The Journal of Chemical Physics

A neural network potential with self-trained atomic fingerprints: A test with the mW water potential

Cite as: J. Chem. Phys. **158**, 104501 (2023); https://doi.org/10.1063/5.0139245
Submitted: 19 December 2022 • Accepted: 16 February 2023 • Accepted Manuscript Online: 16 February 2023 • Published Online: 08 March 2023

Francesco Guidarelli Mattioli, Francesco Sciortino and John Russo

AIP Publishing

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

A neural network potential with self-trained atomic fingerprints: A test with the mW water potential

Cite as: J. Chem. Phys. 158, 104501 (2023); doi: 10.1063/5.0139245
Submitted: 19 December 2022 • Accepted: 16 February 2023 •
Published Online: 8 March 2023

Francesco Guidarelli Mattioli, Francesco Sciortino, and John Russo[a]

AFFILIATIONS

Sapienza University of Rome, Piazzale Aldo Moro 2, 00185 Rome, Italy

Note: This paper is part of the JCP Special Topic on Machine Learning Hits Molecular Simulations.
[a]**Author to whom correspondence should be addressed:** john.russo@uniroma1.it

ABSTRACT

We present a neural network (NN) potential based on a new set of atomic fingerprints built upon two- and three-body contributions that probe distances and local orientational order, respectively. Compared with the existing NN potentials, the atomic fingerprints depend on a small set of tunable parameters that are trained together with the NN weights. In addition to simplifying the selection of the atomic fingerprints, this strategy can also considerably increase the overall accuracy of the network representation. To tackle the simultaneous training of the atomic fingerprint parameters and NN weights, we adopt an annealing protocol that progressively cycles the learning rate, significantly improving the accuracy of the NN potential. We test the performance of the network potential against the mW model of water, which is a classical three-body potential that well captures the anomalies of the liquid phase. Trained on just three state points, the NN potential is able to reproduce the mW model in a very wide range of densities and temperatures, from negative pressures to several GPa, capturing the transition from an open random tetrahedral network to a dense interpenetrated network. The NN potential also reproduces very well properties for which it was not explicitly trained, such as dynamical properties and the structure of the stable crystalline phases of mW.

I. INTRODUCTION

Machine learning (ML) potentials represent one of the emerging trends in condensed matter physics and are revolutionizing the landscape of computational research. Nowadays, different methods to derive ML potentials have been proposed, providing a powerful methodology to model liquid and solid phases in a large variety of molecular systems.[1–17] Among these methods, probably the most successful representation of a ML potential so far is given by Neural Network (NN) potentials, where the potential energy surface is the output of a feed-forward neural network.[18–35]

In short, the idea underlying NN potentials construction is to train a neural network to represent the potential energy surface of a target system. The model is initially trained on a set of configurations generated *ad hoc*, for which the total energies and forces are known, by minimizing a suitable defined loss function based on the error in the energy and force predictions. If the training set is sufficiently broad and representative, the model can then be used to evaluate the total energy and forces of any related atomic configuration with an accuracy comparable with the original potential. Typically, the original potential will include additional degrees of freedom, such as the electron density for density functional theory (DFT) calculations, or solvent atoms in protein simulations, which make the full computation very expensive. By training the network only on a subset of the original degrees of freedom, one obtains a coarse-grained representation that can be simulated at a much-reduced computational cost. NN potentials thus combine the best of two worlds, retaining the accuracy of the underlying potential model, at the much lower cost of coarse-grained classical molecular dynamics (MD) simulations. The accuracy of the NN potential depends crucially on how local atomic positions are encoded in the input of the neural network, which needs to retain the symmetries of the underlying Hamiltonian, i.e., rotational, translational, and index permutation invariance. Several methods have been proposed in the literature,[12,36] such as the approaches based on the Behler–Parrinello (BP) symmetry functions,[18] the Smooth Overlap of Atomic Positions (SOAP),[37]

N-body iterative contraction of equivariants (NICE)[38] and polynomial symmetry functions,[39] or frameworks like the DeepMD,[23] SchNet,[22] and RuNNer.[18] In all cases, atomic positions are transformed into atomic fingerprints (AFs). The choice of the AFs is particularly relevant, as it greatly affects the accuracy and generality of the resulting NN potential. It is achieved either via physical intuition or with a feature selection algorithm[40,41] to fix the AFs parameters independently from neural network weights. Then, the parameters of the AFs are kept fixed, and only the neural network weights are optmized in the training procedure.

We develop here a fully learnable NN potential in which the AFs, although retaining the simplicity of typical local fingerprints, do not need to be fixed beforehand but instead are learned during the training procedure. The simultaneous training of the atomic fingerprint parameters and the network weights makes the NN training process more efficient since the NN representation is spontaneously built on a variable atomic fingerprint representation and eliminates arguably the most difficult step (feature selection of AFs) in setting up a neural network potential. To tackle the combined minimization of the AF parameters and of the network weights, we adopt an efficient annealing procedure, which periodically cycles the learning rate, i.e., the step size of the minimization algorithm, resulting in a fast and accurate training process.

We validate the NN potential on the mW model of water,[42] which is a one-site classical potential that has found a widespread adoption to study water's anomalies[43,44] and crystallization phenomena.[45,46] Since the first pioneering MD simulations,[47,48] water is often chosen as a prototypical case study, as the large number of distinct local structures that are compatible with its tetrahedral coordination makes it the molecule with the most complex thermodynamic behavior,[49] for example displaying a liquid–liquid critical point at supercooled conditions.[50–54] NN potentials for water have been developed starting from density functional calculations, with different levels of accuracy.[55–62] NN potentials have also been proposed to parametrize accurate classical models for water with the aim of speeding up the calculations when multi-body interactions are included,[63] as in the MBpol model[64–66] or for testing the relevance of the long-range interactions, as for the SPC/E model.[67] We choose the mW potential as our benchmark system because its explicit three-body potential term offers a challenge to the NN representation that is not found in molecular models built from pair-wise interactions. We stress that we train the NN-potential against data that can be generated easily and for which structural and dynamic properties are well known (or can be evaluated with small numerical errors) in a wide range of temperatures and densities. In this way, we can perform a quantitative accurate comparison between the original mW model and the hereby proposed NN model.

Our results show that training the NN potential at even just one density–temperature state point provides an accurate description of the mW model in a surrounding phase space region that is ~100 K wide. A training based on three different state points extends the convergence window extensively, accurately reproducing state points at extreme conditions, i.e., large negative and (crushingly) positive pressures. We will show that the NN reproduces thermodynamic, structural, and dynamical properties of the mW liquid state, as well as the structural properties of all the stable crystalline phases of mW water.

The paper is organized as follows: In Sec. II, we describe the new atomic fingerprints and the details about the Neural Network potential implementation, including the *warm restart* procedure used to train the weights and the fingerprints at the same time. In Sec. III, we present the results, which include the accuracy of the models built from training sets that include one or three state points, and a comparison of the thermodynamic, structural, and dynamic properties with those of the original mW model. In Sec. IV, we provide the conclusion.

II. THE NEURAL NETWORK MODEL

The most important step in the design of a feed-forward neural network potential is the choice of how to define the first and the last layers of the network, respectively, named the *input* and *output* layers. We start with the output layer, as it determines the NN potential architecture to be constructed. Here, we follow the Behler Parrinello NN potential architecture,[18] in which the total energy of the system is decomposed as the sum of local fields (E_i), each one representing the contribution of a local environment centered around atom i. Being this a many-body contribution, it is important to note that E_i is not the energy of the single atom i, but of all its environment (see also Appendix A). With this choice, the total energy of the system is simply the sum over all atoms, $E = \sum E_i$, and the force $\vec{f}_i$ acting on atom i is the negative gradient of the *total* energy with respect to the coordinates ν of atom i, e.g., $f_{i\nu} = -\partial E/\partial x_{i\nu}$. We have to point out that a NN potential is differentiable, and hence, it is possible to evaluate the gradient of the energy analytically. This allows us to compute forces of the NN potential in the same way as other force fields, e.g., by the negative gradient of the total potential energy.

The input layer is built from two-body (distances) and three-body (angles) descriptors of the local environment, $\vec{D}^{(i)}$ and $\boldsymbol{T}^{(i)}$, respectively, ensuring translational and rotational invariance. The first layer of the neural network is the *Atomic Fingerprint Constructor* (*AFC*), as shown in Fig. 1, which applies an exponential weighting on the atomic descriptors, restoring the invariance under permutations of atomic indices. The outputs of this first layer are the atomic fingerprints (AFs), and in turn, these are given to the first *hidden layer*. We will show how this organization of the AFC layer allows for the internal parameters of the exponential weighting to be trained together with the weights in the hidden layers of the network. In the following, we describe in detail the construction of the inputs and the calculation flow in the first layers.

A. The atomic fingerprints

The choice of the input layer presents considerably more freedom, and it is here that we deviate from previous NN potentials. The data in this layer should retain all the information needed to properly evaluate the forces and energies of the particles in the system, possibly exploiting the internal symmetries of the Hamiltonian (which in isotropic fluids are the rotational, translational, and permutational invariance) to reduce the number of degenerate inputs. Given that the output was chosen as E_i, the energy of the atomic environment surrounding atom i, the input uses an atom-centered representation of the local environment of atom i.

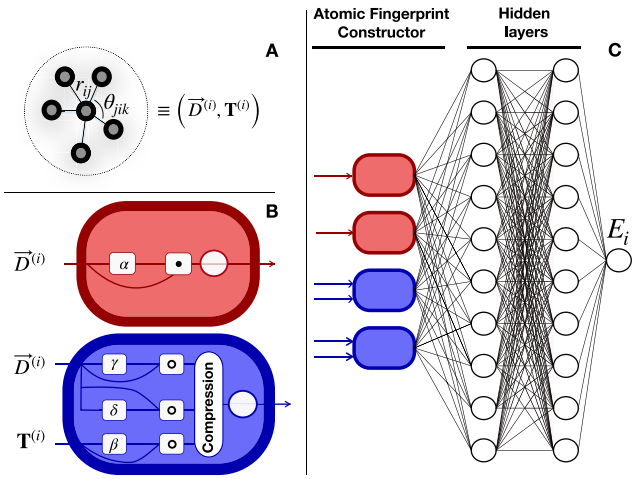

FIG. 1. Schematic representation of the Neural Network Potential flow. (a) Starting from the relative distances and the triplets angles between neighboring atoms, the input layer evaluates the atomic descriptors $\vec{D}^{(i)} = \{D_j^{(i)}\}$ [Eq. (1)] and $\boldsymbol{T}^{(i)} = \{T_{jk}^{(i)}\}$ [Eq. (2)]. (b) The first layer is the Atomic Fingerprint Constructor (AFC), which combines the atomic descriptors into atomic fingerprints, weighting them with an exponential function. The red nodes perform the calculation of Eq. (5), where from the two-body descriptors a weighting vector $\vec{D}_w^{(i)}(\alpha) = \{e^{\alpha D_j^{(i)}}\}$ is calculated (square with α) and then the scalar product $\vec{D}^{(i)} \cdot \vec{D}_w^{(i)}(\alpha)$ is computed (square with point) and finally a logarithm is applied (circle). The blue nodes perform the calculation of Eq. (7), where two weighting vectors are calculated from the two-body descriptors namely $\vec{D}_w^{(i)}(\gamma)$ and $\vec{D}_w^{(i)}(\delta)$ and one weighting matrix from the three-body descriptors $\boldsymbol{T}_w^{(i)}(\beta) = \{e^{\beta T_{jk}^{(i)}}/2\}$. Finally in the compression unit [Eq. (6)], the values are combined as $0.5[\vec{D}^{(i)} \circ \vec{D}_w^{(i)}(\gamma)]^T[\boldsymbol{T}^{(i)} \circ \boldsymbol{T}_w^{(i)}(\beta)][\vec{D}^{(i)} \circ \vec{D}_w^{(i)}(\delta)]$ where we use the circle symbol for the element-wise multiplication. The output value of the compression unit is given to the logarithm function (circle). The complete network (d) is made of ten AFC units and two hidden layers with 25 nodes per layer, and here, it is depicted 2.5 times smaller.

In the input layer, we define an atom-centered representation of the local environment of atom i, considering both the distances r_{ij} with the nearest neighbors j within a spatial cutoff R_c and the angles θ_{jik} between atom i and the pair of neighbors jk that are within a cutoff R_c'. More precisely, for each atom j within R_c from i, we calculate the following descriptors

$$D_j^{(i)}(r_{ij}; R_c) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\pi \frac{r_{ij}}{R_c}\right)\right], & r_{ij} \le R_c, \\ 0, & r_{ij} > R_c, \end{cases} \tag{1}$$

and, for each triplet $j - i - k$ within R_c' from i,

$$T_{jk}^{(i)}(r_{ij}, r_{ik}, \theta_{jik}) = \frac{1}{2}\left[1 + \cos(\theta_{jik})\right] D_j^{(i)}(r_{ij}; R_c') \times D_k^{(i)}(r_{ik}; R_c'). \tag{2}$$

Here, i indicates the label of i-th particle, whereas indices j and k run over all other particles in the system. In Eq. (1), $D_j^{(i)}(r_{ij}; R_c)$ is a function that goes continuously to zero at the cutoff (including its derivatives). The choice of this functional form guarantees that $D_j^{(i)}$ is able to express contributions even from neighbors close to the cutoff. Other choices, based on polynomials or other non-linear functions, have been tested in the past.[31] For example, we tested a parabolic cutoff function that produced considerably worse results than the cutoff function in Eq. (1). The function $T_{jk}^{(i)}(r_{ij}, r_{ik}, \theta_{jik})$ is also continuous at the triplet cutoff R_c'. The angular function $\frac{1}{2}\left[1 + \cos(\theta_{jik})\right]$ guarantees that $0 \le T_{jk}^{(i)}(r_{ij}, r_{ik}, \theta_{jik}) \le 1$. We note that the use of relative distances and angles in Eqs. (1) and (2) guarantees translational and rotational invariance.

The pairs and triplets descriptors are then fed to the *AFC layer* to compute the atomic fingerprints, AFs. These are computed by projecting the $D_j^{(i)}$ and $T_{jk}^{(i)}$ descriptors on a set of exponential functions defined by

$$\overline{D}^{(i)}(\alpha) = \ln\left[\sum_{j \ne i} D_j^{(i)} e^{\alpha D_j^{(i)}} + \epsilon\right] - Z_\alpha, \tag{3}$$

$$\overline{T}^{(i)}(\beta, \gamma, \delta) = \ln\left[\sum_{j \ne k \ne i} \frac{T_{jk}^{(i)} e^{\beta T_{jk}^{(i)}} e^{\gamma D_j^{(i)}} e^{\delta D_k^{(i)}}}{2} + \epsilon\right] - Z_{\beta\gamma\delta}. \tag{4}$$

These AFs are built summing over all pairs and all triplets involving particle i, making them invariant under permutations and multiplying each descriptor by an exponential filter whose parameters are called α for distance AFs and β, γ, δ for the triplet AFs. These parameters play the role of feature selectors, i.e., by choosing an appropriate list of $\alpha, \beta, \gamma, \delta$, the AFs can extract the necessary information from the atomic descriptors. The best choice of $\alpha, \beta, \gamma, \delta$ will emerge automatically during the training stage. In Eqs. (3) and (4), the number ϵ is set to 10^{-3} and fixes the value of energy in the rare event that no neighbors are found inside the cutoff. Parameters Z_α and $Z_{\beta\gamma\delta}$ are optimized during the training process, shifting the AFs toward positive or negative values, and act as normalization factors that improve the representation of the NN.

The definitions in Eqs. (3) and (4) can be reformulated in terms of product between vectors and matrices in the following way. The descriptors in Eqs. (1) and (2) for particle i can be represented as a vector $\vec{D}^{(i)} = \{D_j^{(i)}\}$ and a matrix $\boldsymbol{T}^{(i)} = \{T_{jk}^{(i)}\}$, respectively. Given a choice of α, β, γ and δ, three weighting vector $\vec{D}_w^{(i)}(\alpha) = \{e^{\alpha D_j^{(i)}}\}$, $\vec{D}_w^{(i)}(\gamma) = \{e^{\gamma D_j^{(i)}}\}$, and $\vec{D}_w^{(i)}(\delta) = \{e^{\delta D_j^{(i)}}\}$ and one weighting matrix $\boldsymbol{T}_w^{(i)}(\beta) = \{e^{\beta T_{jk}^{(i)}}/2\}$ are calculated from $\vec{D}^{(i)}$ and $\boldsymbol{T}^{(i)}$. The two-body atomic fingerprint [Eq. (3)] is finally computed as

$$\overline{D}^{(i)}(\alpha) = \ln\left[\vec{D}^{(i)} \cdot \vec{D}_w^{(i)}(\alpha) + \epsilon\right] - Z_\alpha. \tag{5}$$

The three-body atomic fingerprint [Eq. (4)] is computed first by what we call the *compression* step in Fig. 1 as

$$\overline{T}_c^{(i)} = \frac{[\vec{D}^{(i)} \circ \vec{D}_w^{(i)}(\gamma)]^T[\boldsymbol{T}^{(i)} \circ \boldsymbol{T}_w^{(i)}(\beta)][\vec{D}^{(i)} \circ \vec{D}_w^{(i)}(\delta)]}{2}, \tag{6}$$

and finally by

$$\overline{T}^{(i)}(\beta, \gamma, \delta) = \ln\left[\overline{T}_c^{(i)}(\beta, \gamma, \delta) + \epsilon\right] - Z_{\beta\gamma\delta}, \tag{7}$$

where we use the circle symbol for the element-wise multiplication. The NN potential flow is depicted in Fig. 1 following the vectorial representation.

In summary, our AFs select the local descriptors useful for the reconstruction of the potential by weighting them with an exponential factor tuned with exponents $\alpha, \beta, \gamma, \delta$. A similar weighting procedure has been shown to be extremely powerful in the selection of complex patterns and is widely applied in the so-called *attention layer* first introduced by Google Brain.[68] However, the AFC layer imposes additionally physically motivated constraints on the neural network representation.

We note that the expression for the system energy is a sum over the fields E_i, but the local fields E_i are not additive energies, involving all the pair distances and triplets angles within the cutoff sphere centered on particle i. This non-additive feature favors the NN ability to capture higher-order correlations (multi-body contribution to the energy) and has been shown to outperform additive models in complex datasets.[69] The NN non-additivity requires the derivative of the whole energy E (as opposed to E_i) to estimate the force on particle i. In this way, contributions to the force on particle i come not only from the descriptors of i but also from the descriptors of all particles who have i as a neighbor, de facto enlarging the effective region in space where interactions between the particles are included. This allows the network to include contributions from length scales larger than the cutoffs that define the atomic descriptors. Appendix A provides further information on this point.

B. Hidden layers

We employ a standard feed-forward fully connected neural network composed of two hidden layers with 25 nodes per layer and use the hyperbolic tangent (tanh) as the activation function. The nodes of the first hidden layer are fully connected to the ones in the second layer, and these connections have associated weights W that are optimized during the training stage.

The input of the first hidden layer is given by the AFC layer where we used five nodes for the two-body AFs [Eq. (3)] and five nodes for the three-body AFs [Eq. (4)] for a total of 10 AFs for each atom. We explore the performance of some combinations for the number of two-body and three-body AF in Appendix B, and we find that the choice of five and five is the best compromise between accuracy and computational cost.

The output is the local field E_i, for each atomic environment i, whose sum $E = \sum_{i=1}^{N} E_i$ represents the NN estimate of the potential energy E of the whole system.

C. Loss function and training strategy

To train the NN potential, we minimize a loss function computed over n_f frames, i.e., the number of independent configurations extracted from an equilibrium simulation of the liquid phase of the target potential (in our case the mW potential). The loss function is the sum of two contributions.

The first contribution, $H[\{\Delta\epsilon^k, \Delta f_{iv}^k\}]$, expresses the difference in each frame k between the NN estimates and the target values for both the total potential energy (normalized by total number of atoms) ϵ^k and the atomic forces f_{iv}^k acting in direction v on atom i. The n_f energy ϵ^k values and $3Nn_f$ force f_{iv}^k values are combined in the following expression:

$$H[\{\Delta\epsilon^k, \Delta f_{iv}^k\}] = \frac{p_e}{n_f}\sum_{k=1}^{n_f} h_{\text{Huber}}(\Delta\epsilon^k) + \frac{p_f}{3Nn_f}\sum_{k=1}^{n_f}\sum_{i=1}^{N}\sum_{v=1}^{3} h_{\text{Huber}}(\Delta f_{iv}^k), \tag{8}$$

where $p_e = 0.1$ and $p_f = 1$ control the relative contribution of the energy and the forces to the loss function, and $h_{\text{Huber}}(x)$ is the so-called Huber function,

$$h_{\text{Huber}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| \le 1, \\ 0.5 + (|x| - 1) & \text{if } |x| > 1, \end{cases} \tag{9}$$

p_e and p_f are hyper-parameters of the model, and we selected them with some preliminary tests that found those values to be near the optimal ones. The Huber function[70] is an optimal choice whenever the exploration of the loss function goes through large errors caused by outliers, i.e., data points that differ significantly from previous inputs. Indeed, when a large deviation between the model and data occurs, a mean square error minimization may give rise to an anomalous trajectory in parameters space, largely affecting the stability of the training procedure. This may happen especially in the first part of the training procedure when the parameter optimization, relaxing both on the energy and forces error surfaces may experience some instabilities.

The second contribution to the loss function is a regularization function, $R[\{\alpha^l, \beta^m, \gamma^m, \delta^m\}]$, that serves to limit the range of positive values of α^l and of the triplets $\beta^m, \gamma^m, \delta^m$ (where the indices l and m run over the five different values of α and five different triplets of values for β, γ, and δ) in the window $-\infty$ to 5. To this aim, we select the commonly used ReLU function,

$$r_{\text{relu}}(x) = \begin{cases} x - 5 & \text{if } x > 5, \\ 0 & \text{if } x \le 5, \end{cases} \tag{10}$$

and write

$$R[\{\alpha^l, \beta^m, \gamma^m, \delta^m\}] = \sum_{l=1}^{5} r_{\text{relu}}(\alpha^l) + \sum_{m=1}^{5}\left[r_{\text{relu}}(\beta^m) + r_{\text{relu}}(\gamma^m) + r_{\text{relu}}(\delta^m)\right]. \tag{11}$$

Thus, the R function is activated whenever one parameters of the *AFC layer* becomes, during the minimization, larger than 5.

To summarize, the global loss function $\mathcal{L}$ used in the training of the NN is

$$\mathcal{L}[\epsilon, f] = H[\{\Delta\epsilon^k, \Delta f_{iv}^k\}] + p_b R[\{\alpha^l, \beta^m, \gamma^m, \delta^m\}], \tag{12}$$

where $p_b = 1$ weights the relative contribution of R compared with H.

Compared with a standard NN-potential, we train not only the network weights W but also the AFs parameters $\Sigma \equiv \{\alpha^l, \beta^m, \gamma^m, \delta^m\}$ at the same time. The simultaneous optimization of the weights W and AFs Σ prevents possible bottlenecks in the optimization of W at a fixed representation of Σ. Other NN potential approaches implement a separate initial procedure to optimize the Σ parameters

followed by the optimization of W at fixed Σ.[40] The two-step procedure (TSP) not only requires a specific methodological choice for optimizing Σ but also may not result in the optimal values, compared with a search in the full parameter space (i.e., both Σ and W). In Appendix C, we compare our approach with a popular two-step procedure.[40] Since the complexity of the loss function has increased, we have investigated in some detail some efficient strategies that lead to fast and accurate trainings. First, we initialize the parameters W via the Xavier algorithm, in which the weights are extracted from a random uniform distribution.[71] To initialize the Σ parameters, we used a uniform distribution in interval $[-5, 5]$. We then minimize the loss function using the *warm restart procedure* proposed in Ref. 72. In this procedure, the learning rate η is reinitialized at every cycle l, and inside each cycle, it decays as a function of the number of training steps t following equation:

$$\eta^{(l)}(t) = A_l\left\{\frac{(1-\xi_f)}{2}\left[1+\cos\left(\frac{\pi t}{T_l}\right)\right]+\xi_f\right\} \quad 0 \le t \le T_l, \quad (13)$$

where $\xi_f = 10^{-7}$, $A_l = \eta_0\xi_0^l$ is the initial learning rate of the l-th cycle with $\eta_0 = 0.01$ and $\xi_0 = 0.9$, and $T_l = b\tau^l$ is the period of the l-th cycle with $\tau = 1.4$ and $b = 40$. The absolute number of training steps n during cycle l can be calculated summing over the length of all previous cycles as $n = \tau + \sum_{m=0}^{l-1} T_m$.

We also decided to evaluate the loss function for groups of four frames (mini-batch), and we randomly select 200 frames $n_f = 200$ for a system of 1000 atoms, and hence, we split this dataset in 160 frames (%80) for the training set and the 40 frames (%20) for the test set.

In Fig. 2(a) we represent the typical decay of the learning rate of the warm restart procedure, which will be compared to the standard exponential decay protocol in the Results section.

D. The target model

To test the quality of the proposed NN, we train the NN with data produced with the mW[42] model of water. This potential, a reparametrization of the Stillinger–Weber model for silicon,[73] uses a combination of pairwise functions complemented with an additive three-body potential term,

$$E = \sum_i\sum_{j>i} U_2(r_{ij}) + \lambda\sum_i\sum_{j\neq i}\sum_{j>k} U_3(r_{ij}, r_{ik}, \theta_{jik}), \quad (14)$$

where the two-body contribution between two particles i and j at a relative distance of r_{ij} is a generalized Lennard-Jones potential,

$$U_2(r_{ij}) = A\epsilon\left[B\left(\frac{\sigma}{r_{ij}}\right)^p - \left(\frac{\sigma}{r_{ij}}\right)^q\right]\exp\left(\frac{\sigma}{r_{ij}-a\sigma}\right), \quad (15)$$

where the $p = 12$ and $q = 6$ powers are substituted by $q = 0$ and $p = 4$, multiplied by an exponential cutoff that brings the potential to zero at $a\sigma$, with $a = 1.8$ and $\sigma = 2.3925$ Å. $A\epsilon$ (with $A = 7.050$ and $\epsilon = 6.189$ kcal mol^{-1}) controls the strength of the two body part. B controls the two-body repulsion (with $B = 0.602$).

The three-body contribution is computed from all possible ordered triplets formed by the central particle with the interacting neighbors (with the same cutoff $a\sigma$ as the two-body term) and favors the tetrahedral coordination of the atoms via the following functional form,

$$U_3(r_{ij}, r_{ik}, \theta_{jik}) = \epsilon\left[\cos(\theta_{jik}) - \cos(\theta_0)\right]^2 \exp\left(\frac{\gamma\sigma}{r_{ij}-a\sigma}\right) \times \exp\left(\frac{\gamma\sigma}{r_{ik}-a\sigma}\right), \quad (16)$$

where θ_{jik} is the angle formed in the triplet jik, and $\gamma = 1.2$ controls the smoothness of the cutoff function on approaching the cutoff. Finally, $\theta_0 = 109.47^\circ$ and $\lambda = 23.15$ control the strength of the angular part of the potential.

The mW model, with its three-body terms centered around a specific angle and non-monotonic radial interactions, is based on

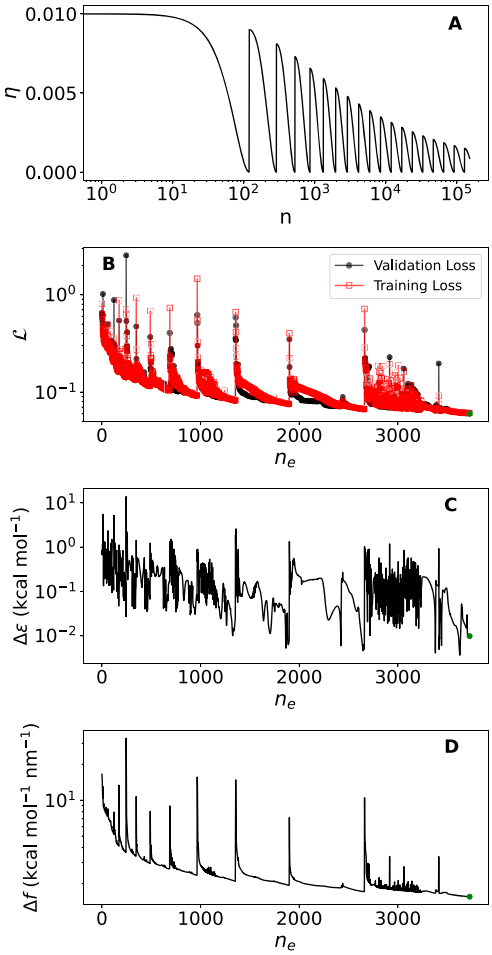

FIG. 2. Model convergence properties: (a) Learning rate schedule [Eq. (13)] as a function of the absolute training step n (one step is defined as an update of the network parameters). (b) The training and validation loss [see $\mathcal{L}[\epsilon, f]$ in Eq. (12)] evolution during the training procedure, reported as a function of the number of epoch n_e (an epoch is defined as a complete evaluation of the training dataset). Root mean square (rms) error of the total potential energy per particle (c) and of the force Cartesian components (d) during the training evaluated in the test dataset. Data in panels (b)–(d) refer to the NN3 model, and the green point shows the best model location.

a functional form that is quite different from the radial and angular descriptors selected in the NN model. The NN is thus agnostic with respect to the functional form that describes the physical system (the mW in this case). However, having a reference model with explicit three-body contributions offers a more challenging target for the NN potential compared with potential models built entirely from pairwise interactions. The mW model is thus an excellent candidate to test the performance of the proposed NN potential.

III. RESULTS

A. Training

We study two different NN models, indicated with the labels NN1 and NN3, differing in the number of state points included in the training set. These two models are built with a cutoff of $R_c = 4.545$ Å for the two-body atomic descriptors and a cutoff of $R_c' = 4.306$ Å for the three-body atomic descriptors. R_c' is the same as the mW cutoff, whereas R_c was made slightly larger to mitigate the suppression of information at the boundaries by the cutoff functions. The NN1 model uses only training information based on mW equilibrium configurations from one state point at $\rho_1 = 1.07$ g cm^{-3}, $T_1 = 270.9$ K where the stable phase is the liquid. The NN3 model uses training information based on mW liquid configurations in three different state points, two state points at $\rho_1 = 0.92$ g cm^{-3}, $T_1 = 221.1$ K, and $\rho_2 = 0.92$ g cm^{-3}, $T_2 = 270.9$ K, where the stable solid phase is the clathrate Si34/Si136[74] and one state point at $\rho_3 = 1.15$ g cm^{-3}, $T_2 = 270.9$ K.

This choice of points in the phase diagram is aimed to improve agreement with the low temperature-low density as well as high-density regions of the phase diagram. Importantly, all configurations come from either stable or metastable liquid state configurations. Indeed, the point at $\rho_2 = 0.92$ g cm^{-3}, $T_2 = 270.9$ K is quite close to the limit of stability (respect to cavitation) of the liquid state.

To generate the training set, we simulate a system of $N = 1000$ mW particles with a standard molecular dynamics code in the NVT ensemble, where we use a time step of 4 fs and run 10^7 steps for each state point. From these trajectories, we randomly select 200 configurations (frames) to create a dataset of positions, total energies, and forces. We then split the dataset in the *training* and in the *test* datasets, the first one containing 80% of the data. We then run the training for 4000 epochs with a minibatch of 4 frames. At the end of every epoch, we check if the validation loss is improved and we save the model parameters. In Fig. 2, we plot the loss function for the training and test datasets (b), the root mean square error of the total energy per particle (c), and of the force (d) for the NN3 model. The results show that the learning rate schedule of Eq. (13) is very effective in reducing both the loss and error functions.

Interestingly, the neural network seems to avoid overfitting (i.e., the validation loss is decreasing at the same rate as the loss on the training data), and the best model (deepest local minimum explored), in a given window of training steps, is always found at the end of that window, which also indicates that the accuracy could be further improved by running more training steps. Indeed, we found that by increasing the number of training steps by one order of magnitude, the error in the forces decreases further by 30%. Similar accuracy of the training stage is obtained also for the NN1 model (not shown).

The training procedure always terminates with an error on the test set equal or less than $\Delta\epsilon \simeq 0.01$ kcal mol^{-1} (0.43 meV) for the energy and $\Delta f \simeq 1.55$ kcal mol^{-1} nm^{-1} (6.72 meV Å^{-1}) for the forces. These values are comparable with the state-of-the-art NN potentials[23,56,57,63] and within the typical accuracy of DFT calculations.[75]

We can compare the precision of our model with that of alternative NN potentials trained on a range of water models. In Ref. 24, a neural network potential was trained on the mW model from a dataset made of 1991 configurations of 128 particles at different pressures and temperatures (including both liquid and ice structures) with Behler–Parinello symmetry functions. The training of this model (which uses more atomic fingerprints and a larger cutoff radius) converged to an error in the energy of $\Delta\epsilon \simeq 0.0062$ kcal mol^{-1} (0.27 meV) and $\Delta f \simeq 3.46$ kcal mol^{-1} nm^{-1} (15.70 meV Å^{-1}) for the forces. In a recent study searching for liquid–liquid transition signatures in an *ab initio* water NN model,[57] a dataset of configurations spanning a temperature range of 0–600 K and a pressure range of 0–50 GPa was selected. For a system of 192 particles, the training converged to an error in the energy of $\Delta\epsilon \simeq 0.010$ kcal mol^{-1} (0.46 meV) and $\Delta f \simeq 9.96$ kcal mol^{-1} nm^{-1} (43.2 meV Å^{-1}) for the forces. In the NN model of MB-POL,[63] a dataset spanning a temperature range from 198 to 368 K at ambient pressure was selected. In this case, for a system of 256 water molecules, an accuracy of $\Delta\epsilon \simeq 0.01$ kcal mol^{-1} (0.43 meV) and $\Delta f \simeq 10$ kcal mol^{-1} nm^{-1} (43.36 meV Å^{-1}) was reached. Finally, the NN for water at $T = 300$ K used in Ref. 56, reached precisions of $\Delta\epsilon \simeq 0.046$ kcal mol^{-1} (2 meV) and $\Delta f \simeq 25.36$ kcal mol^{-1} nm^{-1} (110 meV Å^{-1}). Although a direct comparison between NN potentials trained on different reference potentials is not a valid test to rank the respective accuracies, the comparisons above show that our NN potential reaches a similar precision in energies and possibly an improved error in the force estimation. Moreover, our results suggest that the difficulty to properly reproduce a two- and three-body potential that can be calculated analytically can be comparable with the difficulty to represent DFT calculations, which may suffer from intrinsic statistical errors/noise.

The accuracy of the NN potential could be further improved by extending the size of the dataset and the choice of the state points. In fact, although the datasets in Refs. 56, 57, and 63 have been built with optimized procedures, the dataset used in this study was prepared by sampling just one (NN1) or three (NN3) state points. Also, the size of the datasets used in the present work is smaller or comparable with the ones of Refs. 56, 57, and 63.

In Appendix C, we compare the simultaneous training of AFs and NN weights with a two-step procedure in which the AFs are preselected according to the Farthest Point Sampling (FPS) method.[40] The results show that the simultaneous training of AFs leads to a more accurate training, even when using an overall small number of AFs.

In Fig. 3, we compare the error in the energies (a) and the forces (b) between 60 independent training runs using the standard exponential decay of the learning rate (points) and the warm restart protocol (squares). The figure shows that although the errors in the energy computations are comparable between the two methods, the warm restart protocol allows the forces to be computed with higher accuracy. Moreover, we found that the warm restart procedure is less dependent on the initial seed and that it reaches deeper basins than the standard exponential cooling rate.

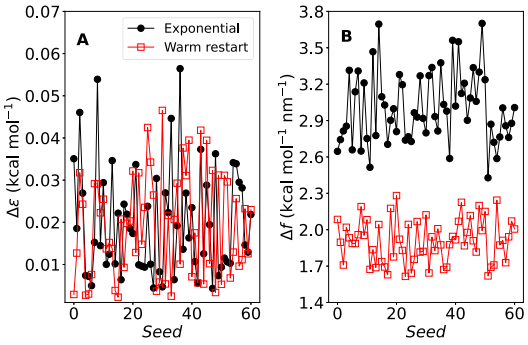

FIG. 3. Comparison of the root mean square error calculated on the validation set for 60 replicas differing in the initial seed of the training procedure using both an exponential decay of the learning rate (points) and the warm restart method (squares) for the energy [panel (a)] and the forces [panel (b)]. For the forces, a significant improvement both in the average error and in its variance is found for the warm restart schedule.

B. Comparing NN1 with NN3

The NN potential model was implemented in a custom MD code that makes use of the TensorFlow C API.[76] We adopted the same time step (4 fs), the same number of particles ($N = 1000$) and the same number of steps (10^7) as for the simulations in the mW model.

As described in the Training Section, we compare the accuracy of two different training strategies: NN1, which was trained on a single state point, and NN3, which is instead trained on three different state points. In Fig. 4, we plot the energy error ($\Delta\epsilon$) between the NN potential and the mW model with both NN1 [panel (a)] and NN3 [panel (b)]. Starting from NN1, we see that the model already provides excellent accuracy for a large range of temperatures and for densities close to the training density. The biggest shortcoming of the NN1 model is at densities lower than the trained density, where the NN potential model cavitates and does not retain the long-lived metastable liquid state displayed by the mW model. We speculate that this behavior is due to the absence of low-density configurations in the training set, which prevents the NN potential model from correctly reproducing the attractive tails of the mW potential.

To overcome this limitation, we have included two additional state points at a low density in the NN3 model. In this case, Fig. 4(b) shows that NN3 provides a quite accurate reproduction of the energy in the entire explored density and temperature window (despite being trained only with data at $\rho = 0.92$ g cm^{-3} and $\rho = 1.15$ g cm^{-3}).

We can also compare the accuracy obtained during production runs against the accuracy reached during training, which was $\Delta\epsilon \simeq 0.01$ kcal mol^{-1}. Figure 4(b) shows the error is of the order of 0.032 kcal mol^{-1} (1.3 meV), for density above the training set density. However, in the density region between 0.92 and 1.15, the error is even smaller, around 0.017 kcal mol^{-1} (0.7 meV) at the lowest density boundary.

We can thus conclude that the NN3 model, which adds to the NN1 model information at lower density and temperature in the region where tetrahedality in the water structure is enhanced, is indeed capable to represent, with only three state points, a quite large region of the phase space, encompassing dense and stretched liquid states. This suggests that a training based on few state points at the boundary of the density/temperature region that needs to be studied is sufficient to produce a high-quality NN model. In the following, we focus entirely on the NN3 model.

C. Comparison of thermodynamic, structural, and dynamical quantities

In Fig. 5, we present a comparison of thermodynamic data between the mW model (squares) and its NN potential representation (points) across a wide range of state points. Figure 5(a) plots the energy as a function of density for temperatures ranging from melting to deeply supercooled conditions. Perhaps, the most interesting result is that the NN potential is able to capture

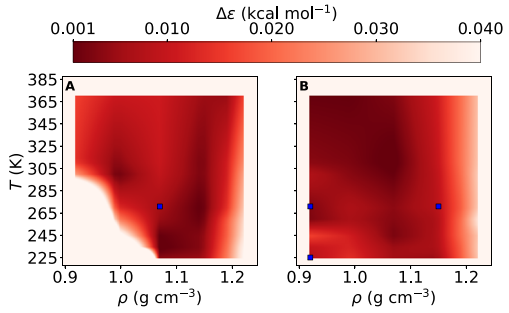

FIG. 4. Comparison between the mW total energy and the NN1 model (a) and NN3 model (b) for different temperatures and densities. Although the NN3 model is able to reproduce the mW total energy with good agreement in a wide region of densities and temperatures, the NN1 provide a good representation only in a limited region of density and temperature values. Blue squares represent the state points used for building the NN models.

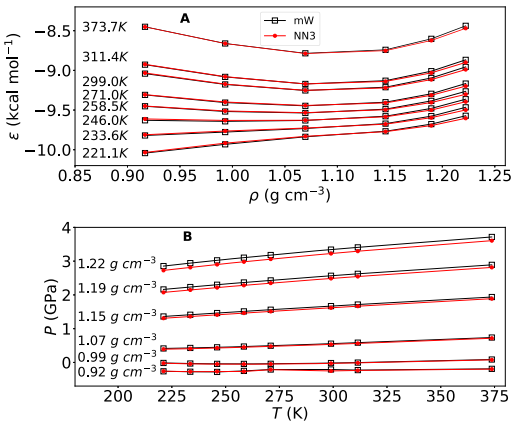

FIG. 5. Comparison between the mW total energy and the NN3 total energy as a function of density along different isotherms (a) and comparison between the mW pressure and the NN3 pressure as a function of temperature along different isochores (b). The relative error of the NN vs the mW potential grows with density but remains within 3% even for densities larger than the densities used in the training set.

the energy minimum, also called the *optimal network forming density*, which is a distinctive anomalous property of water and other *empty liquids*.[77]

Figure 5(b) shows the pressure as a function of the temperature for different densities, comparing the mW with the NN3 model. Also, the pressure shows good agreement between the two models in the region of densities between $\rho = 0.92$ g cm^{-3} and $\rho = 1.15$ g cm^{-3}, which, as for the energy, tends to deteriorate at $\rho = 1.22$ g cm^{-3}.

In the large-density region explored, the structure of the liquid changes considerably. On increasing density, a transition from tetrahedral-coordinated local structure, prevalent at low T and low ρ, toward denser local environments with interstitial molecules included in the first coordination shell takes place. This structural change is well displayed in the radial distribution function, shown for different densities at a fixed temperature in Fig. 6. Figure 6 also shows the progressive onset of a peak around 3.5 Å developing on increasing pressure, which signals the growth of interstitial molecules, coexisting with open tetrahedral local structures.[78,79] At the highest density, the tetrahedral peak completely merges with the interstitial peak. The NN3 model reproduces quite accurately all features of the radial distribution functions, maxima and minima positions, and their relative amplitudes, at all densities, from the tetrahedral-dominated to the interstitial-dominated limits. In general, the NN3 model reproduces quite well the mW potential in energies, pressures, and structures, and it appreciably deviates from mW pressure and energy quantities only at densities (above 1.15 g/cm^3) that are outside of the training region.

To assess the ability of NN potential to correctly describe also the crystal phases of the mW potential, we compare in Fig. 7 the $g(r)$ of mW with the $g(r)$ of the NN3 model for four different stable solid phases:[74] hexagonal and cubic ice ($\rho = 1.00$ g cm^{-3} and $T = 246$ K), the dense crystal SC16 ($\rho = 1.20$ g cm^{-3} and $T = 234$ K), and the clathrate phase Si136 ($\rho = 0.80$ g cm^{-3} and $T = 221$ K). The results, shown in Fig. 7, show that despite no crystal configurations having been included in the training set, a quite accurate representation of the crystal structure at finite temperature is provided by the NN3 model for all distinct sampled lattices.

Finally, we compare in Fig. 8 the diffusion coefficient (evaluated from the long time limit of the mean square displacement) for the mW and the NN3 model in a wide range of temperatures and densities, where water displays a diffusion anomaly. Figure 8 shows again that also for dynamical quantities, the NN potential offers an excellent representation of the mW potential, despite the fact that no dynamical quantity was included in the training set. A comparison between fluctuations of energy and pressure of mW and NN3 potential is reported in Appendix D.

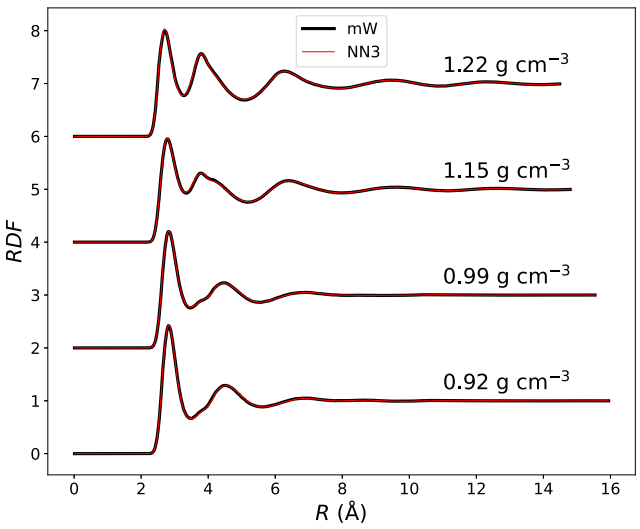

FIG. 6. Comparison between the mW radial distribution functions $g(r)$ and the NN3 $g(r)$ at $T = 270.9$ K for four different densities. The tetrahedral structure (signaled by the peak at 4.54 Å) progressively weakens in favor of an interstitial peak progressively growing at 3.5–3.8 Å. Different $g(r)$ have been progressively shifted by two to improve clarity.

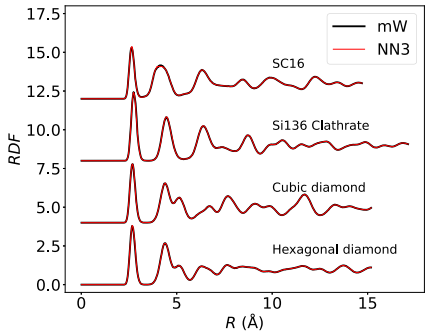

FIG. 7. Comparison between the mW radial distribution functions $g(r)$ and the NN3 $g(r)$ for four different lattices: (a) hexagonal diamond (the oxygen positions of the ice I_h), (b) cubic diamond (the oxygen positions of the ice I_c, (c) the SC16 crystal (the dense crystal form stable at large pressures in the mW model), and (d) the Si136 clathrate structure, which is stable at negative pressures in the mW model. Different $g(r)$ have been progressively shifted by four to improve clarity.

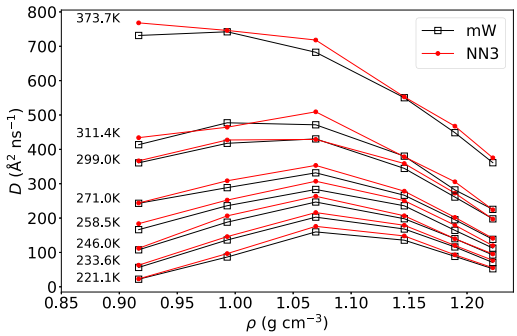

FIG. 8. Comparison between the mW diffusion coefficient D and the NN3 corresponding quantity for different temperatures and densities in the interval 221–271 K. In this dynamic quantity, the relative error is, for all temperatures, around 8%. Note also that in this T window, the diffusion coefficient shows a clear maximum, reproducing one of the well-know diffusion anomalies of water. Diffusion coefficients have been calculated in the NVT ensemble using the same Andersen thermostat algorithm[80] for mW and NN3 potential.

IV. CONCLUSIONS

In this work, we have presented a novel neural network (NN) potential based on a new set of atomic fingerprints (AFs) built from two- and three-body local descriptors that are combined in a permutation-invariant way through an exponential filter [see Eq. (3) and (4)]. One of the distinctive advantages of our scheme is that the AF's parameters are optimized during the training procedure, making the present algorithm a self-training network that automatically selects the best AFs for the potential of interest. Indeed, this scheme eliminates the feature selection step, so that a NN potential can be trained with a unified procedure. Moreover, this scheme improves the convergence of the training, allowing for better accuracy and/or the use of a smaller number of AFs (see Appendix C).

We have shown that the added complexity in the concurrent training of the AFs and NN weights can be overcome with an annealing procedure based on the warm restart method,[72] where the learning rate goes through damped oscillatory ramps. This strategy not only gives better accuracy compared with the commonly implemented exponential learning rate decay but also allows the training procedure to converge rapidly independently from the initialization strategies of the model's parameters.

Moreover, we show in Appendix E that the potential hypersurface of the NN model has the same smoothness as the target model, as confirmed by (i) the possibility to use the same time step in the NN and in the target model when integrating the equation of motion and (ii) by the possibility of simulating the NN model even in the NVE ensemble with proper energy conservation.

We test the novel NN on the mW model,[42] a one-component model system commonly used to describe water in classical simulations. This model, a re-parametrization of the Stillinger–Weber model for silicon,[73] although treating the water molecule as a simple point, is able to reproduce the characteristic tetrahedral local structure of water (and its distortion on increasing density) via the use of three-body interactions. Indeed, water changes from a liquid of tetrahedrally coordinated molecules to a denser liquid, in which a relevant fraction of interstitial molecules is present in the first nearest-neighbor shell. The complexity of the mW model, both due to its functional form as well as to the variety of different local structures that characterize water, makes it an ideal benchmark system to test our NN potential.

We find that training based on configurations extracted by three different state points is able to provide a very accurate representation of the mW potential hypersurface, when the densities and temperatures of the training state points delimit the region in which the NN potential is expected to work. We also find that the error in the NN estimate of the total energy is low, always smaller than 0.03 kcal mol^{-1}, with a mean error of 0.013 kcal mol^{-1}. The NN model reproduces very well not only the thermodynamic properties but also the structural properties, as quantified by the radial distribution function, and the dynamic properties, as expressed by the diffusion coefficient, in the extended density interval from $\rho = 0.92$ g cm^{-3} to $\rho = 1.22$ g cm^{-3}.

Interestingly, we find that the NN model, trained only on disordered configurations, is also able to properly describe the radial distribution of the ordered lattices that characterize the mW phase diagram, encompassing the cubic and hexagonal ices, SC16, and Si136 clathrate structures.[74] In this respect, the ability of the NN model to properly represent crystal states suggests that, in the case of the mW and as such probably in the case of water, the geometrical information relevant to the ordered structures is contained in the sampling of phase space typical of the disordered liquid phase. These findings have been recently discussed in Ref. 81 where it has been demonstrated that liquid water contains all the building blocks of diverse ice phases.

We conclude by noticing that the present approach can be generalized to multicomponent systems, following the same strategy implemented by previous approaches.[18,23] Work in this direction is under way.

ACKNOWLEDGMENTS

F.G.M. and J.R. acknowledge support from the European Research Council Grant No. DLV-759187 and CINECA grant ISCRAB NNPROT. F.G.M. thanks Aldo Glielmo for insightful discussions.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Francesco Guidarelli Mattioli: Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Software (equal); Supervision (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Francesco Sciortino**: Conceptualization (equal); Investigation (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **John Russo**: Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Software (equal); Supervision (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

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

APPENDIX A: THE RANGE OF THE NN POTENTIAL

In this appendix, we discuss the effective spatial range covered by a NN potential whose fingerprints are defined based on pair information confined within a sphere of cutoff radius R_c.

As noted in Ref. 31, multi-body potentials and especially non-additive multibody potentials induce local interactions beyond the cutoff radius, enlarging the sphere of interaction. Indeed, the force on particle i comes from the derivative of the local field of i and of all its neighbors with respect to the coordinates of particle i.

Figure 9 graphically explains the effective role of R_c in the NN potential. In panel (a), we describe particle 1 with only one neighbor (particle 2) within R_c. We also represent the sphere centered on particle 3, which also includes particle 2 as one of its neighbor. In

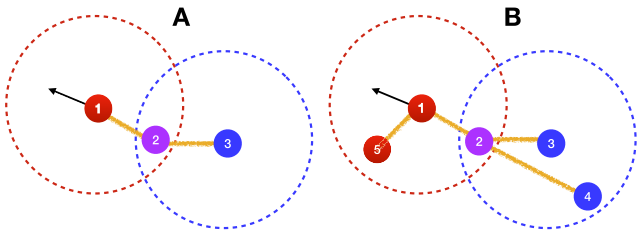

FIG. 9. (a) Two-body interactions and (b) three-body interactions in a non-linear local field model E_i. The non-linearity of the local field enlarges the interaction cut-off where a neighbor particle (blue) makes a bridge between non-neighboring particle (red and blue).

this case, the energy of the system will be represented as a sum over the local fields E_1, E_2 and E_3. Due to the intrinsic non-linearity of the NN, the field E_i mixes together the AFs, and consequently, the distances and angles entering in the AFs are non-linearly mixed in E_i. The force on atom 1 is then written as

$$f_{1v} = -\frac{\partial E_1(r_{12})}{\partial x_{1v}} - \frac{\partial E_2(r_{21}, r_{23})}{\partial x_{1v}} = -\frac{\partial E_1(r_{12})}{\partial x_{1v}} - \frac{\partial E_2(r_{21}, r_{23})}{\partial r_{21}}\frac{\partial r_{21}}{\partial x_{1v}} - \frac{\partial E_2(r_{21}, r_{23})}{\partial r_{23}}\frac{\partial r_{23}}{\partial x_{1v}} \quad \text{(A1)}$$

Although the last term vanishes, the next to the last retains an intrinsic dependence on the coordinates both of particle 2 as well as of particle 3, if the local field E_2 is non-linear. Thus, even if particle 3 is further than R_c, it enters in the determination of the force acting on particle 1. A similar effect is also present in the angular part of the AFs, as shown graphically in panel (b). Indeed, for the angular component of the AF, the force on particle 1 is

$$f_{1v} = -\frac{\partial E_1(\theta_{512})}{\partial x_{1v}} - \frac{\partial E_2(\theta_{123}, \theta_{124}, \theta_{324})}{\partial x_{1v}}. \quad \text{(A2)}$$

Also, in this case, two contributions can be separated: (i) the interaction of particle 1 with triplets 123 and 124 is an effect of the three-body AF and it is present also in additive-models such as the mW model and (ii) the interaction of particle 1 with triplet 324 is an effect of the non-additive nature of the NN local field E_i.

APPENDIX B: ACCURACY WITH VARYING THE NUMBER OF AFs

In this appendix, we investigate the efficiency of the training over different choices for the number and types of atomic fingerprints introduced in the Neural Network Model section. We start by using only one three-body ($n_{3b} = 1$) and one two-body ($n_{2b} = 1$) AF and subsequently increasing the number of the AF. For every combination of n_{2b} and n_{3b}, we run a 4000 epochs training, and at the end of each training, we extract the best model. We summarized these results in Table I where we compare the error on forces over all the investigated model. From Table I, it emerges that the choice of $n_{3b} = 5$ and $n_{2b} = 5$ is the more convenient both for accuracy and computational efficiency. Doubling the number of the three-body AF marginally improves the error on forces, whereas increases the computational cost due to the increase in the size of the input layer of the first hidden layer and due to the additional time to compute the three-body AF. Moreover, in the RESULTS section, we show that the choice $n_{3b} = 5$ and $n_{2b} = 5$ is sufficient to represent the target potential. Finally, the accuracy of the training after doubling the configurations in the dataset reaches an error on forces of $\Delta_f = 5.85$ meV Å^{-1} that is 0.87 times the error value found with a half of the dataset.

TABLE I. Table of errors on forces at the end of the 4000 epoch-long training procedure for different combination of the number and type of the AF.

n_{3b}	n_{2b}	Δf (meV Å^{-1})	n_{3b}	n_{2b}	Δf (meV Å^{-1})
1	1	72.79	5	1	16.53
1	2	67.92	5	2	7.53
1	5	56.25	5	5	6.72
1	10	56.00	5	10	6.87
1	15	56.02	5	15	6.95
2	1	53.76	10	1	7.98
2	2	43.95	10	2	7.17
2	5	32.43	10	5	5.79
2	10	32.39	10	10	6.55
2	15	24.70	10	15	6.19

APPENDIX C: COMPARISON WITH TWO-STEP TRAINING

In this appendix, we compare our simultaneous training of AFs and NN weights, with a two-step procedure (TSP) that selects AFs and trains NN weights separately. The two step-procedure is performed as follows: (1) all the parameters of the AFs in Eqs. (3) and (4) are fixed by a feature selection scheme and then (2) the neural network potential is trained by keeping frozen the previously selected AF parameters and optimizing only the hidden layer weights $\boldsymbol{W}$. Different methods have been proposed for the AF parameters selection, and we choose to use the Farthest Point Sampling (FPS).[40] Hence, we initialized the parameters for 1000 AFs, 500 for each type of AFs (two-body and three-body), and we evaluate all the 1000 AFs on a dataset of 168 configurations of $N = 1000$ particles (the same of the training set in the Results section). Next, we implement the FPS algorithm, and we select the first best 40 AFs among the initial 1000 AFs. To test this AFs selection, we train a neural network with 2 layers and 25 nodes per layer (same of NN3) with the first 5, 10, 15, 20, and 25 AFs as the input for the first hidden layer. We repeat the FPS algorithm with two independent selections of the first AF, and we show the results in Fig. 10. Panels (a) and (c) plot the Root mean square (rms) error for the forces and energies, respectively, when using up to 10 AFs. The figures show that the TSP procedure is far obtaining the accuracy of the NN3 model (which also employs 10 AFs). Moreover, for the TSP procedure there is a clear saturation of the learning curve for both forces and energies. In Fig. 10, we increase the number of AFs used in the TSP procedure and compare the results with NN3. Also, in this case, NN3 (with 10 AFs) has a better accuracy [especially for the forces, panel (b)] compared with the best model built with TSP (that uses 20 AFs).

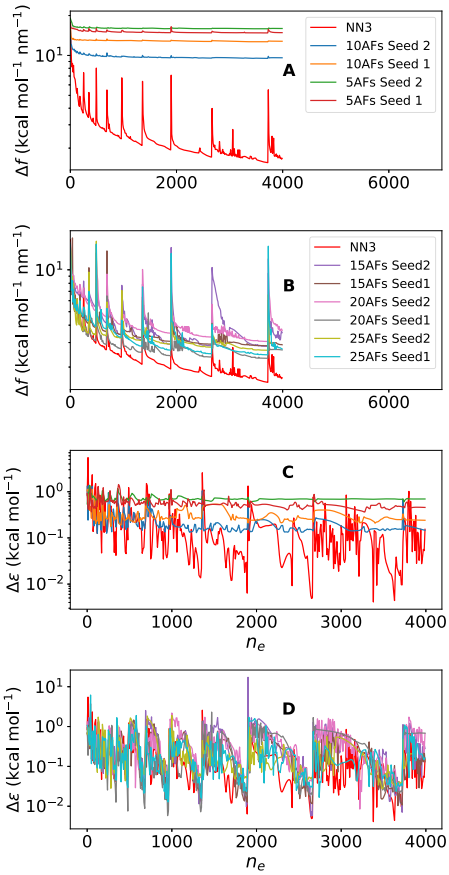

FIG. 10. Comparison of Full and two-step training procedure. (a) Forces training curve for NN3 (Full) and for the two-step by using 5 and 10 AFs and two seeds. (b) Forces training curve for NN3 (Full) and for the two-step by using 15, 20, 25, and two seeds. (c) Energy training curve for NN3 (Full) and for the Two-step by using 5 and 10 AFs and two seeds. (d) Energy training curve for NN3 (Full) and for the two-step by using 15, 20, 25, and two seeds.

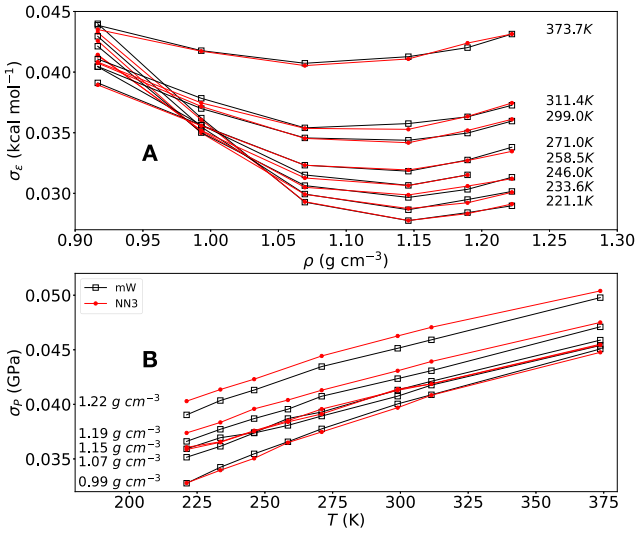

FIG. 11. (a) Standard deviation of total energy (normalized with the number of particles) and (b) standard deviation of virial pressure for both NN3 model (red) and mW model (black).

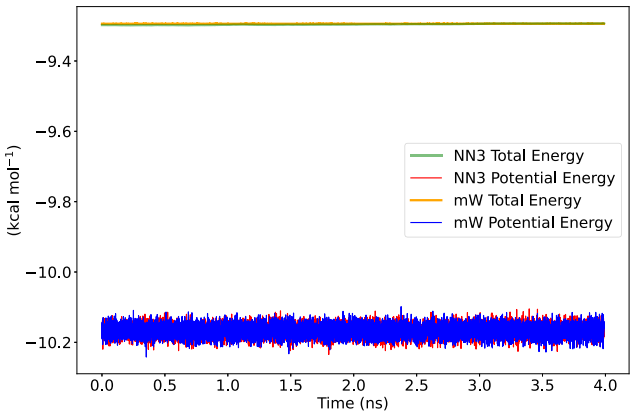

FIG. 12. NVE molecular dynamics at $T = 299$ K and $\rho = 1.07$ g cm^{-3} for both NN3 and mW model. The time step is $dt = 4$ fs for both models.

APPENDIX D: ENERGY AND PRESSURE FLUCTUATIONS

In this appendix, we provide further thermodynamics comparisons between mW and NN3 potential focusing on the pressure and energy fluctuations. We depict in Fig. 11 the standard deviations of the total energy (normalized by N) in panel (a) and the standard deviation of virial pressure in panel (b). Energy fluctuations of NN3 follow qualitatively and quantitatively the trend of mW potential. Pressure fluctuations of NN3 are in good agreement with the mW model but, as for the pressure [Fig. 5(b)], the accuracy decreases approaching state points outside the density range used for the training.

APPENDIX E: ENERGY CONSERVATION

In this appendix, we show a comparison between the mW and NN3 potentials in terms of the energy conservation in the NVE ensemble. In Fig. 12, we depict both total energy and potential energy for mW and NN3 potential. The potential energy and total energy of the two models are in good agreement.

REFERENCES

[1] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Müller, and A. Tkatchenko, J. Phys. Chem. Lett. **6**, 2326 (2015).
[2] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Sci. Adv. **3**, e1603015 (2017).
[3] M. Haghighatlari and J. Hachmann, Curr. Opin. Chem. Eng. **23**, 51 (2019).
[4] A. Glielmo, C. Zeni, Á. Fekete, and A. De Vita, *Machine Learning Meets Quantum Physics* (Springer, 2020), pp. 67–98.
[5] S. Manzhos and T. Carrington, Jr., Chem. Rev. **121**, 10187 (2020).
[6] P. Gkeka, G. Stoltz, A. Barati Farimani, Z. Belkacemi, M. Ceriotti, J. D. Chodera, A. R. Dinner, A. L. Ferguson, J.-B. Maillet, H. Minoux *et al.*, J. Chem. Theory Comput. **16**, 4757 (2020).

[7]Z. L. Glick, D. P. Metcalf, A. Koutsoukas, S. A. Spronk, D. L. Cheney, and C. D. Sherrill, J. Chem. Phys. **153**, 044112 (2020).
[8]M. Benoit, J. Amodeo, S. Combettes, I. Khaled, A. Roux, and J. Lam, Mach. Learn.: Sci. Technol. **2**, 025003 (2020).
[9]M. Dijkstra and E. Luijten, Nat. Mater. **20**, 762 (2021).
[10]O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, Chem. Rev. **121**, 10142 (2021).
[11]G. Campos-Villalobos, E. Boattini, L. Filion, and M. Dijkstra, J. Chem. Phys. **155**, 174902 (2021).
[12]F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, Chem. Rev. **121**, 9759 (2021).
[13]G. Campos-Villalobos, G. Giunta, S. Marín-Aguilar, and M. Dijkstra, J. Chem. Phys. **157**, 024902 (2022).
[14]J. Goniakowski, S. Menon, G. Laurens, and J. Lam, J. Phys. Chem. C **126**, 17456 (2022).
[15]A. Glielmo, C. Zeni, B. Cheng, G. Csányi, and A. Laio, PNAS Nexus **1**, pgac039 (2022).
[16]S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, Nat. Commun. **13**, 2453 (2022).
[17]G. Tallec, G. Laurens, O. Fresse-Colson, and J. Lam, *Quantum Chemistry in the Age of Machine Learning* (Elserview, 2023), pp. 253–277.
[18]J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[19]J. Behler, J. Phys.: Condens. Matter **26**, 183001 (2014).
[20]J. S. Smith, O. Isayev, and A. E. Roitberg, Chem. Sci. **8**, 3192 (2017).
[21]J. S. Smith, O. Isayev, and A. E. Roitberg, Sci. Data **4**, 170193 (2017).
[22]K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nat. Commun. **8**, 13890 (2017).
[23]L. Zhang, J. Han, H. Wang, W. Saidi, R. Car *et al.*, *Advances in Neural Information Processing Systems* (NeurIPS, 2018), Vol. 31.
[24]A. Singraber, J. Behler, and C. Dellago, J. Chem. Theory Comput. **15**, 1827 (2019).
[25]A. Singraber, T. Morawietz, J. Behler, and C. Dellago, J. Chem. Theory Comput. **15**, 3075 (2019).
[26]B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis *et al.*, J. Chem. Phys. **153**, 194101 (2020).
[27]B. Cheng, G. Mazzola, C. J. Pickard, and M. Ceriotti, Nature **585**, 217 (2020).
[28]L. Zhang, H. Wang, R. Car, and W. E, Phys. Rev. Lett. **126**, 236001 (2021).
[29]D. Lu, H. Wang, M. Chen, L. Lin, R. Car, W. E, W. Jia, and L. Zhang, Comput. Phys. Commun. **259**, 107624 (2021).
[30]D. Tisi, L. Zhang, R. Bertossa, H. Wang, R. Car, and S. Baroni, Phys. Rev. B **104**, 224202 (2021).
[31]J. Behler, Chem. Rev. **121**, 10037 (2021).
[32]T. Zubatiuk and O. Isayev, Acc. Chem. Res. **54**, 1575 (2021).
[33]L. D. Jacobson, J. M. Stevenson, F. Ramezanghorbani, D. Ghoreishi, K. Leswing, E. D. Harder, and R. Abel, J. Chem. Theory Comput. **18**, 2354 (2022).
[34]T. E. Gartner III, P. M. Piaggi, R. Car, A. Z. Panagiotopoulos, and P. G. Debenedetti, Phys. Rev. Lett. **127**, 255702 (2022).
[35]C. Malosso, L. Zhang, R. Car, S. Baroni, and D. Tisi, npj Comput. Mater. **8**, 139 (2022).
[36]T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, Nat. Commun. **12**, 398 (2021).
[37]A. P. Bartók, R. Kondor, and G. Csányi, Phys. Rev. B **87**, 184115 (2013).
[38]J. Nigam, S. Pozdnyakov, and M. Ceriotti, J. Chem. Phys. **153**, 121101 (2020).
[39]M. P. Bircher, A. Singraber, and C. Dellago, Mach. Learn.: Sci. Technol. **2**, 035026 (2021).
[40]G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, J. Chem. Phys. **148**, 241730 (2018).
[41]E. Boattini, N. Bezem, S. N. Punnathanam, F. Smallenburg, and L. Filion, J. Chem. Phys. **153**, 064902 (2020).
[42]V. Molinero and E. B. Moore, J. Phys. Chem. B **113**, 4008 (2009).
[43]J. Russo, K. Akahane, and H. Tanaka, Proc. Natl. Acad. Sci. U. S. A. **115**, E3333 (2018).
[44]V. Holten, D. T. Limmer, V. Molinero, and M. A. Anisimov, J. Chem. Phys. **138**, 174501 (2013).
[45]E. B. Moore and V. Molinero, Nature **479**, 506 (2011).
[46]M. B. Davies, M. Fitzner, and A. Michaelides, Proc. Natl. Acad. Sci. U. S. A. **119**, e2205347119 (2022).
[47]J. A. Barker and R. O. Watts, Chem. Phys. Lett. **3**, 144 (1969).
[48]A. Rahman and F. H. Stillinger, J. Chem. Phys. **55**, 3336 (1971).
[49]H. Tanaka, J. Non-Cryst. Solids: X **13**, 100076 (2022).
[50]P. H. Poole, F. Sciortino, U. Essmann, and H. E. Stanley, Nature **360**, 324 (1992).
[51]G. A. Cisneros, K. T. Wikfeldt, L. Ojamäe, J. Lu, Y. Xu, H. Torabifard, A. P. Bartók, G. Csányi, V. Molinero, and F. Paesani, Chem. Rev. **116**, 7501 (2016).
[52]P. G. Debenedetti, F. Sciortino, and G. H. Zerze, Science **369**, 289 (2020).
[53]K. H. Kim, K. Amann-Winkel, N. Giovambattista, A. Späh, F. Perakis, H. Pathak, M. L. Parada, C. Yang, D. Mariedahl, T. Eklund *et al.*, Science **370**, 978 (2020).
[54]J. Weis, F. Sciortino, A. Z. Panagiotopoulos, and P. G. Debenedetti, J. Chem. Phys. **157**, 024502 (2022).
[55]T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, and F. Paesani, J. Chem. Phys. **148**, 241725 (2018).
[56]B. Cheng, E. A. Engel, J. Behler, C. Dellago, and M. Ceriotti, Proc. Natl. Acad. Sci. U. S. A. **116**, 1110 (2019).
[57]T. E. Gartner III, L. Zhang, P. M. Piaggi, R. Car, A. Z. Panagiotopoulos, and P. G. Debenedetti, Proc. Natl. Acad. Sci. U. S. A. **117**, 26040 (2020).
[58]O. Wohlfahrt, C. Dellago, and M. Sega, J. Chem. Phys. **153**, 144710 (2020).
[59]A. Torres, L. S. Pedroza, M. Fernandez-Serra, and A. R. Rocha, J. Phys. Chem. B **125**, 10772 (2021).
[60]A. Reinhardt and B. Cheng, Nat. Commun. **12**, 588 (2021).
[61]E. Lambros, S. Dasgupta, E. Palos, S. Swee, J. Hu, and F. Paesani, J. Chem. Theory Comput. **17**, 5635 (2021).
[62]P. M. Piaggi, J. Weis, A. Z. Panagiotopoulos, P. G. Debenedetti, and R. Car, arXiv:2203.01376 (2022).
[63]Y. Zhai, A. Caruso, S. L. Bore, Z. Luo, and F. Paesani, J. Chem. Phys. **158**, 084111 (2023).
[64]V. Babin, C. Leforestier, and F. Paesani, J. Chem. Theory Comput. **9**, 5395 (2013).
[65]V. Babin, G. R. Medders, and F. Paesani, J. Chem. Theory Comput. **10**, 1599 (2014).
[66]G. R. Medders, V. Babin, and F. Paesani, J. Chem. Theory Comput. **10**, 2906 (2014).
[67]S. Yue, M. C. Muniz, M. F. Calegari Andrade, L. Zhang, R. Car, and A. Z. Panagiotopoulos, J. Chem. Phys. **154**, 034111 (2021).
[68]A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Advances in Neural Information Processing Systems* (NeurIPS, 2017), Vol. 30.
[69]S. N. Pozdnyakov, M. J. Willatt, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, Phys. Rev. Lett. **125**, 166001 (2020).
[70]P. J. Huber, Ann. Math. Stat. **35**, 73 (1964).
[71]X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (JMLR Workshop and Conference Proceedings, 2010), pp. 249–256.
[72]I. Loshchilov and F. Hutter, in *Proceedings of the International Conference on Learning Representations* (ICLR, 2017).
[73]F. H. Stillinger and T. A. Weber, Phys. Rev. B **31**, 5262 (1985).
[74]F. Romano, J. Russo, and H. Tanaka, Phys. Rev. B **90**, 014204 (2014).
[75]M. J. Gillan, D. Alfè, and A. Michaelides, J. Chem. Phys. **144**, 130901 (2016).
[76]TensorFlow, Tensorflow c 2.7, https://www.tensorflow.org/install/lang_c.
[77]J. Russo, F. Leoni, F. Martelli, and F. Sciortino, Rep. Prog. Phys. **85**, 016601 (2021).
[78]R. Foffi and F. Sciortino, Phys. Rev. Lett. **127**, 175502 (2021).
[79]R. Foffi, J. Russo, and F. Sciortino, J. Chem. Phys. **154**, 184506 (2021).
[80]H. C. Andersen, J. Chem. Phys. **72**, 2384 (1980).
[81]B. Monserrat, J. G. Brandenburg, E. A. Engel, and B. Cheng, Nat. Commun. **11**, 5757 (2020).