functions for particles within a cutoff of $R_{\max} = 1.5$, (c) angular distribution functions in the shell with distance between $R_{\min} = 1.5$ and $R_{\max} = 2.42$. Simulations of 3×10^4 equilibration steps and 3×10^4 production steps were carried out in the NPT ensemble at $T = 0.92$ and $P = 5.68$.

2. Symmetry functions

An accurate neural network for the recognition of local configurations relies on symmetry functions that carry sufficient information on the geometric features of local atomic arrangements. Appropriate symmetry functions can be constructed based on the distributions of distances and angles in the local environment of a given atom as depicted in Fig. 4. The positions and widths of the maxima and minima of the radial distributions functions shown in panel (a) guide the selection of the parameters governing the radial part of the symmetry functions (in particular the symmetry functions of types 2 and 3). For instance, a symmetry function sensitive to particles at distance $R = 1.55$ should provide information to distinguish between hcp and fcc from liquid and bcc. Similarly, symmetry functions responsive to distances of $R = 1.9$ should carry important information because the radial distribution functions of all crystalline phases have a maximum at this position, which is also the distance at which the local environments of fcc and hcp start to differ. The angular distributions of the first and second shell around the central atoms, shown in panels (b) and (c) of Fig. 4, supply useful information for the definition of the angular parts of the symmetry functions. In the first shell, angles $\alpha = 115^\circ$, 120° , and 150° appear to be important while in the second shell the angles $\alpha = 50^\circ$, 80° , 130° , and 115° should be considered. Based on such considerations, we have defined a set of 45 symmetry functions designed to detect the geometrical features of the various structures. The parameters of these symmetry functions are listed in Table I. For the cutoff function given in Eq. (13) we have used $\epsilon_c = 0.2\sigma$ and $\alpha_c = 30\sigma^{-1}$.

TABLE I. Parameters of the symmetry functions for the Lennard-Jones system. The cutoff parameters used in this case are $R_c = 2.6\sigma$, $\alpha_c = 30\sigma^{-1}$, and $\epsilon_c = 0.2\sigma$.

G_2	$\eta (\sigma^{-2})$	$R_s (\sigma)$			
1	1.38889	1.310			
2	2.00000	1.820			
3	3.12500	1.745			
4	3.12500	2.180			
5	4.08163	1.970			
6	8.00000	1.940			
7	8.00000	2.030			
8	8.00000	2.075			
9	12.5000	1.115			
10	12.5000	1.880			
11	12.5000	2.030			
12	12.5000	2.570			
13	22.2222	2.090			
14	50.0000	2.150			
15	50.0000	2.375			
16	50.0000	2.600			
17	200.000	2.300			
18	200.000	2.510			
19	800.000	2.420			
G_3	$\kappa (\sigma^{-1})$				
20	0.5				
21	1.5				
22	2.0				
23	3.5				
24	4.5				
25	6.5				
26	8.5				
27	10.5				
28	12.0				
29	14.5				
30	18.0				
G_6	$\eta (\sigma^{-1})$	$\zeta (1)$	$\lambda (1)$	$\mu (\sigma)$	
31	2.50000	7	-1	1.95	
32	2.85714	16	-1	1.10	
33	3.33333	12	1	2.00	
G_7	$\alpha (1)$	$\eta (1)$			
34	0.2	9.0			
35	0.6	8.4			
36	0.6	4.8			
37	0.8	9.9			
38	0.9	9.0			
39	1.3	7.8			
40	1.4	9.6			
41	2.0	9.6			
42	2.3	8.7			
43	2.7	8.7			
44	2.8	13.2			
45	2.9	18.0			

3. Neural network training

Based on the training set and the symmetry functions described above, we have determined the weights and bias values for a $45 \times 35 \times 35 \times 5$ network using the Kalman filtering technique. In this learning procedure, which takes a few days of computation time using a highly parallelized CUDA⁴¹

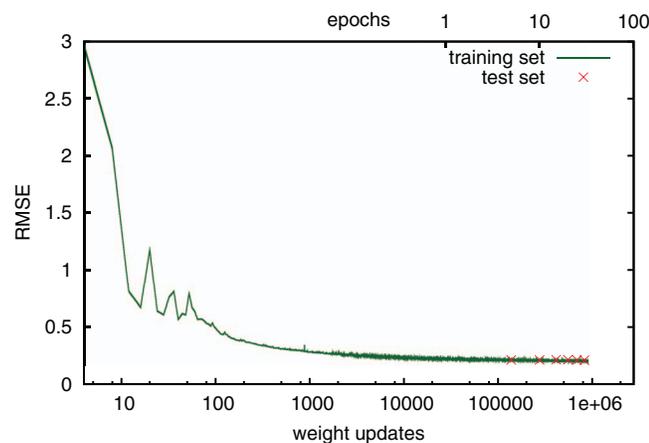


FIG. 5. Typical learning curve of a neural network used to detect different phases in the Lennard-Jones system. The root mean square error (RMSE) between the predicted and target structure vectors of the training set (green curve) and the test set (red crosses) are shown as functions of the number of parameter updates in the Kalman filter learning procedure. The number of epochs, i.e., the number of times the procedure has looped over the entire training set, is displayed on the x -axis at the top.

program running on a cluster of GPUs (Graphics Processing Units), a total of 3050 parameters are optimized. The number of output nodes is given by the number of phases one would like to distinguish and the number of input nodes is equal to the number of input functions. We selected the number of hidden nodes to be of the same order of the number of input nodes. We have also tried neural networks with different numbers of hidden layers and different numbers of nodes in these layers, but have found that the topology given above results in the most accurate structure prediction.

In the Kalman filtering technique we used for the learning procedure, the parameters of the neural network are adapted iteratively. The iteration is stopped when the root mean square error (RMSE) between the predicted and the target structure vectors does not change by more than a certain threshold in subsequent iterations, or epochs, of the Kalman filtering procedure. The iteration is also stopped if over-fitting takes place, which may occur if due to the large number of parameters the learning algorithm tries to generate a unique path through the network for each entry in the training set, instead of doing this for a class of entries. One knows that over-fitting takes place, if the RMSE of the training set continues to decrease in the course of the iteration while the RMSE for the test set increases. A typical learning curve of the neural network, i.e., the RMSE as a function of the number of parameter updates, is shown in Fig. 5. As can be inferred from the figure, the RMSE converges to a finite value rather than to zero due to the selected symmetry functions, which do not contain sufficient information for a perfect structure assignment. The RMSE of the test set and the training set is essentially identical, i.e., no over-fitting occurs.

4. Structure detection

Results for the structure assignment carried out with the neural network for the Lennard-Jones system are shown in Fig. 6. The color-coded bars in this figure indicate the frac-

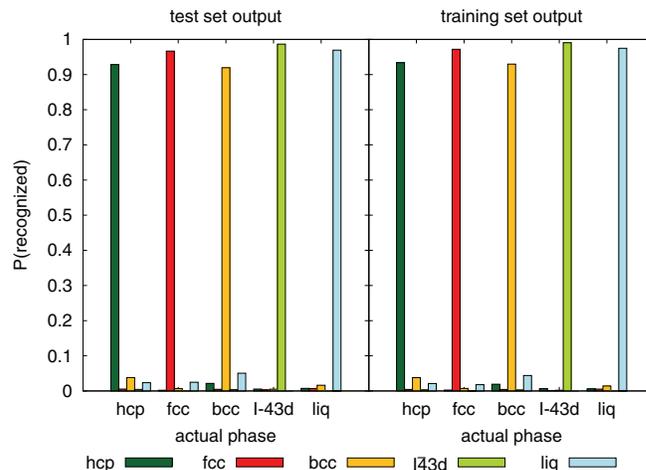


FIG. 6. Distribution of structures assigned by the neural network for the test set (left) and the training set (right) for the Lennard-Jones system. For each of the different phases indicated on the x -axis, the heights of the color-coded bars correspond to the fractions of structures that have been assigned the type specified by the color at the bottom.

tion of structures that have been assigned a particular type. The colors of the bars specify to which type a structure is assigned and the labels on the x -axis indicate the phase from which the structure is taken. As can be inferred from the figure, the results obtained for the test set (left) are of the same quality as the results for the training set (right), demonstrating that no over-fitting has occurred in training the network. For all phases, structures are assigned with an accuracy better than 90%. The fraction of misassigned structures is particularly small for structures from the liquid phase and the I-43d phase, while for the hcp and bcc phases a few percent of the structures are assigned the wrong type, probably because at lower densities the local structures can deviate considerably from the perfect crystalline arrangement due to thermal fluctuations.

For comparison, we have also determined local structure types for the Lennard-Jones system using Steinhardt bond order parameters constructed from spherical harmonics.¹² More specifically, we use the bond order parameters \bar{q}_4 and \bar{q}_6 obtained from the Steinhardt bond order parameters q_4 and q_6 by averaging over the first neighbor shell.¹⁵ As shown previously, these order parameters provide a practical and accurate way to distinguish between all phases occurring in the Lennard-Jones system. A scatter plot of the order parameters \bar{q}_4 and \bar{q}_6 computed for the configurations in the training set is shown in the left panel of Fig. 7. The dots in the scatter plot are colored according to the phase from which the respective configurations were taken. To distinguish between the various structures, we divided the (\bar{q}_4, \bar{q}_6) -plane in regions as denoted by the lines in Fig. 7. Each configuration is then simply assigned the structure corresponding to the region in which the respective (\bar{q}_4, \bar{q}_6) -pair is located. (An alternative way to assign structure type consists in determining the frequency at which (\bar{q}_4, \bar{q}_6) -pairs occur in the various phases and then selecting the phase with the highest frequency.⁵) Results of the structural analysis carried out in this way for the training set are shown in the right panel of Fig. 7. While liquid

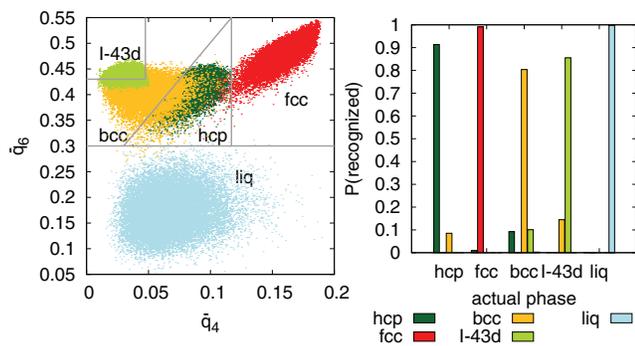


FIG. 7. Left: Scatter plot in the (\bar{q}_4, \bar{q}_6) -plane of local structures taken from the different phases of the Lennard-Jones system. The lines delimit the regions used for the structure assignment. Right: Distribution of structures assigned based on \bar{q}_4 and \bar{q}_6 . For each of the different phases indicated on the x -axis, the heights of the color-coded bars correspond to the fractions of structures that have been assigned the type specified by the color at the bottom.

and fcc structures are recognized very accurately, the structure assignment is less accurate for hcp and, in particular, for bcc and I-43d structures. Thus, the order parameters \bar{q}_4 and \bar{q}_6 do not contain sufficient information on these structures as is also evident from the pronounced overlap of the hcp, bcc, and I-43d phases in the (\bar{q}_4, \bar{q}_6) -plane. In comparison to the structure assignment based on \bar{q}_4 and \bar{q}_6 , the neural network yields a roughly uniform accuracy for all phases with a higher precision on the average.

B. Water and ice

As shown in Sec. III A 4, local bond order parameters based on spherical harmonics perform well in detecting the various simple crystalline structures such as those occurring in the Lennard-Jones systems. They are less accurate, however, when applied to more complex structures such as liquid water and the various phases of ice.¹⁷ While modifications of these order parameters have been successfully used to distinguish hexagonal and cubic ice (ice Ih and ice Ic) from liquid water,^{21,42} the high pressure phases of ice (ice II, III, ...) are not separated sufficiently in this projection. Distributions of the averaged Steinhardt bond order parameters \bar{q}_4 , \bar{q}_6 , \bar{q}_8 , and \bar{q}_{10} calculated for the various phases of ice, and shown in the right panel of Fig. 8, display a pronounced overlap. If spread out in the two-dimensional (\bar{q}_4, \bar{q}_6) -plane, the distributions for ice Ih, ice Ic, and liquid water separate (see left panel of Fig. 8). Hence, these three phases can be distinguished from each other based on \bar{q}_4 and \bar{q}_6 , as observed recently by Sanz *et al.*²¹ The distributions for the ice II, III, and V, however, do not separate such that the averaged order parameters do not provide sufficient information to discriminate between these high pressure phases. Note that to generate the plots of Fig. 8 we have used configurations obtained for only a few temperatures and pressures, thus the overlap of the distributions becomes even stronger if wider regions in the P - T -plane are considered. One reason for the inability of the Steinhardt bond order parameters to distinguish between the high pressure forms of ice is that these phases include oxygen atoms with different spatial environments. The unit cell of ice III, for instance,

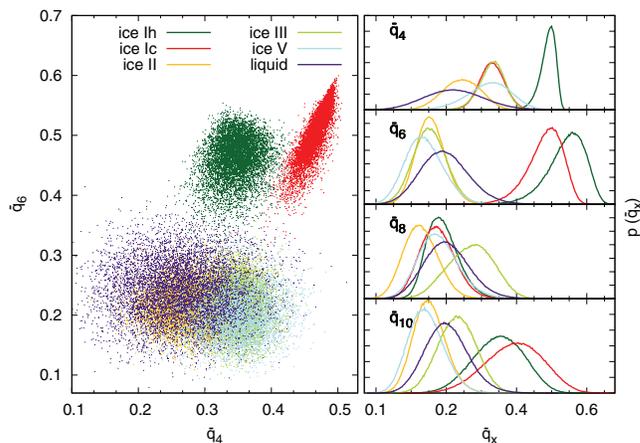


FIG. 8. (Left) Scatter plot in the (\bar{q}_4, \bar{q}_6) -plane for structures taken from liquid water and from different phases of ice. To calculate the bond order parameters only the positions of the oxygen atoms were taken into account. All results are obtained from NVT simulations carried out at $T = 270$ K and densities $\rho_{\text{Ih}} = \rho_{\text{Ic}} = 910$ kg/m³, $\rho_{\text{II}} = 1145$ kg/m³, $\rho_{\text{III}} = 1115$ kg/m³, $\rho_{\text{V}} = 1210$ kg/m³, and $\rho_{\text{liq}} = 985$ kg/m³. (Right) Distributions of \bar{q}_4 , \bar{q}_6 , \bar{q}_8 , and \bar{q}_{10} obtained for the same systems.

features two distinct oxygen positions and for ice V there are even four unique oxygen positions.

In an effort to enhance the recognition for ice structures, several techniques have been suggested. Chau and Hardwick¹⁹ have developed an order parameter based on detecting the occurrence of local tetrahedral structures in ice. (A scaled version, which is defined in the range 0 to 1, is given in Ref. 20.) This method is very accurate for the distinction of ice Ih or ice Ic from liquid water, but fails for the higher density forms of ice. Similarly, the maximum projection method of Brukhno *et al.*¹⁷ can be used to distinguish between liquid water, and hexagonal and cubic ice only. Here, we demonstrate that an appropriately trained neural network can be used to distinguish locally between liquid water and several phases of ice (Ih, Ic, II, III, and V) over wide ranges of pressure and temperature.

1. Training set

All calculations for ice and liquid water are carried out with the TIP4P/Ice model,⁴³ which is fitted to experimental properties of low and high density ice. In this model, each water molecule is represented by four sites rigidly connected with each other and interacting pairwise via Coulomb and Lennard-Jones potentials. Polarization effects are neglected. The TIP4P/Ice model reproduces all ice phases consisting of intact water molecules and leads to a phase diagram with the correct topology and coexistence lines that are only slightly displaced with respect to the experimental phase diagram.⁴⁴ The model also accurately reproduces the densities of the ice phases and of the liquid phase.

Using molecular dynamics simulations, we have generated training and test sets that consist of configurations for the six studied water phases over a wide range of temperatures and densities, such that configurations of highly metastable structures are also included. We have carried out simulations in the NVT and isotropic NPT ensembles integrating the equation of motion with a slightly modified

version of the Verlet-like algorithm proposed by Kamberaj *et al.*,⁴⁵ based on the Trotter decomposition schemes applied by Miller *et al.*⁴⁶ and Martyna *et al.*³⁹ In these integration schemes, the canonical and isothermal-isobaric ensembles are implemented through thermostat chains based on the Nosé-Hoover^{47,48} and the Andersen⁴⁹ approaches. Long range electrostatic interaction are treated using Ewald summation with 1152 k -space vectors.

For each phase, we have generated molecular dynamics trajectories for different pressures and densities encompassing the entire range of mechanical stability. System sizes were chosen such that the simulation box accommodates a defect-free crystal of the respective type. In particular, molecule numbers of $N_{\text{Ih}} = 896$, $N_{\text{Ic}} = 1000$, $N_{\text{II}} = 960$, $N_{\text{III}} = 768$, $N_{\text{V}} = 756$ and $N_{\text{liq}} = 940$, were used for ice Ih, ice Ic, ice II, ice III, ice V, and the liquid, respectively. Initial conditions were constructed for each crystalline phase using the space group of a cell, its cell parameters (side length and angles) and the fractional coordinates taken from literature. Particular care has to be taken to generate the proton order/disorder appropriate for each ice phase. While the proton ordered crystals are simply produced using the fractional coordinates for all oxygen and hydrogen atoms provided in the literature, the disordered phases have to be generated in a way that obeys the Bernal-Fowler ice rules.⁵⁰ Here, we have used the method developed by Ayala and Tchijov⁵¹ to generate permitted hydrogen atom disorder of proton ordered arrangements. Starting from a perfect proton ordered crystal, hydrogen atoms are shifted along closed loops of neighboring H-bonded molecules. In order to reach a sufficient degree of disorder, this procedure is repeated for a minimum number of iterations and until a zero net dipole moment is achieved.

The proton ordered hexagonal ice cells were taken from Ref. 52. From this arrangement, disordered ice Ih was generated by applying the method described above. Cubic ice Ic crystals were generated according to Lekner.⁵³ The proton ordered ice II crystals were generated according to Ref. 54. The atomic coordinates and hydrogen site occupancies for ice IX, the proton ordered counter part of ice III, were taken from Refs. 55 and 56. And finally, ice V/XIII were set up according to Refs. 56 and 57.

Each molecular dynamics trajectory was 100 ps long and every 0.1 ps the environmental information of a randomly chosen oxygen atom was stored and used to generate an entry for the training or test set. A total of 100 trajectories were computed yielding 10^5 configurations. The training and test sets were then generated by computing the symmetry functions for each configuration and assigning to it the structure type of the simulated phase. This way to construct the training and test set relies on the assumption that during the molecular dynamics simulation the entire sample remains in the original phase and defects or excursions to other phases do not occur even transiently.

2. Symmetry functions

To define symmetry functions capable of capturing the important details of local structures we have computed radial

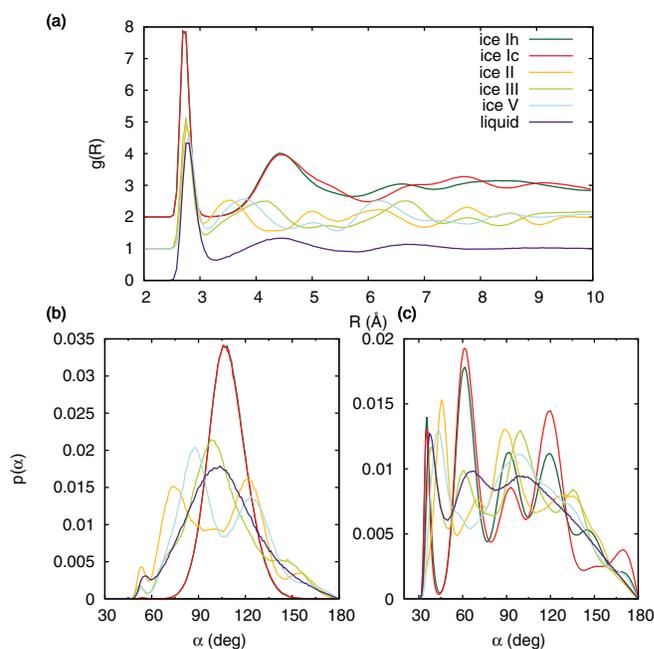


FIG. 9. Radial and angular distribution functions for different phases of water. (a) Oxygen-oxygen radial distribution functions $g_{\text{OO}}(R)$ for low density and high density forms of ice as well as liquid water. All curves except the one for liquid water are shifted for better readability. (b) Distributions of the OOO-angle for atoms within a distance of $R_{\text{cut}} = 3.25$ Å from the central atom. (c) Distributions of the OOO-angle for atoms within the shell $3.5 \text{ \AA} \leq R \leq 5.5 \text{ \AA}$ around the central atom. All curves were determined in NVT simulations at temperatures $T_{\text{Ih}} = T_{\text{Ic}} = 210$ K, $T_{\text{II}} = T_{\text{III}} = T_{\text{liq}} = 270$ K and $T_{\text{V}} = 240$ K. For the densities the values $\rho_{\text{Ih}} = \rho_{\text{Ic}} = \rho_{\text{liq}} = 1000$ kg/m³ and $\rho_{\text{II}} = \rho_{\text{III}} = \rho_{\text{V}} = 1200$ kg/m³ were chosen.

distribution functions as well as the distributions of OOO-angles for all phases considered here. The results of these calculations are shown in Fig. 9. As one can infer from Fig. 9(b), for distances smaller than 3.25 \AA the angles $\alpha = 50^\circ, 72^\circ, 83^\circ, 105^\circ, 125^\circ$, and 154° appear to be useful symmetry function parameters, whereas in the shell of distances between 3.5 \AA and 5.5 \AA $\alpha = 45^\circ, 61^\circ, 120^\circ, 136^\circ$ are good candidates. Based on these distribution functions, we have defined the symmetry functions listed in Table II. For these symmetry functions we have used a neighbor cutoff of $R_c = 6 \text{ \AA}$ and the parameters $\epsilon_c = 0.2 \text{ \AA}$ and $\alpha_c = 30 \text{ \AA}^{-1}$, respectively. We have also prepared a set of symmetry function parameters for a much larger cutoff of $R_c = 8.234 \text{ \AA}$. In both cases, only the positions of the oxygen atoms are used to compute the symmetry functions. Including also hydrogen positions or, equivalently, information on dipole orientations, may improve the accuracy of the detection further.

3. Neural network training and structure detection

Using the training set and the symmetry functions described in Sec. III B 2 we have trained two different neural networks to distinguish local structures of ice Ih, Ic, II, III, V, and of liquid water. The two networks differ in the network topology, $35 \times 30 \times 30 \times 6$ and $39 \times 35 \times 35 \times 6$, and also in the cutoff radius, $R_c = 6 \text{ \AA}$ and $R_c = 8.234 \text{ \AA}$. With these

TABLE II. Parameters of the symmetry functions for water and ice modeled with the TIP4P/Ice potential for the cutoff radii $R_c = 4.4335 \text{ \AA}$, $R_c = 6.0 \text{ \AA}$, and $R_c = 8.234 \text{ \AA}$. The other cutoff parameters are $\alpha_c = 30 \text{ \AA}^{-1}$ and $\epsilon_c = 0.2 \text{ \AA}$. For each of the three cutoff radii R_c we used the same parameters, but did not use all of the symmetry functions. In particular, for $R_c = 4.4335 \text{ \AA}$ we excluded symmetry functions 2, 3, 5, 8, 16, 39, and 49, for $R_c = 6.0 \text{ \AA}$ we excluded symmetry functions 10–12, 14, 17, 19, 28, 29, 32, and 36–40, and for $R_c = 8.234 \text{ \AA}$ we excluded symmetry functions 10–12, 14, 17, 19, 28, 29, 32, 36–38, and 40.

G_2	$\eta (\text{\AA}^{-2})$	$R_s (\text{\AA})$	G_3	$\kappa (\text{\AA}^{-1})$	
1	0.049857	2.533440	20	0.157888	
2	0.138492	2.533440	21	0.315776	
3	0.406998	3.103464	22	0.473664	
4	0.553970	3.768492	23	0.631552	
5	0.797717	6.143592	24	1.420993	
6	1.246432	2.960958	25	1.736769	
7	1.246432	4.148508	26	1.894657	
8	1.246432	5.241054	27	2.526209	
9	2.215880	2.960958	28	3.947202	
10	2.215880	3.198468	29	4.262978	
11	2.215880	3.958500			
12	2.215880	4.14850			
13	4.985730	2.770950			
14	4.985730	3.055962			
15	4.985730	3.293472			
16	4.985730	4.766034			
17	19.94292	2.913456			
18	19.94292	3.103464			
19	19.94292	3.768492			
G_4	$\eta (\text{\AA}^{-2})$	$\zeta (1)$	$\lambda (1)$		
30	0.031578	10.0	-1		
31	0.031578	10.0	1		
32	0.045111	4.5	-1		
33	0.052629	6.0	-1		
34	0.083099	5.5	-1		
35	0.143535	14.0	1		
36	0.185751	3.75	-1		
37	0.210517	6.50	1		
38	0.242905	2.5	-1		
39	0.332396	1.4	-1		
G_8	$\alpha (1)$	$\eta (1)$	$a_l (\text{\AA})$	$a_r (\text{\AA})$	$\nu (\text{\AA}^{-1})$
40	0.38	0.2350	2.00	3.15	5
41	1.22	0.1375	2.00	3.15	5
42	0.35	0.1000	2.00	6.00	5
43	0.75	0.2275	2.00	6.00	5
44	0.63	0.1825	2.00	6.00	5
45	0.70	0.1000	3.00	4.00	5
46	0.58	0.1000	3.15	6.00	5
47	1.05	0.1075	3.15	6.00	5
48	0.63	0.1825	3.15	6.00	5
49	1.05	0.1000	4.00	6.00	5

cutoffs the local regions encompass two and three neighbor shells, respectively. Results obtained with these two neural networks are shown in Fig. 10. For the smaller cutoff the detection accuracy is between 85% for ice V and 98% for ice III. With the larger cutoff the detection accuracy is nearly 100% for all phases. The spatial resolution, however, is reduced with the larger cutoff.

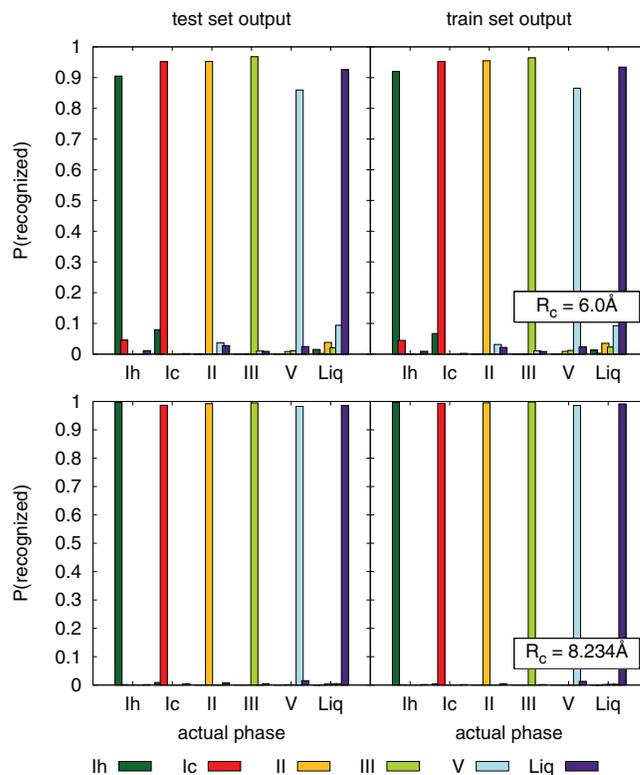


FIG. 10. Distribution of structures assigned by the neural network for the test set (left) and the training set (right) for liquid water and ice Ih, Ic, II, III, and V. For each of the different phases indicated on the x-axis, the heights of the color-coded bars correspond to the fractions of structures that have been assigned the type specified by the color at the bottom. Results were obtained for a cutoff of $R_c = 6 \text{ \AA}$ (top) and for a cutoff of $R_c = 8.234 \text{ \AA}$ (bottom).

4. Crystallization of supercooled water

In order to demonstrate the practical applicability of the neural network for structure detection, we have used it to study the freezing of supercooled water to hexagonal ice. The free energetics of this nucleation process has been studied previously using metadynamics⁵⁸ and umbrella sampling.^{16,17,59} Here, we determine the free energy of the system using the size n of the largest crystalline cluster as order parameter. To compute this order parameter for a particular configuration, first all water molecules are assigned a structure type by applying the neural network. Then, crystalline molecules are grouped into connected clusters and the order parameter is given by the number n of molecules in the largest of these crystalline clusters. Two molecules are considered to be connected if their oxygen atoms are closer than 3.8 \AA .

In these simulations we have used a neural network for structure detection trained specifically to distinguish only liquid water from ice I, without discriminating between ice Ih, and ice Ic. Furthermore, a cutoff radius of $R_c = 4.43352 \text{ \AA}$ was used, implying that only information from the first neighbor shell was included in computing the symmetry functions. In order to improve the accuracy of the structure detection near the surface of the crystalline nucleus, we have also included crystalline nuclei of cubic and spherical shape embedded in liquid water, as well as liquid/solid slab geometries in our training set. We generated these configurations fixing a

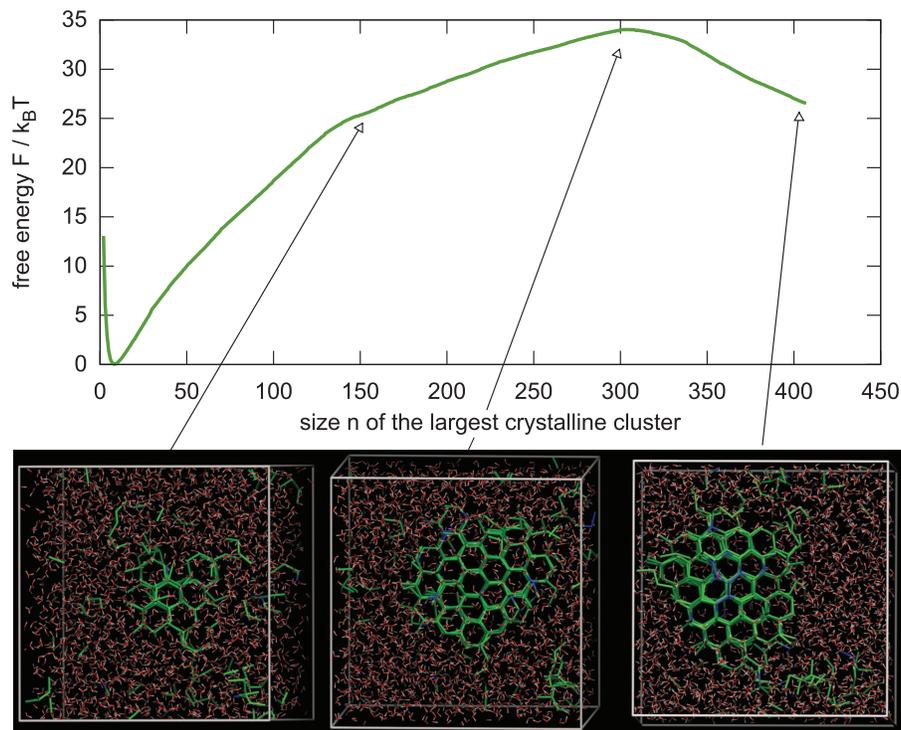


FIG. 11. Free energy $\beta F(n)$ as a function of the size n of the largest crystalline cluster for a system of $N = 2880$ TIP4P/Ice water molecules at temperature $T = 235$ K and pressure $P = 1$ bar corresponding to 13% undercooling. The pictures at the bottom show snapshots of the system in different stages of the transition from liquid water to hexagonal ice, ice Ih. Crystalline hexagonal and cubic molecules and the bonds between them are shown in green and blue, respectively. Note that all crystalline molecules are shown and not only those belonging to the largest crystalline cluster. The orientation of the each simulation box is such that the visibility of the crystalline structure is enhanced.

fraction of the molecules to their respective positions in the perfect lattice using harmonic springs. Then we heated the system to melt the mobile molecules and subsequently relaxed it at 235 K with removed springs. Note that for the analysis, however, we have used a neural network that can distinguish between hexagonal and cubic ice.

To determine the free energy as a function of the size n of the largest crystalline cluster, we have used an umbrella sampling procedure with overlapping windows with hard walls. We study a system consisting of $N = 2880$ molecules at temperature $T = 235$ K and pressure of $P = 1$ bar, corresponding to 13% supercooling with respect to the melting temperature $T_{\text{melt}} = 270 \pm 3$ K of TIP4P/Ice model.⁶⁰ We chose this particular temperature such that the critical nucleus is of a size that fits in the simulation box. For this temperature, the growth rate of the crystalline nucleus is still sufficiently high such that the simulations are not hindered by the sluggish dynamics of water at low temperatures.

For each window, corresponding to a certain range of the order parameter, we have sampled NPT configurations using the hybrid Monte Carlo (HMC) method,⁶¹ in which new configurations are generated by performing NVE molecular dynamics simulations. These short dynamical trajectories are computed without bias and the constraints on the order parameter imposed by the windows are taken into account only in the Monte Carlo acceptance step. We have integrated the equations of motion with a modified version of the algorithm of Miller *et al.*,⁴⁶ which follows the algorithm of Omelyan.⁶² This algorithm requires two force evaluations per integration

step and exactly conserves phase space volume as required by the hybrid Monte Carlo scheme. Five molecular dynamics steps of length $\Delta t = 7.2$ fs are carried out for each short trajectory, yielding an acceptance probability of roughly 50%. To improve the sampling efficiency of the simulation we also carry out exchanges of configuration between adjacent windows following the replica exchange procedure of Auer and Frenkel.¹⁴ We have carried out 5×10^6 equilibration HMC steps and 10^7 production HMC steps in each of the 40 windows. Histograms of the order parameter obtained from the each window separately are combined with the self-consistent histogram method² and the free energy is then obtained as the logarithm of the resulting distribution.

The free energy $F(n)$ as a function of the size n of largest crystalline cluster obtained from our simulations is shown in Fig. 11. The shape of this free energy profile with a barrier separating the supercooled state for small values of n from the completely crystalline state at large values of n is typical for a phase transition proceeding via nucleation and growth. In the picture of classical nucleation theory, which provides a qualitative description of nucleation processes, the barrier is due to the free energetic cost required to form an interface between the crystal nucleus and the surrounding metastable phase. Only after the crystalline nucleus reaches the so called critical size does the lower free energy of the crystalline phase prevail and the free energy decreases from this size on. This qualitative picture of the mechanism for crystallization is confirmed by the snapshots in the bottom row of Fig. 11, showing the system at different stages of the crystallization process.

Note that the kink in the free energy profile visible at about $n = 140$ is most likely the result of insufficient sampling possibly due to slow shape equilibration.

Our calculations, carried out at $T = 235$ K and pressure $P = 1$ bar, yield a barrier height of $34 k_B T$ and a critical crystalline nucleus consisting of $n \sim 300$ molecules. By running molecular dynamics simulations starting from configurations with sizes $n = 110, 130,$ and 240 we confirmed that the crystalline clusters tended to shrink as expected for subcritical nuclei.²¹ In trajectories started from configurations with clusters of sizes $n = 400$ and 415 , i.e., beyond the free energy barrier, the cluster size evolved very slowly such that the system did not commit to the completely liquid or fully crystallized state on the nanosecond time scale of our simulations. For the system sizes studied here we could not go to larger initial crystallites, for which the complete crystallization is expected to be faster. Note that the critical size obtained from our free energy calculation is smaller than the critical cluster size of 615 molecules found by Li *et al.*⁴² at 235 K for the mW model, which has a melting point of 274.6 K.⁶³ These critical nuclei were identified from a dynamical criterion based on committor calculations such that no information on the height of the free energy barrier is available in this case. This discrepancy in the size of the critical nucleus might be due to difference in the water model, specifically in the melting point, the surface tension, and in the heat of melting, but also to the different ways used to detect crystallinity.

Previous computer simulation studies carried out by Brukhno *et al.*¹⁷ as well as Li *et al.*⁴² have indicated that during the formation of hexagonal ice the crystalline nucleus may include regions of cubic ice. In contrast, our simulations yielded only crystalline nuclei with purely hexagonal structure. Single molecules with cubic environments were found only on the surface of the nucleus as can be seen in Fig. 11.

IV. SUMMARY AND CONCLUSIONS

In summary, we have developed an artificial neural network for the detection and the classification of local atomic structures. Different classes of atomic arrangements are distinguished based on a set of symmetry functions that capture the essential structural features and are invariant with respect to translations, rotations, and permutations of identical atoms. These symmetry functions are fed into the input layer of the neural network, which then performs a series of summations and nonlinear operations that finally yield an output vector, in which each component corresponds to one of the structures to be recognized. Typically, our neural networks have 30–40 input nodes (one for each symmetry function) and 2 hidden layers with about 40 nodes each. They depend on a few thousand parameters that provide the flexibility required to adapt the neural network to a variety of different structures. The parameters are tuned during a learning phase, in which the neural network is trained based on a training set consisting of several tens of thousands of local atomic configurations with known structure types. Such a training set, and an independent test set used to assess the quality of the structure recognition, can be prepared with molecular dynamics or

Monte Carlo simulations performed in the pure phases. The neural networks developed here can be used to analyze the results of molecular dynamics and Monte Carlo simulations *a posteriori*, but also to drive such simulations to important but rarely visited regions of configuration space. Applying such a bias is, for instance, useful in studying nucleation phenomena that involve rare barrier crossing events. By applying the neural network method to the Lennard-Jones system as well as to water/ice we have demonstrated that this approach works well not only for simple phases but also for complex, open structures for which conventional structure recognition methods fail. Note, however, that the neural network developed here is designed for the classification of structures into known structure types rather than for the discovery of new structures.

In the development of a neural network for structure recognition an appropriate definition of the symmetry functions is the most crucial step and the performance of the method strongly depends on the quality of the symmetry functions. The symmetry functions need to be designed such that they provide sufficient information to distinguish the various structures. Structural features gleaned from distributions of distances and angles can be useful in this process. While in this work we have used only symmetry functions of the type developed by Behler and Parrinello for energy calculations,^{22,23} the structure recognition may be enhanced by adding also other structural fingerprints such as the Steinhardt bond order parameters to the input for the neural network. In the application of the neural network the computation of the symmetry functions is usually the computationally most time-consuming step while the processing done by the neural network is relatively inexpensive. Therefore, it is advantageous to keep the number of symmetry functions as low as possible. A sensitivity analysis, which determines how strongly the output of a network depends on a particular input, may help to eliminate redundant symmetry functions without sacrificing the accuracy of the structure detection.

Another important factor in the development of a neural network for structure detection is the generation of an appropriate training set. This set of configurations needs to include arrangements similar to all structures to which the neural network is eventually applied. Since the neural network is nothing else than a complicated fitting function, atomic structures that differ markedly from all structures included in the training set will not be assigned correctly. In practice, this implies that structures for the training set must be collected from simulations carried out over a broad range of pressures and temperatures. We have also found that it is important that configurations included in the training set are statistically independent from each other.

The range within which atoms are considered for the calculation of the symmetry function influences both the accuracy as well as the spatial resolution of the structure recognition. While including structural information from a large region around the central atom increases the accuracy of the method, it reduces its resolution and leads to unwanted averaging effects particularly near inhomogeneities such as interfaces and defects. Thus, particularly in the study of

nucleation phenomena where interfaces between different phases play an important role, it is advisable to choose cut-offs that do not exceed those of other detection methods typically including atoms up to the second neighbor shell.^{13–15} The general method developed here may be applied also to detect defects such as interstitials, vacancies, and dislocations, in which case a small cutoff is particularly important.

As mentioned in Sec. III B 4, in the example of the crystallization of water we have improved the accuracy of the structure detection near the solid-liquid interface by adding configurations containing such an interface to the training set. These training configurations were generated by fixing molecules in crystalline regions of spherical, cubic, or slab-like geometry to their lattice site, while equilibrating the unconstrained liquid surrounding the crystalline parts. While here we have used this scheme only to improve the distinction between molecules with liquid and crystalline environments, one could use configurations containing interfaces also to specifically train the neural network to recognize typical interface structures and identify molecules belonging to the interface layer.

ACKNOWLEDGMENTS

This work was supported by the Austrian Science Fund (FWF) within the SFB ViCoM (Grant No. F 41) and Project P22087-N16. We thank Georg Menzl and Andreas Singraber for useful discussions. All simulations were carried out on the Vienna Scientific Cluster (VSC).

¹G. M. Torrie and J. P. Valleau, *J. Comput. Phys.* **23**, 187 (1977).

²D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, 2002).

³A. Laio and M. Parrinello, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 12562 (2002).

⁴D. Moroni, P. R. ten Wolde, and P. G. Bolhuis, *Phys. Rev. Lett.* **94**, 235703 (2005).

⁵W. Lechner, C. Dellago, and P. G. Bolhuis, *Phys. Rev. Lett.* **106**, 085701 (2011).

⁶U. Gasser, E. Weeks, A. Schofield, P. Pusey, and D. Weitz, *Science* **292**, 258 (2001).

⁷J. D. Honeycutt and H. C. Andersen, *J. Chem. Phys.* **91**, 4950 (1987).

⁸C. L. Cleveland, W. D. Luedtke, and U. Landman, *Phys. Rev. B* **60**, 5065 (1999).

⁹D. Faken and H. Jónsson, *Comput. Mater. Sci.* **2**, 279 (1994).

¹⁰C. L. Kelchner, S. J. Plimpton, and J. C. Hamilton, *Phys. Rev. B* **58**, 11085 (1998).

¹¹G. J. Ackland and A. P. Jones, *Phys. Rev. B* **73**, 054104 (2006).

¹²P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, *Phys. Rev. B* **28**, 784 (1983).

¹³P.-R. ten Wolde, M. J. Ruiz-Montero, and D. Frenkel, *Faraday Discuss.* **104**, 93 (1996).

¹⁴S. Auer and D. Frenkel, *Adv. Polym. Sci.* **173**, 149 (2005).

¹⁵W. Lechner and C. Dellago, *J. Chem. Phys.* **129**, 114707 (2008).

¹⁶R. Radhakrishnan and B. L. Trout, *J. Am. Chem. Soc.* **125**, 7743 (2003).

¹⁷A. V. Brukhno, J. Anwar, R. Davidchack, and R. Handel, *J. Phys.: Condens. Matter* **20**, 494243 (2008).

¹⁸A. Reinhardt, J. P. K. Doye, E. G. Noya, and C. Vega, *J. Chem. Phys.* **137**, 194504 (2012).

¹⁹P. L. Chau and A. J. Hardwick, *Mol. Phys.* **93**, 511 (1998).

²⁰J. R. Errington and P. G. Debenedetti, *Nature (London)* **409**, 318 (2001).

²¹E. Sanz, C. Vega De Las Heras, J. R. Espinosa, R. Caballero-Bernal, J. L. F. Abascal, and C. Valeriani, *J. Am. Chem. Soc.* **135**, 15008–15017 (2013).

²²J. Behler and M. Parrinello, *Phys. Rev. Lett.* **98**, 146401 (2007).

²³J. Behler, *Phys. Chem. Chem. Phys.* **13**, 17930 (2011).

²⁴S. Lorenz, A. Groß, and M. Scheffler, *Chem. Phys. Lett.* **395**, 210 (2004).

²⁵J. Behler, *J. Chem. Phys.* **134**, 074106 (2011).

²⁶J. M. Zurada, A. Malinowski, and I. Cloete, in *IEEE International Symposium on Circuits and Systems* (IEEE, NY, 1994), pp. 447–450.

²⁷H. L. Viktor, A. P. Engelbrecht, and I. Cloete, in *IEEE International Conference on Neural Networks* (IEEE, NY, 1995), Vol. 4, pp. 1788–1793.

²⁸A. P. Engelbrecht and I. Cloete, in *International Conference on Systems, Signals, Control, Computers* (IAAMSAD and ANS, 1998), Vol. 2, pp. 221–225.

²⁹D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Nature (London)* **323**, 533 (1986).

³⁰M. T. Hagan and M. Menhaj, *IEEE Trans. Neural Netw.* **5**, 989 (1994).

³¹S. Singhal and L. Wu, *Advances in Neural Information Processing Systems* (IEEE, NY, 1989), Vol. 1, pp. 133–140.

³²S. Haykin, *Kalman Filtering and Neural Networks* (Wiley, 2001).

³³F. Heimes, in *IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, NY, 1998), Vol. 2, pp. 1639–1644.

³⁴Y. Zhang and X. Li, *IEEE Trans. Neural Netw.* **10**, 930 (1999).

³⁵D. J. Lary and H. Y. Mussa, *Atmos. Chem. Phys. Discuss.* **4**, 3653 (2004).

³⁶D. Nguyen and B. Widrow, in *IEEE International Joint Conference on Neural Networks* (IEEE, NY, 1990), Vol. 3, pp. 21–26.

³⁷T. Morawietz, Master's thesis, University of Bochum, 2010.

³⁸H. Eshet, F. Bruneval, and M. Parrinello, *J. Chem. Phys.* **129**, 026101 (2008).

³⁹G. J. Martyna, M. E. Tuckerman, D. Tobias, and M. L. Klein, *Mol. Phys.* **87**, 1117 (1996).

⁴⁰M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010).

⁴¹E. Lindholm, J. Nickolls, S. F. Oberman, and J. Montrym, *IEEE MICRO* **28**, 39 (2008).

⁴²T. Li, D. Donadio, G. Russo, and G. Galli, *Phys. Chem. Chem. Phys.* **13**, 19807 (2011).

⁴³J. L. F. Abascal, E. Sanz, R. G. Fernandez, and C. Vega, *J. Chem. Phys.* **122**, 234511 (2005).

⁴⁴C. Vega and J. L. F. Abascal, *Phys. Chem. Chem. Phys.* **13**, 19663 (2011).

⁴⁵H. Kamberaj, R. J. Low, and M. P. Neal, *J. Chem. Phys.* **122**, 224114 (2005).

⁴⁶T. F. Miller III, M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. J. Martyna, *J. Chem. Phys.* **116**, 8649 (2002).

⁴⁷S. Nosé, *J. Chem. Phys.* **81**, 511 (1984).

⁴⁸W. G. Hoover, *Phys. Rev. A* **31**, 1695 (1985).

⁴⁹H. C. Andersen, *J. Chem. Phys.* **72**, 2384 (1980).

⁵⁰V. Petrenko and R. Whitworth, *Physics of Ice* (Oxford University Press, 1999).

⁵¹R. Ayala and V. Tchijov, *Can. J. Phys.* **81**, 11 (2003).

⁵²T. K. Hirsch and L. Ojamäe, *J. Phys. Chem. B* **108**, 15856 (2004).

⁵³J. Lekner, *Physica B* **240**, 263 (1997).

⁵⁴B. Kamb, W. C. Hamilton, S. J. La Placa, and A. Prakash, *J. Chem. Phys.* **55**, 1934 (1971).

⁵⁵S. J. La Placa, W. C. Hamilton, B. Kamb, and A. Prakash, *J. Chem. Phys.* **58**, 567 (1973).

⁵⁶C. Lobban, J. L. Finney, and W. F. Kuhs, *J. Chem. Phys.* **112**, 7169 (2000).

⁵⁷C. G. Salzmann, P. G. Radaelli, A. Hallbrucker, E. Mayer, and J. L. Finney, *Science* **311**, 1758 (2006).

⁵⁸D. Quigley and P. M. Rodger, *J. Chem. Phys.* **128**, 154518 (2008).

⁵⁹A. Reinhardt and J. P. K. Doye, *J. Chem. Phys.* **136**, 054501 (2012).

⁶⁰R. G. Fernández, J. L. F. Abascal, and C. Vega, *J. Chem. Phys.* **124**, 144506 (2006).

⁶¹B. Mehlig, D. W. Heermann, and B. M. Forrest, *Phys. Rev. B* **45**, 679 (1992).

⁶²I. P. Omelyan, I. M. Mryglod, and R. Folk, *Phys. Rev. E* **65**, 056706 (2002).

⁶³V. Molinero and E. B. Moore, *J. Phys. Chem. B* **113**, 4008 (2009).