JOURNAL OF CHEMICAL INFORMATION AND MODELING

Subscriber access provided by UCL Library Services

Computational Chemistry

DeepIce: a Deep Neural Network Approach to Identify Ice and Water Molecules

Maxwell Fulford, Matteo Salvalaglio, and Carla Molteni

J. Chem. Inf. Model., **Just Accepted Manuscript** • DOI: 10.1021/acs.jcim.9b00005 • Publication Date (Web): 15 Mar 2019

Downloaded from http://pubs.acs.org on March 19, 2019

Just Accepted

"Just Accepted" manuscripts have been peer-reviewed and accepted for publication. They are posted online prior to technical editing, formatting for publication and author proofing. The American Chemical Society provides "Just Accepted" as a service to the research community to expedite the dissemination of scientific material as soon as possible after acceptance. "Just Accepted" manuscripts appear in full in PDF format accompanied by an HTML abstract. "Just Accepted" manuscripts have been fully peer reviewed, but should not be considered the official version of record. They are citable by the Digital Object Identifier (DOI®). "Just Accepted" is an optional service offered to authors. Therefore, the "Just Accepted" Web site may not include all articles that will be published in the journal. After a manuscript is technically edited and formatted, it will be removed from the "Just Accepted" Web site and published as an ASAP article. Note that technical editing may introduce minor changes to the manuscript text and/or graphics which could affect content, and all legal disclaimers and ethical guidelines that apply to the journal pertain. ACS cannot be held responsible for errors or consequences arising from the use of information contained in these "Just Accepted" manuscripts.

ACS Publications

is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

Published by American Chemical Society. Copyright © American Chemical Society. However, no copyright claim is made to original U.S. Government works, or works produced by employees of any Commonwealth realm Crown government in the course of their duties.

DeepIce: a Deep Neural Network Approach to Identify Ice and Water Molecules

Maxwell Fulford,[†] Matteo Salvalaglio,[‡] and Carla Molteni[*,†]

†*King's College London, Department of Physics, Strand, London WC2R 2LS, United Kingdom*

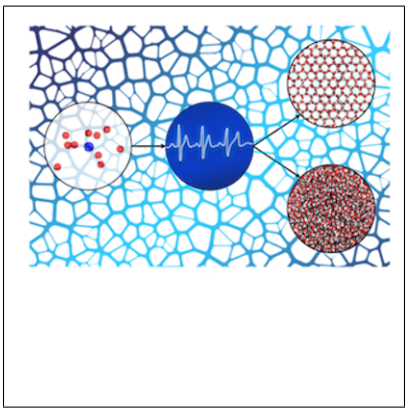
‡*University College London, Department of Chemical Engineering, Torrington Place, London WC1E 7JE, United Kingdom*

E-mail: carla.molteni@kcl.ac.uk

Abstract

Computer simulation studies of multi-phase systems rely on the accurate identification of local molecular structures and arrangements in order to extract useful insights. Local order parameters, such as Steinhardt parameters, are widely used for this identification task; however, the parameters are often tailored to specific local structural geometries and generalize poorly to new structures and distorted or under-coordinated bonding environments. Motivated by the desire to simplify the process and improve the accuracy, we introduce DeepIce, a novel deep neural network designed to identify ice and water molecules, which can be generalized to new structures where multiple bonding environments are present. DeepIce demonstrates that the characteristics of a crystalline or liquid molecule can be classified using as input simply the Cartesian coordinates of the nearest neighbors without compromising the accuracy. The network is flexible and capable of inferring rotational invariance, and produces a high predictive accuracy compared to the Steinhardt approach, the tetrahedral order parameter and polyhedral template matching in the detection of the phase of molecules in premelted ice surfaces.

Graphical TOC Entry



Introduction

Identifying the local arrangements of molecules within condensed phases is a key task in computer simulations aimed at studying phenomena such as phase transitions,[1–5] nucleation,[6–8] crystal growth[9,10] and defect formation.[11,12] Ice surfaces, which strongly influence the macroscopic properties of ice, are an interesting example of multi-phase systems.[13] Over a significant temperature range, ice surfaces premelt and develop a quasi-liquid layer (QLL) which mediates crystal growth and chemical reactions.[14–20] Computational studies of molecules at interfaces, including at the environmentally significant ice-QLL interface, rely on a precise and accurate method to distinguish between the local environment of molecules. Success of such studies is limited by an imprecise identification of molecular phases. As materials are increasingly studied at the molecular level using computer simulations, reducing errors in molecular phase identification is becoming more and more important.

Inspired by the importance and difficulty of the task, we set out to improve on the current state-of-the-art for identifying ice-like and liquid-like water molecules by introducing DeepIce, a collection of deep neural networks, that achieves a remarkable accuracy in comparison with a selection of commonly used methods. DeepIce is designed to identify bulk and surface ice-like and liquid-like water molecules in a slab and is conceived with a flexible framework that can in principle be generalized to any crystalline phase. Our approach builds upon the neural network proposed by Geiger and Dellago[21] which uses as input a collection of symmetry functions sensitive to the positions of atoms. Our proposed neural network simply requires as input the atomic coordinates of the molecules in the system and can adapt to any symmetry without modification.

Discriminating Local Environments: a Review of Existing Methods

A variety of local order parameters have been developed to discriminate between ordered crystal phases and disordered local arrangements.[22–30] Local order parameters can classify ice and water phases based on the coordinates of the oxygen atoms. In the following we briefly review a selection of common local order parameters which we will use as comparison with DeepIce. Besides a description of the tetrahedral order parameter, we outline in detail the popular Steinhardt approach[23–27] in the context of hexagonal ice and water molecule detection. In addition, we introduce a modified Steinhardt parameter, which as demonstrated in later sections, reduces the error of hexagonal ice and water molecule recognition compared to previous modifications of the 3rd order Steinhardt parameter. Moreover, we describe the Geiger-Dellago neural network[21] and the polyhedral template matching method[31] that can be applied to determine the molecular phase by matching the local topology with pre-computed template structures.

Tetrahedral Order Parameter

The tetrahedral order parameter identifies tetrahedral bonding environments, such as hexagonal ice, by calculating bond angles between first-shell neighbors. The parameter was first introduced by Chau and Hardwick[22] and refined by Errington and Debenedetti.[32] It has successfully been applied in a range of ice and water studies.[15,33–36] The parameter is defined for each molecule i as

$$q_{tet}(i) = \left[1 - \frac{3}{8}\sum_{j=1}^{3}\sum_{k=j+1}^{4}\left(\cos(\theta_{j,i,k}) + \frac{1}{3}\right)^2\right] \tag{1}$$

where $\theta_{j,i,k}$ is the bond angle formed by the oxygens of molecules j, i and k. The parameter is computed from a sum over the (four) nearest neighbours of molecule i. Its value is equal

to 1 when the four neighboring molecules form a perfect tetrahedral bonding environment; a threshold is usually set for finite temperature simulations. It is a simple parameter and straightforward to calculate; however, by design it does not capture any structure characterized by a local symmetry different to the tetrahedral one.

Steinhardt Parameters

Steinhardt parameters classify the phase of an atom or molecule based on the coherence of its orientational order with that of its neighbors.[23] The parameters are sensitive to the degree of correlation between the spatial orientation of the vectors which join neighbouring molecules. In hexagonal ice, the tetrahedra centered on neighboring molecules are aligned with a degree of order, whilst in water their alignment is random. The 3rd order Steinhardt parameter, q_3, exploits this difference and returns large values for ice-like molecules and small values for liquid-like molecules.

For the oxygen of molecule i, the rotationally invariant order l Steinhardt parameter $q_l(i)$ is defined as

$$q_l(i) = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |q_{lm}(i)|^2}, \tag{2}$$

where the $(2l+1)$ complex components, $q_{lm}(i)$, are

$$q_{lm}(i) = \frac{1}{N_b(i)} \sum_{j=1}^{N_b(i)} Y_{lm}(\mathbf{r}_{ij}). \tag{3}$$

Here $N_b(i)$ is the number of nearest neighbors within a chosen cutoff distance and $Y_{lm}(\mathbf{r}_{ij})$ are the spherical harmonics calculated for the vector $\mathbf{r}_{ij}$ connecting molecule i to the neighboring molecule j.

An improved distinction between ice-like and liquid-like water molecule is obtained with the averaged form of q_l[26]

$$\bar{q}_l(i) = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\bar{q}_{lm}(i)|^2}, \tag{4}$$

where the q_{lm} vector of molecule i is averaged with that of its nearest neighbors and itself

$$\bar{q}_{lm}(i) = \frac{1}{N_b(i)+1} \sum_{j=0}^{N_b(i)} q_{lm}(j). \tag{5}$$

We introduce here a further modification

$$\tilde{q}_l(i) = \frac{1}{N_b(i)+1} \sum_{j=0}^{N_b(i)} q_l(j), \tag{6}$$

where q_l is averaged with its nearest neighbors and itself. By averaging with the nearest neighbors, $\tilde{q}_l$ acknowledges that the characteristics of the environment of a molecule are highly correlated with the phase of the neighboring ones.

One of the disadvantages of the Steinhardt approach is that the choice of the order of the spherical harmonics relies on intuition and advanced knowledge of the underlying structures. Steinhardt bond order parameters are poor at distinguishing phases which contain multiple distinct local spatial environments within each phase.[7,37] The parameter was developed for bulk conditions and assumes a fully coordinated molecule. It is undefined if there are no neighboring molecules within the cutoff distance. If there is only one nearest neighbor, the parameter returns a value of 1, which implies perfect order even though an under-coordinated molecule is unlikely to belong to an ordered phase In practice choosing the ideal spherical harmonics for identifying many different phases involves a lengthy trial and error process. For example, the $l = 3$ Steinhardt parameter is an appropriate choice for distinguishing liquid and hexagonal ice molecules whereas the $l = 4$ Steinhardt parameter is a poor choice, as we demonstrate in the results section of this work. Previous work has shown that the $l = 3$ parameter best describes tetrahedrality.[7]

Polyhedral Template Matching

The polyhedral template matching (PTM) method[31] detects the topology of the local atomic environment by finding the best structural match, based on the root mean square deviation (RMSD), between an unclassified topology and candidate template structures. It has been demonstrated to be particularly robust at high temperature where bond length fluctuations may affect methods that rely on definition of nearest neighbors through distance cutoff.

For comparison with DeepIce, in this work we investigate classifying water and ice molecule using the PTM approach implemented in the visualization and analysis software OVITO.[38] PTM in OVITO has eight candidate templates: face-centered cubic (FCC), hexagonal close-packed (HCP), body-centered cubic (BCC), icosahedral coordination (ICO), simple cubic (SC), cubic diamond, graphene and hexagonal diamond. An RMSD threshold value is set so that particles that exceed the threshold value are assigned as "other".

Geiger-Dellago Neural Network

It has previously been shown that a neural network trained using a set of symmetry functions can be used to distinguish between liquid water and several phases of ice over a range of temperatures and pressures.[21] The Geiger-Dellago network[21] is based on a neural network method originally developed for energy and force calculations.[39,40] The input of the neural network is, for a given atom, a set of symmetry functions which are constructed using a combination of atomic distances and angles. The symmetry functions serve as the input for a feed forward neural network with two hidden layers. The output is a vector which indicates the predicted structure.

The choice of symmetry functions is key to the accuracy of the neural network and must be sensitive to the local environments of the atoms. The process of selecting the symmetry functions involves first outlining a set of functions believed to be appropriate for the given structures. This requires an initial study of the distribution of the distances and angles within the local environments to gain insight into the structural features that can then be

used to distinguish between the different phases. Based on these considerations, candidate symmetry functions are defined manually and their distributions within the different phases computed. The symmetry functions with the smallest overlap are then shortlisted. The next step involves a sensitivity analysis whereby the neural network is trained with the selected symmetry functions and any function which contributes weakly to the predictions is removed. The final outcome is 30-40 symmetry functions which are then used to retrain the neural network. If a new structure is introduced, the entire process must be repeated.

The Geiger-Dellago neural network has been tested on a Lennard-Jones system as well as ice and water, and has been applied to study the freezing of supercooled water to hexagonal ice.

DeepIce

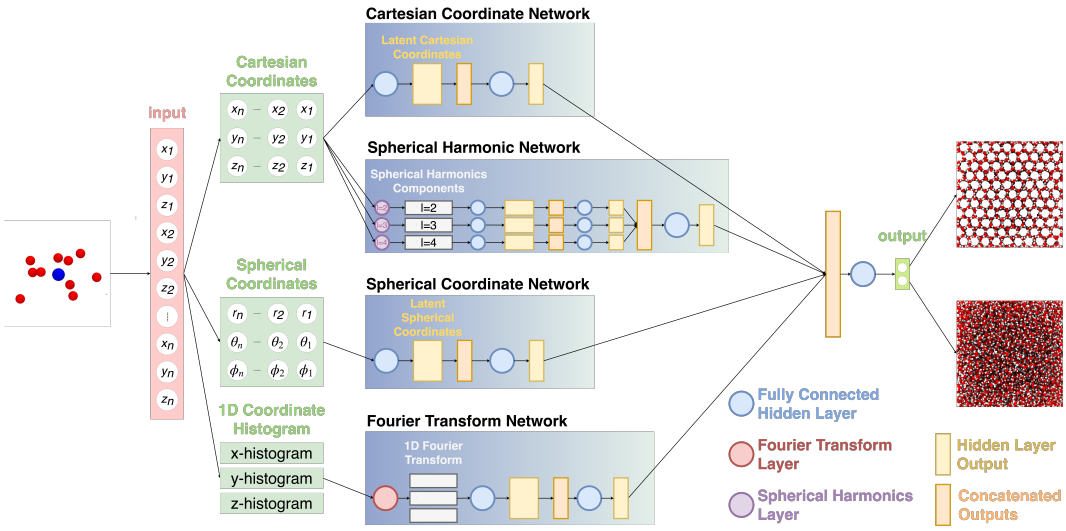


Figure 1: Representation of DeepIce showing the four subnetworks which are trained in unison and combined to produce an accurate phase predictor.

Inspired by the success of the Geiger-Dellago neural network[21] and building upon the strengths of the Steinhardt approach,[23–26] we introduce DeepIce, a novel deep neural network designed to identify the phase of water molecules, and demonstrate how it can be used to study the QLL of hexagonal ice surfaces and detect surface melting.

DeepIce is constructed using the deep learning python package Keras[41] and is powered by TensorFlow.[42] It is composed of four deep neural networks: a Cartesian coordinates network, a spherical coordinates network, a Fourier transform network and a spherical harmonics network. The input of DeepIce consists of the Cartesian coordinates of the n nearest neighbours of the water molecules' oxygen atoms in the form of a 1D vector with $3n$ elements. The Cartesian coordinates are centered on the oxygen atom of interest and defined in relation to it. Whereas the Steinhardt approach utilizes every neighbor within a cutoff distance, here we consider a fixed number of closest neighbors.

The four subnetworks each input the 1D Cartesian coordinates vector and transform it into high dimensional latent feature spaces. The latent representation of the coordinates input produced by the four subnetworks are combined through a final set of hidden layers before the phase prediction is made. The subnetworks are trained in unison and combine to produce an accurate and powerful molecular phase predictor. The overall architecture of DeepIce is outlined in Figure 1. In the following we describe the four subnetworks of DeepIce as well as the output layer.

Cartesian Coordinates Network

The architecture of the Cartesian coordinates network is shown in Figure 2. The network transmits the local Cartesian coordinates the n nearest neighbors of each oxygen atom, one by one, to the first set of fully connected hidden layers. The hidden layers transform each set of Cartesian coordinates into a high dimensional latent feature space. The hidden layers are composed of neurons with the rectified linear unit (RELU) activation.[43] We use two hidden layers with 250 and 50 neurons, as shown in Table 1 which outlines the number of neurons in the different networks. The hidden layer weights are shared across each neighbor to minimize the number of parameters and prevent overfitting. The output of the first set of hidden layers are n latent vectors of length 50, corresponding to the size of the final hidden layer. The number of neurons in the hidden layers is a parameter of the model which can be tuned.

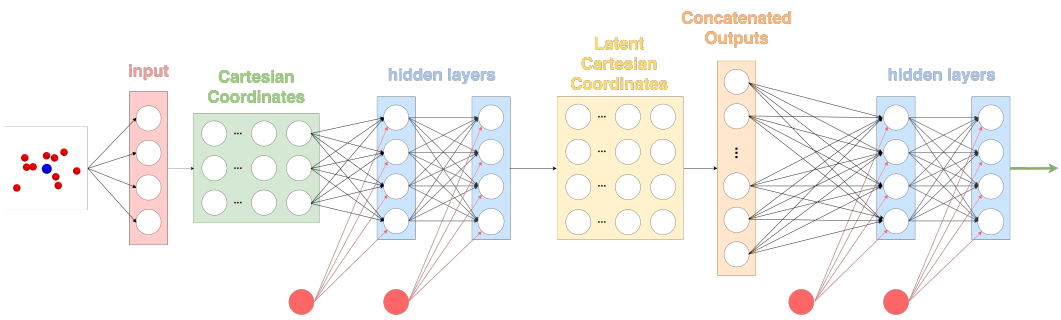


Figure 2: Cartesian coordinates network architecture. On the left, the Cartesian coordinates of the n nearest neighbors of the blue atom of interest are used to generate an input vector of length $3n$. The first set of hidden layers inputs the x, y, z Cartesian coordinates of the nearest neighbors one by one, and generates n latent vectors of length h_2. These latent vectors are combined to produce a concatenated vector of length nh_2 and transformed through a second set of hidden layers to produce a hidden layer output of length h_4. A bias term, represented by the red dots and lines, is learnt during the training procedure and inputted into each hidden layer.

The latent vectors provide a representation of the neighbors of the molecule in question. The n latent vectors produced by the first set of hidden layers are concatenated to produce a single vector. For example, with 10 nearest neighbors and 50 neurons in the final hidden layer, a concatenated vector of length 500 is produced. The concatenated latent vector is fed into the second set of hidden layers, which also uses RELU activation. The second set of hidden layers transforms the individual Cartesian coordinate latent vectors into a shared high dimensional latent feature space. The shared latent vector provides a description of the local molecular environment of the molecule in question and is subsequently combined with the outputs from the three other subnetworks to make a final phase prediction. The weights and biases of each neuron in the hidden layers are tuned during the training procedure and initialized randomly.

Table 1: Number of Neurons in DeepIce Networks

Network	Set 1 Hidden Layers	Set 2 Hidden Layers	Set 3 Hidden Layers
Cartesian Coordinates	250, 50	250, 50	
Spherical Coordinates	250, 50	250, 50	
Fourier Transform	250, 50	250, 50	
Spherical Harmonics	250, 50	250, 50	200, 50
Output	250, 50, 5		

Spherical Coordinates Network

The spherical coordinates network has an identical architecture to the Cartesian coordinates network; however, its input consists of the spherical coordinates as opposed to the Cartesian coordinates. As such, the spherical coordinates network includes an initial layer which transforms the Cartesian coordinates into the spherical domain. Once transformed, the spherical coordinates are passed into the first set of hidden layers with shared weights and transformed into a high dimensional space. The high dimensional latent features are then combined into a single vector which is passed through a second set of hidden layers. The latent output of the second set of hidden layers is combined with the output of the Cartesian coordinates network and the two other subnetworks. Converting the coordinates into the spherical domain enables the network to focus on learning relationships between the angles formed by the nearest neighbors and the molecular phase. The data embedded within the spherical network complements the Cartesian coordinates network and helps DeepIce produce more robust predictions.

Fourier Transform Network

Inspired by crystallography, the Fourier transform network learns to identify a Fourier fingerprint of each phase. The network computes three one-dimensional coordinates histograms using the x, y and z nearest neighbour coordinates. The histograms are treated as periodic functions and their one-dimensional Fourier transforms are computed resulting in three one-dimensional Fourier transform outputs.

The resulting Fourier series are subsequently passed through a deep neural network. The three Fourier outputs are transformed through a set of hidden layers using weights that are shared across each of the Fourier outputs to prevent overfitting. As with the Cartesian and spherical coordinate networks, the hidden layers are composed of RELU units and the respective number of neurons specified in Table 1. The output of the hidden layers are three Fourier transform embedding vectors, which are concatenated together to produce one latent

vector. The latent vector is passed through a second set of hidden layers, composed of RELU units, to produce a final latent output which is combined with the outputs of the other three subnetworks to produce a final phase prediction.

Spherical Harmonics Network

The spherical harmonics network is inspired by the Steinhardt approach.[23–26] The network is composed of an initial embedding layer which computes the l=2, l=3 and l=4 order spherical harmonics using the nearest neighbor coordinates. The three orders of spherical harmonics are passed separately into three subnetworks, each with an architecture identical to the Cartesian coordinate network. Each spherical harmonics subnetwork initially transforms the spherical harmonic components of each nearest neighbor through a set of hidden layers with shared weights and RELU units. The latent outputs of the nearest neighbours are concatenated together and passed through a second set of hidden layers with RELU units. The latent output of the l=2, l=3 and l=4 subnetworks are concatenated together and passed into a third set of hidden layers and transformed into a shared latent feature space to produce a single spherical harmonics latent vector.

Whilst the Steinhardt approach sums the spherical harmonics to produce a parameter, the spherical harmonics network infers how to combine the $2l+1$ components to maximize the predictive accuracy. This allows the network to tailor the transformation of the components to the molecular structures being detected, allowing the network. In addition, the network learns the relationship between the l=2, l=3 and l=4 outputs and how to best combine them to minimize the prediction uncertainty.

For the examples studied here, considering l=2,3,4 spherical harmonics produce accurate results. Including more orders of spherical harmonics would make DeepIce computationally more expensive; however, in principle DeepIce can include additional spherical harmonics, if a user wishes or a system requires.

Output Network

The outputs of the Cartesian coordinates, spherical coordinates, Fourier transform and spherical harmonics networks are concatenated together to produce one large vector that encodes the latent descriptions of the local molecular environment generated by the four subnetworks. The final vector is passed through a set of fully connected hidden layers. The final set is composed of three hidden layers with the architecture outlined in Table 1. The output of the hidden layers is passed to a softmax function to return the predicted phase of the input molecule. DeepIce can be thought of as an ensemble method in which pseudo predictions are made by each subnetwork. The output network considers the pseudo predictions to generate one final phase prediction.

Training of DeepIce

The number of neurons used in each hidden layer within the four networks is outlined in Table 1. Set 1 hidden layers corresponds to the initial hidden layers within the subnetworks that transform the input into a high dimensional feature space. Set 2 hidden layers are the second set of hidden layers within the subnetworks which transform the latent vectors into a shared feature space. The spherical harmonics network has a third set of hidden layers which transforms the three l orders of spherical harmonics embeddings into a shared latent feature space.

DeepIce is trained here to distinguish between hexagonal ice-like and liquid-like water molecules. The process involves adjusting the weights and biases of the neural network to reduce the error of the predictions from the training data. The training data is composed of the input coordinates of the n nearest neighbors, determined according to their distance, and the target output in the form of a 2×1 vector. The target vector has exactly one element equal to 0 and one element equal to one. The order encodes whether a molecule is ice-like or liquid-like. The number of nearest neighbors is a parameter of the model. In this work

we select 10 nearest neighbors in the first instance, and investigate the impact of n on the accuracy later on.

The training process involves tuning the weights and biases. The training set $\mathcal{T}$ consists of the nearest neighbors coordinate vector $\mathbf{X}$ and the corresponding structure vector $\tilde{\mathbf{y}}$,

$$\mathcal{T} = \{\mathbf{X}, \tilde{\mathbf{y}}\} \tag{7}$$

$\tilde{\mathbf{y}}$ is the output vector from the neural network and encodes the classes of the structures. In the case of classifying ice-like and liquid-like, $\tilde{\mathbf{y}}^{(i)}$ is comprised of two elements for each entry i. If the structure of entry i is hexagonal ice, $\tilde{\mathbf{y}}^{(i)} = \{1, 0\}$, and if it is liquid-like $\tilde{\mathbf{y}}^{(i)} = \{0, 1\}$.

The training data set is produced by running molecular dynamics (MD) simulations of the phases in question with the TIP4P/Ice force-field,[44] which reproduces well the ice crystalline phases and the melting point. Bulk hexagonal ice and supercooled water at 260 K are modeled with 2,880 molecules and periodic boundary conditions ensure that there are no interfaces or surfaces. MD simulations are performed using the LAMMPS package (9 Dec 2014 version)[45] in the NVT ensemble with a time step of 1 fs and a Nosé-Hoover thermostat with a relaxation time of 1 fs. The Lennard-Jones potential and the real part of the Coulombic potential are truncated at 12.0 Å. A particle-particle particle-mesh solver is used to compute long-range Coulombic and Lennard-Jones interactions in reciprocal space. The systems are equilibrated for 0.24 ns before production runs for 1 ns from which 1001 trajectory frames are saved every 1 ps.

10 nearest neighbors are collected for each molecule producing a training set, validation set and test set with 4,670,266, 518,918 and 576,576 configurations, respectively. When calculating the relative coordinates of the nearest neighbors, the y-dimension of the simulation boxes are increased by 30 Å to create the effect of an ice/vacuum and water/vacuum interface. The slab generated in this way has two surfaces which in the hexagonal ice case considered here correspond to the $(10\bar{1}0)$ primary prism face. The inclusion of a surface enables the neural network to learn to identify under-coordinated surface ice and water molecules as well

as bulk ice and water molecules. It is assumed that there are no phase transitions during the MD simulations, which is verified by visual inspection and by monitoring the potential energy during the simulations.

The classification error of the network is monitored during the training procedure and used to drive the tuning of the weights and biases. The cross-entropy loss function is used to train DeepIce and is defined as

$$E = -\frac{1}{n_t}\sum_{i}^{n_t}\left[\tilde{y}_t^{(i)}\log\left(y_p(X^{(i)})\right) + \left(1-\tilde{y}_t^{(i)}\right)\log\left(1-y_p(X^{(i)})\right)\right] \tag{8}$$

where n_t is the total number of items in the training data, $\tilde{y}_t^{(i)}$ is the desired output for entry i and $y_p(X^{(i)})$ is the predicted output. E is minimized using the back-propagation algorithm.[46]

The network is trained using a first-order gradient-based optimizer known as *Adam* with an initial learning rate of 0.001, which is the same used by the original authors.[47] Mini-batch learning using a batch size of 30 is performed, such that the weights and biases are tuned each time the learning procedure cycles through every entry in the batch. Each cycle through the entire data set is known as an *epoch* and training is allowed for a maximum of 30 epochs to avoid overfitting. The weights and biases are initially randomized and, at the beginning of each epoch, the training data is shuffled to reduce variance and prevent overfitting. During training the proportion of incorrect classifications within the validation set is monitored to ensure that the network does not overfit. Overfitting occurs if the validation error increases whilst the training error continues to decrease. It is avoided using early stopping whereby the training procedure is halted if the validation error increases during six consecutive epochs. Six epochs is a regularization parameter that can be tuned based on the evolution of the training and validation loss. The training process described here takes a day using 1 CPU on a conventional laptop; prediction on a slab of 5760 molecules takes around 0.75s for each frame on the same computational setup.

Accuracy of DeepIce

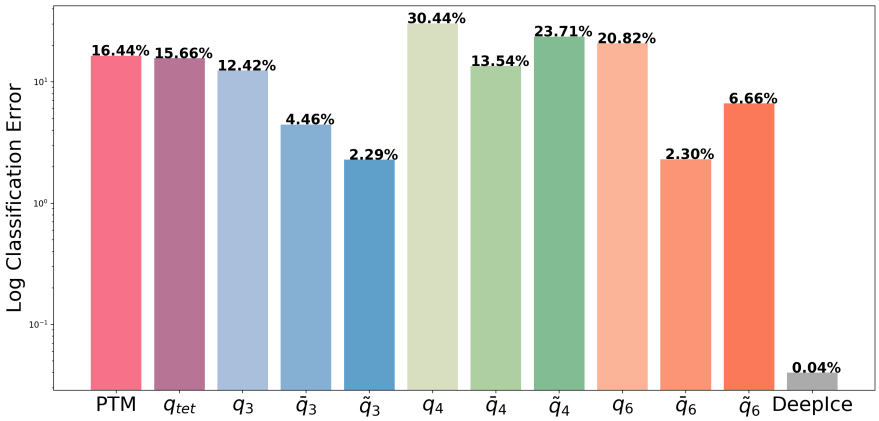


Figure 3: Classification error of the polyhedral template matching (PTM) method, the tetrahedral order parameter (q_{tet}), the Steinhardt approaches and DeepIce trained on the normal training set.

The nearest neighbour training input dataset is randomly split into a training set (81%), validation set (9%) and test set (10%). The classification accuracy of the 3rd, 4th and 6th Steinhardt parameters on the test set, along with the accuracy of the PTM method, the tetrahedral order parameter and DeepIce trained on the training set, are shown in Figure 3.

The Steinhardt parameters are calculated using a cutoff distance of 3.5 Å to calculate the first-shell neighbors, as is common with the TIP4P/Ice[44] force-field.[7,48] Threshold values of 0.680, 0.679, 0.267, 0.496, 0.503, 0.309, 0.534, 0.533 and 0.371 are used for q_3, $\tilde{q}_3$, $\bar{q}_3$, q_4, $\tilde{q}_4$, $\bar{q}_4$, q_6, $\tilde{q}_6$ and $\bar{q}_6$, respectively. Molecules that have a Steinhardt parameter above the threshold are classified as ice and below as liquid. The thresholds used correspond to the values that minimize the rate of false ice and water classification within the dataset used in this work and are determined by a trial and error procedure. Overall the original Steinhardt method performs poorly with errors of 12.42%, 30.44% and 20.82% for q_3, q_4 and q_6, respectively. The 4th order parameters perform particularly poorly in distinguishing hexagonal ice from water. The 6th order Steinhardt parameter performs well using the modification by Lechner and Dellago[26] with an error of 2.30%. However, our modification of the 3rd order parameter,

$\tilde{q}_3$ (see Eq. 6) is the best performing Steinhardt parameter with an error of 2.29%. DeepIce, trained on the training set, produces an astonishing accuracy of 0.04% on the test set, which we know to contain either water or ice molecules. It is clear that DeepIce is a substantial improvement compared to the Steinhardt method which is defined for the bulk and does not handle well under-coordinated molecules.

The tetrahedral order parameter, q_{tet} performs poorly with an error 15.66% using a threshold of 0.905 determined by trial and error to minimise misclassification. q_{tet} is calculated using *iOrder*,[49] an open source python library.

PTM[31,38] is also relatively poor at distinguishing ice and water molecules using templates for BCC, FCC, HCP, ICO, SC, cubic diamond, graphene and hexagonal diamond. With PTM an optimal error rate of 16.44% is achieved by assigning molecules that match the hexagonal diamond topology as ice and the remaining molecules as water. An RMSD threshold of 1.0 Å is used for the calculations. The results are unchanged with a higher RMSD threshold values and decrease in accuracy with smaller values.

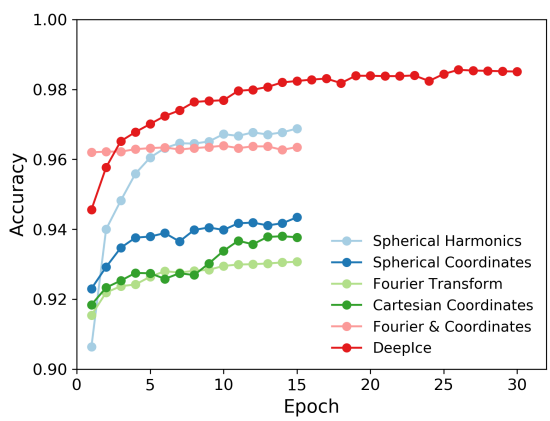


Figure 4: Accuracy of the rotated validation set obtained with DeepIce and subnetworks, trained on the rotated training set, as a function of the number of epochs.

In order to assess the rotational invariance of the network, a new data set is produced from the original nearest neighbour data set by randomly rotating the axes of the nearest neighbor Cartesian coordinates. This is achieved by rotating each x, y, z Cartesian coordinates along a random axis and by a random magnitude. The random rotation of each input is unique such

that no two Cartesian coordinate inputs are rotated in the same way. Training DeepIce on the randomly rotated dataset enables the network to infer rotational invariance. The rotational invariance is embedded into the weights and parameters that are learnt. Previous works on recognizing hand written digits have revealed the benefits of random transformations applied to the training dataset and demonstrated the ability of neural networks to infer rotational invariance.[50] The randomly rotated dataset is referred to as the "*rotated*" dataset in the following, whilst the unrotated dataset is referred to as the "*normal*" dataset. The prediction accuracy of DeepIce trained using the normal dataset is poor on the rotated test set and close to random. This validates the need to train DeepIce to infer rotational invariance. The accuracy of DeepIce trained on the rotated dataset produces an error of 1.02% on the normal test set which remains a significant improvement compared to the Steinhardt approach.

In order to demonstrate the influence of the choice of training and test set, we retrain DeepIce on the first 500 ps of the MD simulation and evaluate on the final 100 ps with a 400 ps buffer in between. We also report the accuracy of DeepIce trained on the first 800 ps and evaluated on the final 100 ps with a 100 ps buffer. The training set and test set are randomly rotated. Trained on the first 800 ps for 15 epochs produces an error of 1.05% in good agreement with the 1.02% achieved on the rotated dataset. Training on the first 500 ps for 15 epochs will inevitably result in a higher error due to the substantially smaller training set; however, an error of only 1.12% is achieved. The results evaluated on a test set with a 100 ps and 400 ps buffer period are in good agreement with the rotated accuracy, instilling confidence that the rotated training and test sets used in this work are not biasing the results.

DeepIce is composed of four subnetworks. Each subnetwork learns to recognize different features and distributions within the dataset. The combination of the four networks results in a more powerful and accurate predictor. To demonstrate this, the four subnetworks are separated and trained to predict the phase independently. In addition, a simplified version of DeepIce composed of the spherical coordinates network, Cartesian coordinate network

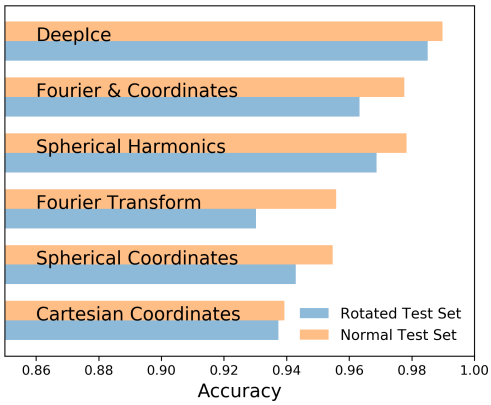


Figure 5: Accuracy of DeepIce and subnetworks trained on the rotated training set. Accuracies on the rotated and normal test sets are reported.

and Fourier transform network but without the more computationally expensive spherical harmonics network is trained. The networks are trained using the rotated training dataset for 15 epochs. The rotated validation accuracy is shown in Figures 4. The results show that the Fourier, spherical coordinates and Cartesian coordinate networks reach a comparable accuracy. However, combined together the three subnetworks perform much better and the accuracy increases and becomes comparable to the spherical harmonics subnetwork. Combining all four subnetworks together results in a further improvement to the DeepIce accuracy. The normal and rotated test prediction accuracy is summarized in Figure 5 and highlights this trend.

Compared to the Geiger-Dellago network,[21] DeepIce does not require the lengthy process of shortlisting and selecting symmetry functions and sensitivity analysis. The input for a given atom is simply the x, y and z nearest neighbor coordinates relative to the atom. The simplicity does not compromise the accuracy and the only parameter of the input is the number of nearest neighbors.

Hexagonal ice have four nearest neighbors in the first coordination shell and 12 in the second. In this work we include 10 nearest neighbors within our input, in the first instance, to take fully into account the first coordination shell and part of the second. In order to assess the impact of the number of nearest neighbors on the accuracy, DeepIce is retrained

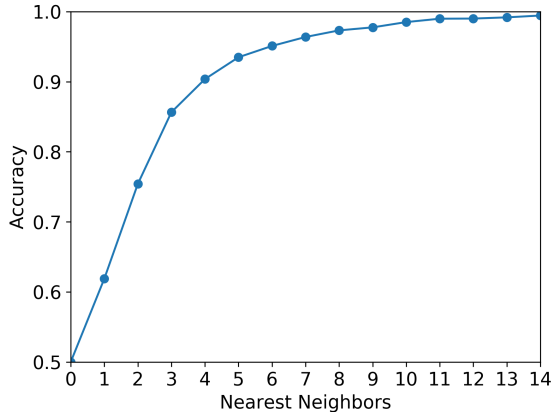


Figure 6: DeepIce accuracy as a function of the number of nearest neighbors. DeepIce is trained using the rotated training set and evaluated on the rotated test set.

using 1 to 14 nearest neighbors for 10 epochs on the rotated training set. The rotated test set accuracy is shown in Figure 6. Using 0 nearest neighbors the accuracy is 50% and equivalent to a random guess, as the number of nearest neighbors increases to three there is a sharp increase in accuracy. At this point DeepIce has enough information on the local molecular environments to correctly classify around 85% of water and ice molecules. The results indicate that the larger the number of nearest neighbors included in the input, the higher the accuracy of the network. The Geiger-Dellago neural network[21] showed an accuracy of 85% for ice V and 98% for ice III when including two neighbour shells and an increase to nearly 100% when including three neighbor shells.

Application to Ice Surfaces

In order to demonstrate DeepIce and the Steinhardt approach in a multi-phase system, we apply the algorithms to a snapshot of a MD simulation at 270 K for a slab of hexagonal ice containing 5760 molecules and two $(\bar{1}2\bar{1}0)$ secondary prism surfaces. The snapshot is taken from a 400 ns MD simulations with the TIP4P/Ice force field[44] using a similar protocol to the one used to produce the training set as previously described. At 270 K, close to the melting point, the surface is composed of a substantial QLL, which is clearly visible in the structure

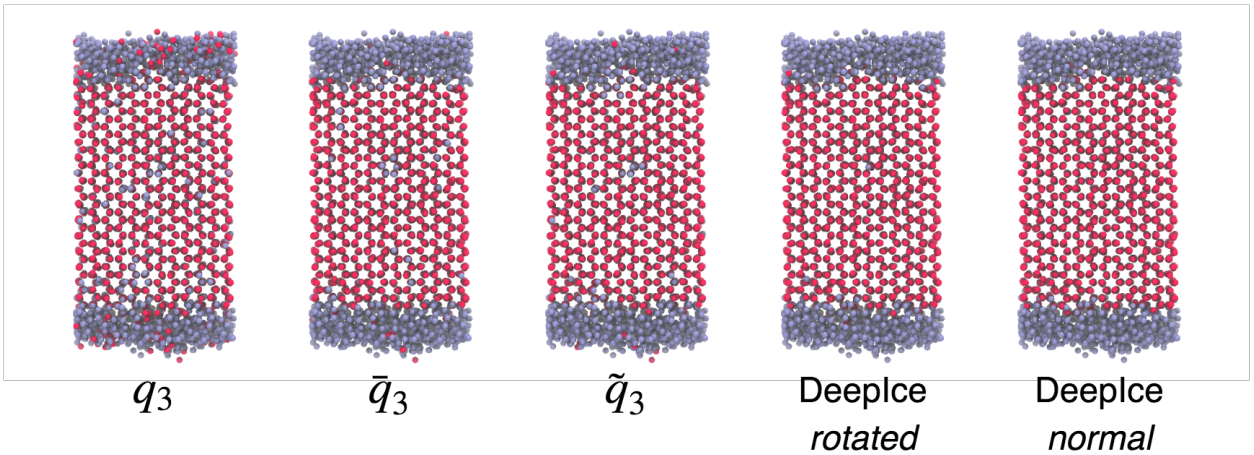


Figure 7: Classification of ice-like (red) and water-like (light blue) molecules by the Steinhardt approaches and DeepIce for a MD snapshot of a hexagonal ice slab with secondary prism surfaces at 270 K.

in Figure 7. Molecules predicted to be ice-like are colored red and molecules predicted to be liquid-like by the various methods are colored light blue.

The results clearly show the superior predictive accuracy of DeepIce compared with the 3rd order Steinhardt parameters. q_3 performs poorly and misclassifies a large number of bulk ice molecules as water. $\bar{q}_3$ and $\tilde{q}_3$ are a significant improvement but nonetheless misclassify a number of obvious bulk ice molecules as water and QLL molecules as ice. DeepIce trained using the rotated data set misclassifies less than five bulk ice molecule as water and makes very few clear mistakes in the QLL. DeepIce trained using the normal data set performs very well and makes almost no evident mistake.

The normal data set is collected using simulations in the same orientation as the snapshot in Figure 7. The improved performance of DeepIce trained on the normal set compared to the rotated dataset indicates that DeepIce is able to more easily predict the phase when we constrain the possible rotations. For studies of slabs or structures that cannot rotate it is therefore desirable to train DeepIce using an unbiased or unrotated simulation data set. However, for simulations where the orientation can vary, such as simulations of nanoparticles or simulation of nucleation processes, a rotated data set is expected to train a superior

predictor. The surface in the snapshot in Figure 7 is the secondary prism plane of hexagonal ice (perpendicular to the x-axis in our simulation set up), whereas DeepIce is trained with an added surface perpendicular to the y-axis and corresponding to the primary prism plane. The ability of the network to accurately classify QLL molecules along the surface normal to the x-axis demonstrates that the neural network is robust and has inferred a general understanding of surfaces.

Conclusions

In conclusion, we have developed a deep neural network which can very accurately identify ice-like and liquid-like water molecules simply using as input the Cartesian coordinates of a selected number of nearest neighbors. Our approach does not rely on any knowledge of the underlying molecular structures and is more precise than approaches based on the Steinhardt parameters.

We offer a much simpler solution compared to the Geiger-Dellago network[21] without compromising the accuracy. Whilst the Geiger-Dellago network relies on, and is therefore limited by, the choice of symmetry functions, ours takes in an essentially raw input data form with only one parameter, the number of nearest neighbors. Using the relative coordinates of the nearest neighbors allows our network to be applied to systems of any size without having to be retrained as shown in the case of the slab with a different surface orientation. A large number of weights and biases are tuned during the training procedure as the network learns features that can be used to recognize different structures. The complicated task of feature engineering and extracting functions of the atomic coordinates which can be used to identify the local environment of atoms is taken care of by the neural network. The multiple subnetworks enable DeepIce to learn features at different levels of abstraction. We compare the performance of DeepIce with the widely used Steinhardt parameter, the tetrahedral order parameter and polyhedral template matching. Our results reveal a significant increase in

accuracy for hexagonal ice and water classification. DeepIce can in principle be extended to classify multiple ordered phases; however, further investigation is required to establish the success of the network in this task.

Acknowledgement

We are grateful for computational support from the UK high performance computing service ARCHER, for which access was obtained via the UKCP consortium and funded by grants EP/K013831/1 and EP/P022472/1 from the Engineering and Physical Sciences Research Council (EPSRC). We thank the Royal Society International Exchange Scheme 2013/R2. MF was supported by an EPSRC doctoral training studentship.

DeepIce code is publicly available at https://github.com/mfulford/DeepIce

References

(1) Shor, S.; Yahel, E.; Makov, G. Evolution of Short Range Order in Ar: Liquid to Glass and Solid Transitions – A Computational Study. *AIP Adv.* **2018**, *8*, 045215.

(2) Xiong, L.; Wang, X.; Yu, Q.; Zhang, H.; Zhang, F.; Sun, Y.; Cao, Q.; Xie, H.; Xiao, T.; Zhang, D.; Wang, C.; Ho, K.; Ren, Y.; Jiang, J. Temperature-Dependent Structure Evolution in Liquid Gallium. *Acta Mater.* **2017**, *128*, 304–312.

(3) Kbirou, M.; Trady, S.; Hasnaoui, A.; Mazroui, M. Cooling Rate Dependence and Local Structure in Aluminum Monatomic Metallic Glass. *Philos. Mag.* **2017**, *97*, 2753–2771.

(4) Cicco, A. D.; Iesari, F.; De Panfilis, S.; Celino, M.; Giusepponi, S.; Filipponi, A. Local Fivefold Symmetry in Liquid and Undercooled Ni Probed by X-Ray Absorption Spectroscopy and Computer Simulations. *Phys. Rev. B* **2014**, *89*, 060102.

(5) Du, C. X.; van Anders, G.; Newman, R. S.; Glotzer, S. C. Shape-Driven Solid–Solid Transitions in Colloids. *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114*, E3892–E3899.

(6) Radhakrishnan, R.; Trout, B. L. Nucleation of Crystalline Phases of Water in Homogeneous and Inhomogeneous Environments. *Phys. Rev. Lett.* **2003**, *90*, 158301.

(7) Reinhardt, A.; Doye, J. P. K.; Noya, E. G.; Vega, C. Local Order Parameters for Use in Driving Homogeneous Ice Nucleation with All-Atom Models of Water. *J. Chem. Phys.* **2012**, *137*, 194504.

(8) Haji-Akbari, A.; Debenedetti, P. G. Computational Investigation of Surface Freezing in a Molecular Model of Water. *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114*, 3316–3321.

(9) Dhakal, S.; Kohlstedt, K. L.; Schatz, G. C.; Mirkin, C. A.; Olvera de la Cruz, M. Growth Dynamics for DNA-Guided Nanoparticle Crystallization. *ACS Nano* **2013**, *7*, 10948–10959.

(10) Anwar, M.; Turci, F.; Schilling, T. Crystallization Mechanism in Melts of Short n-Alkane Chains. *J. Chem. Phys.* **2013**, *139*, 214904.

(11) Archer, A.; Foxhall, H. R.; Allan, N. L.; Gunn, D. S. D.; Harding, J. H.; Todorov, I. T.; Travis, K. P.; Purton, J. A. Order Parameter and Connectivity Topology Analysis of Crystalline Ceramics for Nuclear Waste Immobilization. *J. Phys.: Condens. Matter* **2014**, *26*, 485011.

(12) Kawasaki, T.; Onuki, A. Construction of a Disorder Variable from Steinhardt Order Parameters in Binary Mixtures at High Densities in Three Dimensions. *J. Chem. Phys.* **2011**, *135*, 174109.

(13) Petrenko, V. F.; Whitworth, R. W. *Physics of Ice*; Oxford University Press, Oxford, 1999.

(14) Henson, B. F.; Voss, L. F.; Wilson, K. R.; Robinson, J. M. Thermodynamic Model of Quasiliquid Formation on H2O Ice: Comparison with Experiment. *J. Chem. Phys.* **2005**, *123*, 144707.

(15) Conde, M. M.; Vega, C.; Patrykiejew, A. The Thickness of a Liquid Layer on the Free Surface of Ice as Obtained from Computer Simulation. *J. Chem. Phys.* **2008**, *129*, 014702.

(16) Calvert, M. J., Jack G. Impact on Global Change; Blackwell Scientific; Oxford, Boston, 1994.

(17) Asakawa, H.; Sazaki, G.; Nagashima, K.; Nakatsubo, S.; Furukawa, Y. Prism and Other High-Index Faces of Ice Crystals Exhibit Two Types of Quasi-Liquid Layers. *Cryst. Growth Des.* **2015**, *15*, 3339–3344.

(18) Dosch, H.; Lied, A.; Bilgram, J. H. Glancing-Angle X-Ray Scattering Studies of the Premelting of Ice Surfaces. *Surf. Sci.* **1995**, *327*, 145–164.

(19) Sazaki, G.; Zepeda, S.; Nakatsubo, S.; Yokomine, M.; Furukawa, Y. Quasi-liquid Layers on Ice Crystal Surfaces are Made up of Two Different Phases. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, 1052–1055.

(20) Murata, K.-I.; Asakawa, H.; Nagashima, K.; Furukawa, Y.; Sazaki, G. Thermodynamic Origin of Surface Melting on Ice Crystals. *Proc. Natl. Acad. Sci. U.S.A.* **2016**, *113*, E6741–E6748.

(21) Geiger, P.; Dellago, C. Neural Networks for Local Structure Detection in Polymorphic Systems. *J. Chem. Phys.* **2013**, *139*, 164105.

(22) Chau, P.-L.; Hardwick, A. J. A New Order Parameter for Tetrahedral Configurations. *Mol. Phys.* **1998**, *93*, 511–518.

(23) Steinhardt, P.; Nelson, D.; Ronchetti, M. Bond-Orientational Order in Liquids and Glasses. *Phys. Rev. B* **1983**, *28*, 784–805.

(24) ten Wolde, P.-R.; Ruiz-Montero, M. J.; Frenkel, D. Simulation of Homogeneous Crystal Nucleation Close to Coexistence. *Faraday Discuss.* **1996**, *104*, 93–110.

(25) Auer, S.; Frenkel, D. Numerical Simulation of Crystal Nucleation in Colloids. *Adv. Polym. Sci.* **2005**, *173*, 149.

(26) Lechner, W.; Dellago, C. Accurate Determination of Crystal Structures Based on Averaged Local Bond Order Parameters. *J. Chem. Phys.* **2008**, *129*, 114707.

(27) Paolantoni, M.; Lago, N. F.; Albertí, M.; Laganà, A. Tetrahedral Ordering in Water: Raman Profiles and their Temperature Dependence. *J. Phys. Chem. A* **2009**, *113*, 15100–15105.

(28) Pietrucci, F. In *Handbook of Materials Modeling: Methods: Theory and Modeling*; Andreoni, W., Yip, S., Eds.; Springer International Publishing: Cham, 2018; pp 1–23.

(29) Martelli, F.; Ko, H.-Y.; Oğuz, E. C.; Car, R. Local-Order Metric for Condensed-Phase Environments. *Phys. Rev. B* **2018**, *97*, 064105.

(30) Nguyen, A. H.; Molinero, V. Identification of Clathrate Hydrates, Hexagonal Ice, Cubic Ice, and Liquid Water in Simulations: the CHILL+ Algorithm. *J. Phys. Chem. B* **2015**, *119*, 9369–9376, PMID: 25389702.

(31) Larsen, P. M.; Schmidt, S.; Schiøtz, J. Robust Structural Identification via Polyhedral Template Matching. *Modell. Simul. Mater. Sci. Eng.* **2016**, *24*, 055007.

(32) Errington, J.; G. Debenedetti, P. Relationship between Structural Order and the Anomalies of Liquid Water. *Nature* **2001**, *409*, 318–321.

(33) Giovambattista, N.; Debenedetti, P. G.; Sciortino, F.; Stanley, H. E. Structural Order in Glassy Water. *Phys. Rev. E* **2005**, *71*, 061505.

(34) Yan, Z.; Buldyrev, S. V.; Kumar, P.; Giovambattista, N.; Debenedetti, P. G.; Stanley, H. E. Structure of the First- and Second-Neighbor Shells of Simulated Water: Quantitative Relation to Translational and Orientational Order. *Phys. Rev. E* **2007**, *76*, 051201.

(35) Chatterjee, S.; Debenedetti, P. G.; Stillinger, F. H.; Lynden-Bell, R. M. A Computational Investigation of Thermodynamics, Structure, Dynamics and Solvation Behavior in Modified Water Models. *J. Chem. Phys.* **2008**, *128*, 124511.

(36) Pan, D.; Liu, L.-M.; Slater, B.; Michaelides, A.; Wang, E. Melting the Ice: On the Relation between Melting Temperature and Size for Nanoscale Ice Crystals. *ACS Nano* **2011**, *5*, 4562–4569.

(37) Brukhno, A. V.; Anwar, J.; Davidchack, R.; Handel, R. Challenges in Molecular Simulation of Homogeneous Ice Nucleation. *J. Phys.: Condens. Matter* **2008**, *20*, 494243.

(38) Stukowski, A. Visualization and Analysis of Atomistic Simulation Data with OVITO–the Open Visualization Tool. *Modell. Simul. Mater. Sci. Eng.* **2009**, *18*, 015012.

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

(40) Behler, J. Neural Network Potential-Energy Surfaces in Chemistry: a Tool for Large-Scale Simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930–17955.

(41) Chollet, F. Keras. `https://keras.io`, 2015.

(42) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.;

Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow. `http://tensorflow.org`, 2015.

(43) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning. USA, 2010; pp 807–814.

(44) Abascal, J. L. F.; Sanz, E.; García Fernández, R.; Vega, C. A potential model for the study of ices and amorphous water: TIP4P/Ice. *J. Chem. Phys.* **2005**, *122*, 234511.

(45) Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **1995**, *117*, 1–19.

(46) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533–536.

(47) Kingma, D. P.; Ba, J. Adam: a Method for Stochastic Optimization. *CoRR* **2014**, *abs/1412.6980*.

(48) Sanz, E.; Vega, C.; Espinosa, J. R.; Caballero-Bernal, R.; Abascal, J. L. F.; Valeriani, C. Homogeneous Ice Nucleation at Moderate Supercooling from Molecular Simulation. *J. Am. Chem. Soc.* **2013**, *135*, 15008–15017.

(49) Du, P. iOrder. `https://github.com/ipudu/order`, 2017.

(50) Simard, P. Y.; Steinkraus, D.; Platt, J. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. Seventh International Conference on Document Analysis and Recognition. Proceedings. 2003.