The Journal of
Chemical Physics

Unsupervised learning for local structure detection in colloidal systems

Cite as: J. Chem. Phys. **151**, 154901 (2019); https://doi.org/10.1063/1.5118867
Submitted: 08 July 2019 . Accepted: 25 September 2019 . Published Online: 15 October 2019

Emanuele Boattini, Marjolein Dijkstra, and Laura Filion

View Online Export Citation CrossMark

AIP Publishing

Unsupervised learning for local structure detection in colloidal systems

Cite as: J. Chem. Phys. 151, 154901 (2019); doi: 10.1063/1.5118867
Submitted: 8 July 2019 • Accepted: 25 September 2019 •
Published Online: 15 October 2019

Emanuele Boattini,[a)] Marjolein Dijkstra, and Laura Filion

AFFILIATIONS

Soft Condensed Matter, Debye Institute for Nanomaterials Science, Utrecht University, Utrecht, The Netherlands

[a)]Electronic mail: e.boattini@uu.nl

ABSTRACT

We introduce a simple, fast, and easy to implement unsupervised learning algorithm for detecting different local environments on a single-particle level in colloidal systems. In this algorithm, we use a vector of standard bond-orientational order parameters to describe the local environment of each particle. We then use a neural-network-based autoencoder combined with Gaussian mixture models in order to autonomously group together similar environments. We test the performance of the method on snapshots of a wide variety of colloidal systems obtained via computer simulations, ranging from simple isotropically interacting systems to binary mixtures, and even anisotropic hard cubes. Additionally, we look at a variety of common self-assembled situations such as fluid-crystal and crystal-crystal coexistences, grain boundaries, and nucleation. In all cases, we are able to identify the relevant local environments to a similar precision as "standard," manually tuned, and system-specific, order parameters. In addition to classifying such environments, we also use the trained autoencoder in order to determine the most relevant bond orientational order parameters in the systems analyzed.

I. INTRODUCTION

An important challenge in the study of colloidal self-assembly is the detection of self-assembled products in the system. When the basic phases that the system forms are well characterized, we can design order parameters that detect, on a single particle level, which of the expected phases a particle is in. This strategy has been extensively used in studies ranging from crystal nucleation and growth[1–4] to crystal melting, as well as colloidal glasses,[5] and grain boundary dynamics.[6,7] Over the years, a number of different routes have been taken to characterize such local order, including order parameters based on bond-orientational order,[8–11] common neighbor analysis (CNA),[12,13] and templating.[14] Moreover, developments in the last few years have started to combine these local descriptions with supervised machine learning techniques in order to recognize specific crystal structures.[15–18] A very recent development has shown that this strategy can even be applied, with the aid of deep learning techniques, to simple coordinate information (instead of the standard local order descriptors).[19] In general, supervised learning strategies work very well to distinguish structures that we expect to form.

However, in many cases when exploring the self-assembly of new systems, the exact final structure and its characteristics are unknown, complicating the selection of an ideal order parameter. In such situations, the standard solution has been to apply different potential order parameters to the system, for instance, the bond order parameter q_6, and try and see if different regions appeared. This process is time consuming and difficult, and has the significant drawback that one might miss important structures that are not connected with the order parameters explored. In this context, unsupervised machine learning techniques, which excel in autonomously finding patterns in large data sets, offer a promising solution. In a recent paper,[20] Reinhart *et al.* made an attempt to use unsupervised machine learning in order to identify different crystal structures. In their work, they described the local environment of the particles with the adjacency criterion from adaptive CNA and combined it with the diffusion map technique for dimensionality reduction in order to distinguish different, frequently occurring, structures. Although very successful, this method turned out to be very computationally demanding. In a subsequent article,[21] a faster way of comparing local neighborhoods was introduced, based on their relative graphlet frequencies. This reduced the computational cost of the algorithm by four orders of magnitude.

A different approach was followed by Spellings and Glotzer in Ref. 22, where they used a combination of unsupervised and supervised learning techniques in order to identify the overall crystal structures of bulk self-assembled systems (i.e., systems of which the majority had self-assembled into the same phase).

In this work, we present a new avenue to detect self-assembly products and introduce an unsupervised machine learning algorithm based on bond-orientational order parameters combined with neural-network-based autoencoders[23–27] and Gaussian mixture models (GMMs).[28–30] Autoencoders are a standard technique for nonlinear dimensionality reduction, while mixture models are probabilistic models for identifying distinct clusters within a data set. Using these methods, our algorithm can autonomously classify particles in different groups based on their local order, making it easy to detect any self-assembly product in the system.

In contrast to Ref. 22, the goal here is to identify local environments on a single-particle level—meaning that this method can be used to study processes such as nucleation, grain boundary characteristics, and coexistences. This algorithm has been designed to be computationally fast, easily scalable to very large data sets, and extremely easy to implement. To test its performance, we examine a number of different colloidal systems, ranging from spheres to binary mixtures to anisotropic particles. Moreover, in addition to simply classifying local environments, we also explore whether the unsupervised learning techniques employed can help us identify the distinguishing features of the different particle environments found in the system.

II. METHODS

In this section, we describe in detail the algorithm we use to classify local environments. We start by summarizing the main steps of our approach and then follow with a detailed description of each step in separate subsections.

The overall process consists of three steps. First, we require a method to capture the local environment of each particle in a set of local order parameters. For this, we make use of bond-orientational order parameters. This set of order parameters is in general high-dimensional and may contain significant amounts of redundant and irrelevant information. In order to extract the most relevant information, the second step of our approach makes use of a dimensionality reduction technique, namely, a neural-network-based autoencoder. Once trained, the autoencoder projects the original (high-dimensional) input vectors onto a lower-dimensional subspace encoding the features with the largest variations in the input data. Ideally, in this subspace, particles with similar local environments are grouped together. Finally, we apply a clustering algorithm (Gaussian mixture models) in order to identify the distinct clusters of local environments in this lower-dimensional subspace.

A. Bond order parameters

To characterize the local environment of each particle, we use the averaged bond order parameters (BOPs) introduced by Lechner and Dellago.[8,11] First, we define for any given particle i the complex quantities

$$q_{lm}(i) = \frac{1}{N_b(i)} \sum_{j \in \mathcal{N}_b(i)} Y_l^m(\mathbf{r}_{ij}), \tag{1}$$

where $Y_l^m(\mathbf{r_{ij}})$ are the spherical harmonics of order l, with m an integer that runs from $m = -l$ to $m = +l$. Additionally, $\mathbf{r}_{ij}$ is the vector from particle i to particle j, and $\mathcal{N}_b(i)$ is the set of nearest neighbors of particle i, which we will define later. Note that $\mathcal{N}_b(i)$ contains $N_b(i)$ particles. Then, we can define an average $\bar{q}_{lm}(i)$ as

$$\bar{q}_{lm}(i) = \frac{1}{N_b(i)+1} \sum_{k \in \{i, \mathcal{N}_b(i)\}} q_{lm}(k), \tag{2}$$

where the sum runs over all nearest neighbors of particle i as well as particle i itself. Averaging over the nearest neighbor values of q_{lm} results effectively in also taking next-nearest neighbors into account. Finally, we define rotationally invariant BOPs as

$$\bar{q}_l(i) = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\bar{q}_{lm}(i)|^2}, \tag{3}$$

which, depending on the choice of l, are sensitive to different crystal symmetries.

The optimal set of BOPs to be considered strongly depends on the structures one wishes to distinguish. Since our method is meant to be applied to systems for which such prior knowledge is missing, in order to describe the local environment of one particle, we evaluate several $\bar{q}_l$ with l ranging from 1 to 8. Note that in principle, one could consider a larger (or smaller) range of l. For all cases examined in this paper, however, we found 8 to be sufficient. Therefore, when considering one component systems, our description of the local environment of particle i is encoded into an 8-dimensional vector,

$$\mathbf{Q}(i) = (\{\bar{q}_l(i)\}), \tag{4}$$

with $l \in [1, 8]$. When considering binary mixtures, i.e., systems with two species of particles, the same BOPs are evaluated both considering all the nearest neighbors of the reference particle (regardless of particles' species) and considering only the nearest neighbors of the same species as the reference particle. Hence, for binary mixtures, our description of the local environment of particle i is encoded into a 16-dimensional vector,

$$\mathbf{Q}(i) = (\{\bar{q}_l(i)\}, \{\bar{q}_l^{ss}(i)\}), \tag{5}$$

where s indicates the particles' species. Here, $\{\bar{q}_l\}$ represents the set of BOPs evaluated considering all the nearest neighbors of particle i, while the set $\{\bar{q}_l^{ss}\}$ is evaluated considering only the nearest neighbors of the same species s as particle i.

Thus far, we have not discussed the definition of a nearest neighbor, as used in the definition of the BOPs. There are a number of different avenues for identifying nearest neighbors. The simplest method relies on using a fixed cutoff radius, r_c, such that all particles closer than this distance are considered nearest neighbors. Ideally, this cutoff radius is chosen as the distance at which the radial distribution function has its first minimum. This method has the advantage that it is computationally very cheap and it is symmetric, i.e., i is a neighbor of j if and only if j is a neighbor of i. However, r_c is system and density dependent so that it has to be tuned for every particular case requiring prior knowledge of the system under study. Additionally, the cutoff is defined for the entire system and, as such, is not an optimal choice for systems with large density gradients or interfaces, such as can occur in nucleation studies.

Another standard method for determining nearest neighbors is the Voronoi construction,[31] which has the advantage that it is parameter free. However, it is also relatively computationally expensive, and in this work, we have instead opted to make use of a recently introduced alternative parameter-free nearest-neighbor criterion, called SANN (solid angle nearest neighbor).[32] In this approach, an effective individual cutoff is found for every particle in the system based on its local environment. This method is not inherently symmetric, i.e., j might be a neighbor of i, while i is not a neighbor of j. However, symmetry can be enforced by either adding j to the neighbors of i or removing i from the neighbors of j. In this study, we applied the latter solution. The computational cost of SANN only slightly exceeds that of a cutoff distance, and, since it is a parameter-free method, it is suitable for systems with inhomogeneous densities.

B. Unsupervised learning

1. Nonlinear dimensionality reduction using neural-network-based autoencoders

In order to extract the relevant information contained in the vectors $\mathbf{Q}(i)$, we use neural-network-based autoencoders.[23–27] An autoencoder is a neural network that is trained to perform the identity mapping, where the network inputs are reproduced at the output layer. The network may be viewed as consisting of two parts: an encoder network, which performs a nonlinear projection of the input data onto a low-dimensional subspace, and a decoder network that attempts to reconstruct the input data from the low-dimensional projection. This architecture is represented in Fig. 1.

By training the autoencoder to perform the input reconstruction task over an ensemble of training examples, the encoder is forced to learn a low-dimensional nonlinear projection that preserves the most relevant features of the data and from which the higher-dimensional inputs can be approximately reconstructed by the decoder. In this work, the training data are the vectors $\mathbf{Q}(i)$ [Eq. (4) or (5)] evaluated from snapshots of colloidal systems obtained via computer simulations, and the autoencoder is trained to find a low-dimensional projection of such vectors by eliminating irrelevant and redundant information.

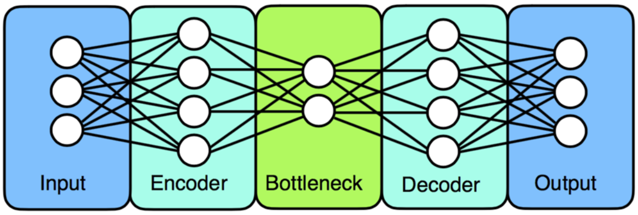


FIG. 1. Architecture of a neural-network based autoencoder. The encoder network finds a low-dimensional representation of the input, from which the decoder reconstructs an approximation of the input as output.

In the present context, we employ feedforward and fully connected autoencoders like the one presented in Fig. 1. The number of input and output nodes, d, is specified by the dimensionality of the input vectors, $\mathbf{Q}(i) \in \mathbb{R}^d$, which are approximately reconstructed by the network in the output layer, $\hat{\mathbf{Q}}(i) \in \mathbb{R}^d$. The bottleneck layer contains the low-dimensional projection to be learned by the encoder, $\mathbf{Y}(i) \in \mathbb{R}^c$, whose dimensionality is controlled by the number of bottleneck nodes, $c < d$. Nonlinearity is achieved by providing both the encoder and the decoder with a fully connected hidden layer with a nonlinear activation function. Here, we set the number of nodes in the hidden layers to $10d$ and use a hyperbolic tangent as the activation function. For the bottleneck and output layers, instead, a linear activation function is used. Note that these choices are fairly standard and were determined empirically for what worked well in our case. More general information on choosing activation functions can be found in, e.g., Refs. 27 and 33.

The internal parameters of the autoencoder, i.e., weights $\mathbf{W} \equiv \{w_j\}$ and biases $\mathbf{B} \equiv \{b_k\}$, are initialized with the normalized initialization proposed by Xavier in Ref. 33, i.e., the biases are initialized to zero, while the weights are drawn from a normal distribution with zero mean and a variance that depends on the size of the layers. During the training, these parameters are optimized by minimizing the reconstruction error of the input data over a training set of N training examples. Specifically, we consider the mean squared error (MSE) with the addition of a weight decay regularization term[27] to control the magnitude of the network weights,

$$E(\mathbf{W}, \mathbf{B}; \{\mathbf{Q}(i)\}) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{Q}(i) - \hat{\mathbf{Q}}(i)\right\|^2 + \lambda\sum_{j=1}^{M} w_j^2, \tag{6}$$

where M is the total number of weights, whose value depends on the dimension of the network, and we set $\lambda = 10^{-5}$. The function in Eq. (6) is minimized using minibatch stochastic gradient descent with momentum.[27,34,35]

The optimal number of nodes in the bottleneck layer, c, which defines the unknown relevant dimensionality of the input data, can be determined by computing the reconstruction MSE and looking for the existence of an elbow in the MSE as a function of c.[36] For convenience, we rescale the MSE by the mean squared deviation (MSD) of the vectors $\mathbf{Q}(i)$,

$$\text{MSD} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{Q}(i) - \bar{\mathbf{Q}}\right\|^2, \tag{7}$$

where $\bar{\mathbf{Q}}$ is the mean input vector. To detect the presence of an elbow, we use the L-method proposed by Salvador and Chan.[37] For all systems examined in this work, we found a dimensionality of $c = 2$ to be sufficient.

Once the autoencoder is trained, the encoder network alone is retained in order to perform the nonlinear mapping of the input vectors $\mathbf{Q}(i)$ onto the low-dimensional subspace defined by the bottleneck layer, $\mathbf{Y}(i)$.

2. Learning from the autoencoder

One of the main advantages of using a neural-network-based autoencoder over other nonlinear techniques for dimensionality reduction is that it furnishes an exact analytical mapping (and an approximate inverse mapping) between the original input space and its low-dimensional projection. In finding such a mapping, the autoencoder must understand which of all the BOPs given as input in the vectors $\mathbf{Q}(i)$ are the most relevant for the system under analysis. Extrapolating this knowledge would help us understand the relevant symmetries distinguishing the different environments possibly present in the system.

Several methods to assess the relative importance of input variables in neural network models have been proposed.[38–41] Here, we consider the input perturbation[38–41] and the improved stepwise[40,41] methods. Both techniques require the use of a single trained model, avoiding having to repeat the training of the autoencoder multiple times.

The input perturbation method assesses the variation in the MSE of the autoencoder by adding, in turn, a small amount of white noise to the kth input while holding all the other inputs at their observed values. Here, we set the white noise to 10% and 50% of each input, as suggested in Ref. 40. The input variables whose changes affect the output the most, leading to a large increase in the MSE, are the ones that have the most relative influence.

The improved stepwise method is very similar in spirit, but instead of adding noise to one of the inputs, it replaces all its values with its mean over the whole dataset. Also in this case, the most relevant input variables are identified as the ones whose replacement causes the largest increase in the MSE.

In both cases, a quantitative measure of the relative importance, RI_k, of the kth input can be obtained as

$$\mathrm{RI}_k = \frac{\Delta E_k}{\sum_{j=1}^{d} \Delta E_j}, \tag{8}$$

where ΔE_k is the variation in the MSE caused by the change applied to the kth input and the sum in the denominator runs over all the input variables.

3. Clustering

In order to cluster together similar environments in the low-dimensional subspace found by the encoder, we use Gaussian mixture models (GMMs) as implemented in scikit-learn.[42]

GMM is a probabilistic model that assumes that the observed data are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Such parameters are optimized iteratively with the expectation-maximization (EM) algorithm[28] in order to create a probability density function that agrees well with the distribution of the data. The number of Gaussian components in the mixture, N_G, is usually found by minimizing the Bayesian information criterion (BIC),[29] which measures how well a GMM fits the observed data while penalizing models with many parameters to prevent overfitting. The output of a trained GMM is a list of probabilities, p_{ij}, corresponding to the posterior probabilities of the ith observation to arise from the jth component in the mixture model.

The simplest form of clustering that can be applied consists in considering each mixture component as generating a separate cluster and assigning each observation to the component with the highest posterior probability. However, while this procedure works perfectly for clusters that are really generated from a mixture of separate multivariate normal distributions, the clusters that underline our data are very often far from being Gaussian-distributed in space. As a consequence, a single cluster in the data may be detected as two or more mixture components (if its distribution is indeed better approximated by a mixture of Gaussians than by a single Gaussian function), meaning that the number of clusters in the data may in general be different from the number of components found by minimizing the BIC.

To overcome this problem, we use the method proposed by Baudry *et al.*[30] The idea is to first use the BIC in order to find a GMM with N_G components that fits the data well. Then, a sequence of candidate clusterings with $K = N_G, N_G - 1, \ldots, 1$ clusters is formed by successively merging a pair of components. At each step, the two mixture components to be merged are chosen so as to minimize the entropy of the resulting clustering, defined as

$$S_K = -\sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij}\ln(p_{ij}), \tag{9}$$

where N is the number of observations and K is the number of clusters. Finally, the optimal number of clusters is found by looking for the existence of an elbow in the entropy S_K as a function of K. Again, we detect the elbow with the L-method of Salvador and Chan.[37]

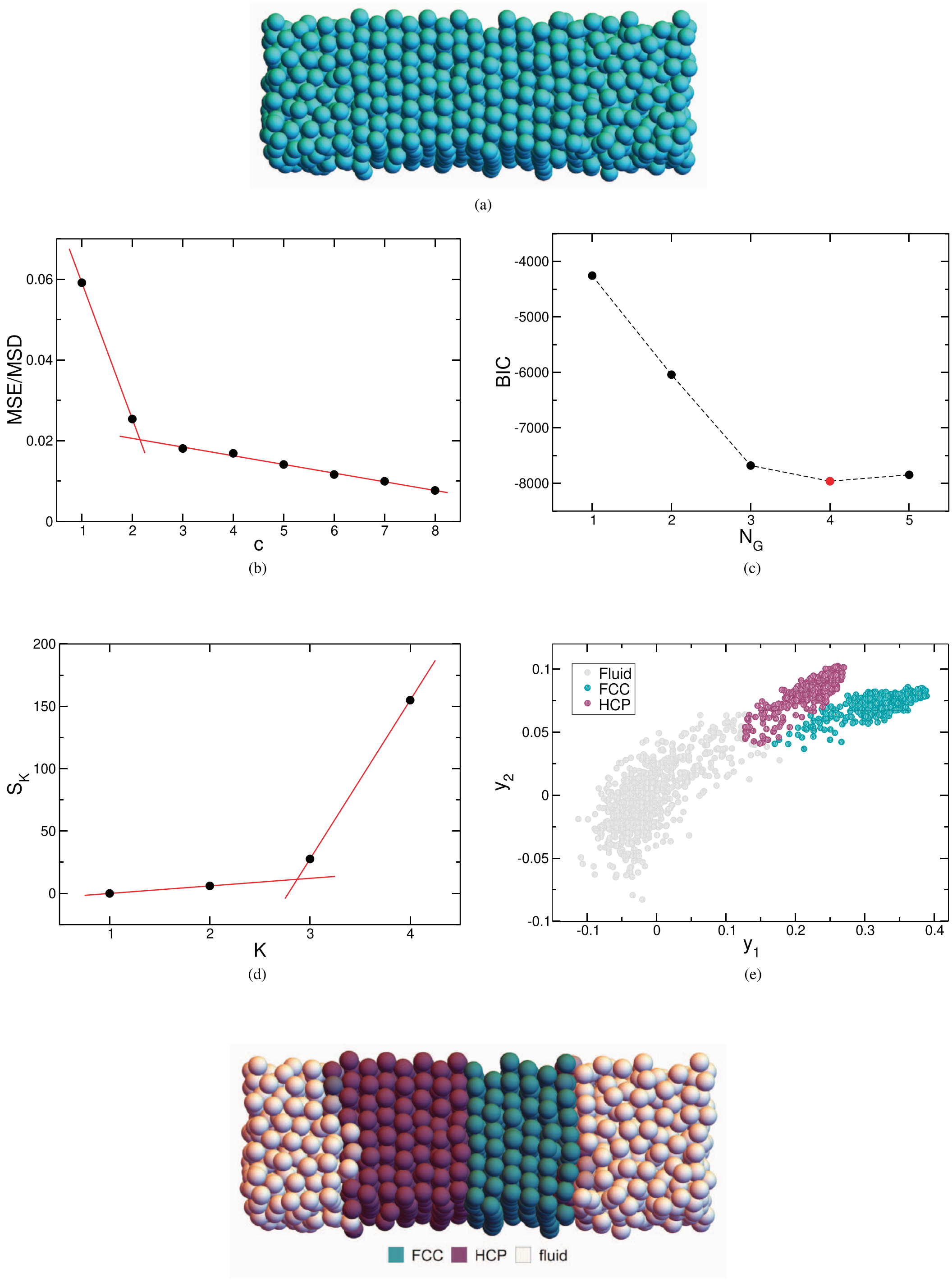
This procedure autonomously finds the number of clusters underlying the data, corresponding to the distinct particle environments present in the system under analysis. Moreover, note that this is a soft clustering technique, meaning that each particle is not simply assigned to a cluster corresponding to a specific local environment, but rather it has a certain probability of belonging to any of the identified clusters. As a result, particles whose environment is not well defined, such as can occur at interfaces, will have similar probabilities of belonging to different clusters. In the following, we will refer to these probabilities as membership probabilities.

III. RESULTS AND DISCUSSION

In this section, we show how our method performs on snapshots of a wide range of colloidal systems obtained via computer simulations. We first present in detail the whole procedure for the analysis of a "test" example for which we compare the results with a more standard, system-specific, methodology. A shorter, more concise, discussion is dedicated to the results obtained for the other systems analyzed, including systems with grain boundaries, anisotropic hard cubes, and binary mixtures.

A. Single-component hard spheres

As a first test case, we examine a snapshot from a Monte Carlo (MC) simulation of single-component hard spheres of diameter σ, which is shown in Fig. 2(a). The simulation was performed in the canonical ensemble (constant number of particles N, volume V, and temperature T), which was initialized as a metastable coexistence



(a)

(b)

(c)

(d)

(e)

(f)

FIG. 2. Analysis of a snapshot from a MC simulation of hard spheres showing a coexistence between fluid, FCC, and HCP crystals. (a) Snapshot under analysis. (b) Rescaled MSE of the autoencoder as a function of the number of bottleneck nodes. Solid lines show the presence of an elbow at $c = 2$. (c) BIC as a function of the number of components in the GMM. The minimum is highlighted in red. Dashed lines are only a guide for the eyes. (d) Entropy of the clustering as a function of the number of clusters. Solid lines show the presence of an elbow at $K = 3$. (e) Projection of the vectors $\mathbf{Q}(i)$ onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering. (f) Classification of the snapshot in panel (a). The RGB color of each particle is a linear combination of the colors of the three phases (see legend) with the associated membership probabilities as coefficients.

between the fluid, hexagonal close-packed (HCP), and face-centered cubic (FCC) phases. The system contained $N = 1536$ particles and was at a number density $\rho\sigma^3 = 1.01$.

1. Analysis

Starting from the raw coordinates of each particle i in the system, we build the vectors $\mathbf{Q}(i)$ in Eq. 4 and use them as an input for the autoencoder. To find the optimal dimensionality of the bottleneck layer, c, we evaluate the reconstruction MSE of the autoencoder for $c \in [1, 8]$. A plot of the rescaled MSE as a function of c is shown in Fig. 2(b). Solid lines, obtained with the L-method, clearly show the presence of an elbow at $c = 2$, indicating that a two-dimensional projection of the original input vectors is sufficient to preserve the relevant information. The projection learned by the encoder is depicted in Fig. 2(e), where each point corresponds to a particle in the system. Note that the way we color the points in Fig. 2(e) will be explained later.

We then apply GMM in this two-dimensional space in order to identify the relevant clusters, i.e., the distinct particle environments. Following the method of Baudry *et al.*, we first optimize the number of Gaussian components in the mixture model, N_G, by minimizing the BIC. The BIC as a function of N_G is shown in Fig. 2(c) and has a minimum for $N_G = 4$. Then, the optimal number of clusters, K, is found by successively merging a pair of components and looking for the existence of an elbow in the clustering entropy as a function of K [see Fig. 2(d)]. The elbow is detected at $K = 3$, meaning that the unsupervised learning identifies three relevant environments. Note that we know beforehand the three phases present in the system (FCC, HCP, and fluid) so that we can easily associate each cluster with the correct phase. An idea of the partitioning of space performed by the clustering is given in Fig. 2(e), where the color of each point is determined by the cluster with the highest membership probability. As discussed in Sec. II B 3, however, some points might have a nonvanishing probability of belonging to more than one cluster. In order to fully account for such additional information, in the snapshot in Fig. 2(f), the RGB color of the particles is obtained as a linear combination of the colors of the three clusters, with the associated membership probabilities as coefficients. As a result, particles with an environment falling at the boundary between two clusters, hence having two nonvanishing membership probabilities of similar magnitude, appear with a different color from those in the legend. This behavior is clearly observed for some of the particles at the crystal-fluid interfaces.

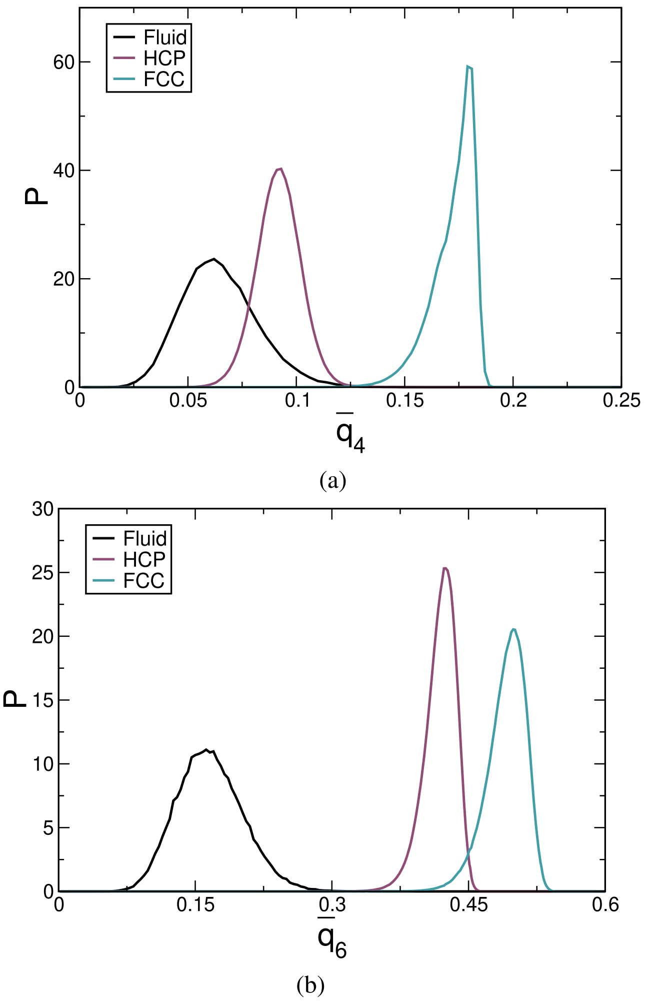
In the following, we compare the results obtained with a more standard methodology specifically tuned for distinguishing the phases in this system.

2. Comparison

A common method to classify different phases on a single-particle basis consists in finding one (or more) pair(s) of BOPs, $(\bar{q}_l, \bar{q}_{l'})$, whose distributions in the phases of interest are considerably different, so that it is possible to define separate regions in the $\bar{q}_l$-$\bar{q}_{l'}$-plane corresponding to different particle environments. The pair of BOPs to consider strongly depends on the environments one wishes to distinguish, and it is usually found by trial and error.

For the FCC, HCP, and fluid phases, $\bar{q}_4$ and $\bar{q}_6$ are the most common choice.[11] Figure 3 shows the probability distribution of $\bar{q}_4$ [3(a)] and $\bar{q}_6$ [3(b)] for the three hard-sphere phases, obtained from separate MC simulations in the canonical ensemble. Simulations of the two crystal phases were performed at a number density just above the coexistence region, $\rho\sigma^3 = 1.05$, while for the fluid phase we used the coexistence density, $\rho\sigma^3 = 0.94$.

The distributions of $\bar{q}_4$ in the two crystals are well separated, while the $\bar{q}_4$ distribution of the fluid phase strongly overlaps the one of the HCP crystal. On the other hand, the $\bar{q}_6$ distributions show a large separation between the fluid and the crystal phases but a small overlap between the two crystals. Alone, none of the BOPs considered completely separates the three phases. However, a separation can be found in the $\bar{q}_4$-$\bar{q}_6$-plane. A comparison of the phases in the $\bar{q}_4$-$\bar{q}_6$-plane is shown in Fig. 4. In this plane, one can easily



(a)

(b)

FIG. 3. (a) Probability distribution of $\bar{q}_4$ for the fluid ($\rho\sigma^3 = 0.94$), FCC ($\rho\sigma^3 = 1.05$), and HCP ($\rho\sigma^3 = 1.05$) hard-sphere phases. (b) Probability distribution of $\bar{q}_6$ for the same phases.

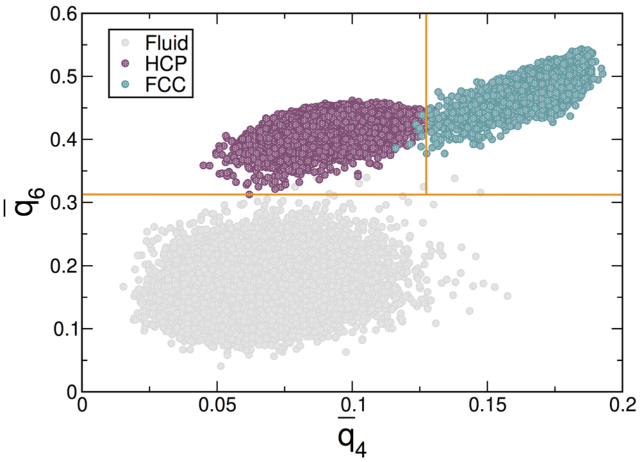


FIG. 4. Comparison between the $\bar{q}_4$-$\bar{q}_6$-plane for the fluid ($\rho\sigma^3$ = 0.94), FCC ($\rho\sigma^3$ = 1.05), and HCP ($\rho\sigma^3$ = 1.05) hard-sphere phases. Each point correspond to a particular particle. 200 00 points from each phase were chosen randomly. Orange solid lines show a possible linear separation of the three phases in this plane.

identify three linearly separated regions associated with the three phases. A possible choice of such a separation is represented by the solid lines in Fig. 4. Based on this definition, the particles in Fig. 2(a) can be classified according to the region in which their BOPs fall. The results of this classification are presented in Fig. 5, and they are in excellent agreement with the ones obtained via unsupervised learning [see Fig. 2(f) for comparison]. As can be seen by comparing Figs. 2(f) and 5, the only differences between the two classifications are at the interfaces and are generally particles for which the unsupervised learning algorithm gave at least two comparable membership probabilities (e.g., identified large probabilities of being in both fluid and FCC).

Note that the method presented above requires, in general, prior knowledge of the phases to be expected in the system under analysis. Moreover, additional simulations of such phases must be performed in order to (i) identify the relevant BOPs and (ii) define separations between the distinct particle environments. Both tasks, (i) and (ii), are autonomously performed by our unsupervised-learning method based *only* on the vectors of BOPs evaluated from the snapshot under study.

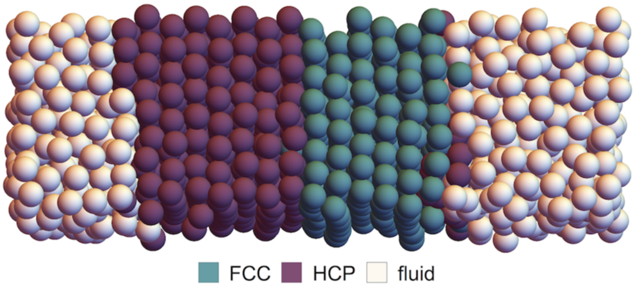


FIG. 5. Classification of the snapshot in Fig. 2 based on $\bar{q}_4$ and $\bar{q}_6$.

Regarding the identification of the relevant BOPs, i.e., task (i), in Sec. III A 3, we present how such information can be extracted from the trained autoencoder.

3. Learning from the autoencoder

As discussed in Sec. II B 2, several methods to assess the relative importance of input variables in neural network models are available. Here, we employ the input perturbation and the improved stepwise methods in order to understand which BOPs were considered to be the most relevant by the autoencoder for the system in Fig. 2. The relative importance of the BOPs evaluated with these methods is shown in Fig. 6.

Only a small subset of three BOPs is found to be relevant and, as expected, $\bar{q}_6$ (RI ~ 78%) and $\bar{q}_4$ (RI ~ 3%) are part of it. Interestingly however, $\bar{q}_8$, which to our knowledge has never been used in the literature, appears to be more important (RI ~ 19%) than $\bar{q}_4$. To understand why this is, we evaluated the $\bar{q}_8$ distributions in the FCC, HCP, and fluid hard-sphere phases from several snapshots (see Fig. 7). Such distributions are very similar to those obtained for $\bar{q}_6$, in the sense that they show a clear separation between the fluid and the crystal phases and only a small overlap between the two crystals. If we now go back to the snapshot shown in Fig. 2(f), which is roughly half fluid and half crystal, then it is easy to understand why $\bar{q}_4$ has a lower relative importance than $\bar{q}_6$ and $\bar{q}_8$.

Note that the RI in Fig. 6 only indicates which are the BOPs with the largest variations in the system, without giving information on their possible correlations. As a consequence, the number of relevant BOPs is in general larger than the dimensionality of the bottleneck layer because the encoder is able to find and remove such (possibly nonlinear) correlations.

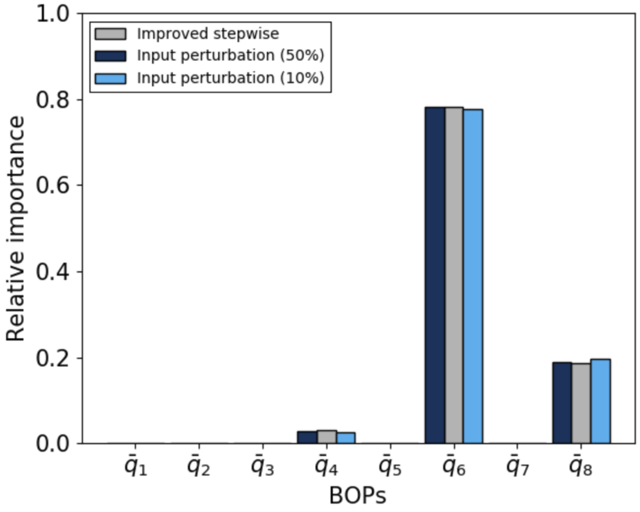


FIG. 6. Relative importance of the BOPs for the system in Fig. 2 assessed with the input perturbation and the improved stepwise methods. The input perturbation method is applied with two different amounts of white noise: 10% and 50% of the input, respectively.

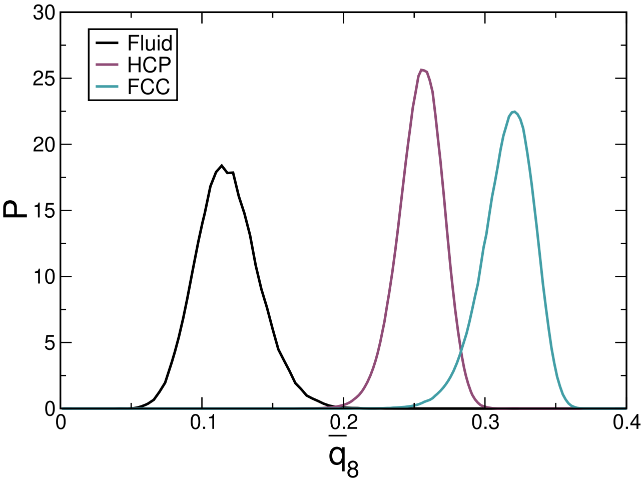


FIG. 7. Probability distribution of $\bar{q}_8$ for the fluid ($\rho\sigma^3$ = 0.94), FCC ($\rho\sigma^3$ = 1.05), and HCP ($\rho\sigma^3$ = 1.05) hard-sphere phases.

B. Grain boundaries

We now consider a snapshot of a system with FCC crystalline domains separated by grain boundaries, depicted in Fig. 8(a). The system contains N = 83 481 particles interacting via the purely repulsive Weeks-Chandler-Andersen (WCA) potential,

$$\beta U_{WCA}(r) = \begin{cases} 4\beta\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} + \frac{1}{4}\right], & \frac{r}{\sigma} \le 2^{1/6} \\ 0, & \frac{r}{\sigma} > 2^{1/6} \end{cases} \tag{10}$$

with σ as the particle diameter, $\beta\varepsilon$ = 40 as the energy scale, and $\beta = 1/k_B T$, where k_B is the Boltzmann constant and T is the temperature. More details about the simulation can be found in Ref. 6.

The results of the unsupervised learning algorithm are summarized in Fig. 8. Specifically, Fig. 8(b) shows the two-dimensional projection of the vectors $\mathbf{Q}(i)$ found by the encoder and the results of the clustering performed in this space. Our method identifies two distinct particle environments, corresponding to particles within the grain boundaries and within the FCC crystalline domains. Note that the particles in the grain boundaries appear disordered and fluidlike. This identification is used to color the particles in Fig. 8(a). The most relevant BOPs in this system according to the autoencoder analysis are $\bar{q}_6$ (RI ~ 78%) and $\bar{q}_8$ (RI ~ 19%), which are indeed those whose distributions show the largest separation between the fluid and crystal phases.

C. Hard cubes

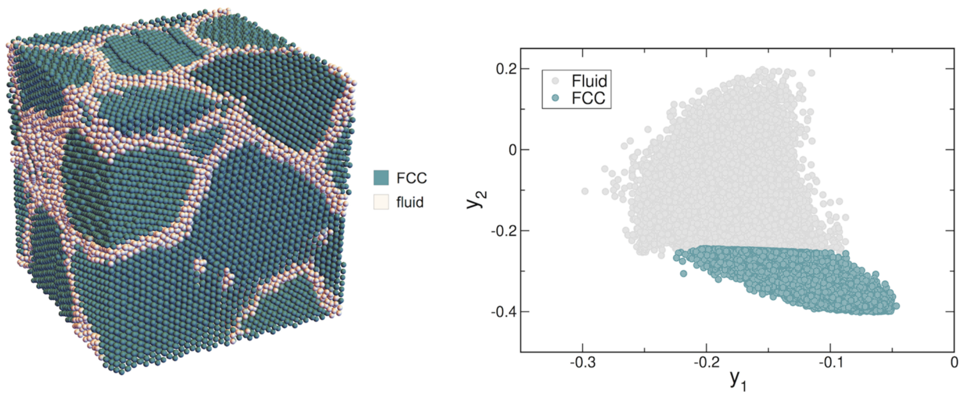
The systems examined so far were all characterized by isotropic interactions between their constituents. As a further test of the performance and generality of our method, we now consider a snapshot of a system of hard cubes with edge length σ [see Fig. 9(a)], obtained from an event-driven molecular dynamics (EDMD) simulation in the canonical ensemble.[43] The simulation was performed with N = 64 000 particles starting from a simple cubic (SC) crystal configuration at a number density in the fluid-crystal coexistence region, $\rho\sigma^3$ = 0.475. More details on the simulation can be found in Ref. 43.

The two-dimensional projection of the vectors $\mathbf{Q}(i)$ found by the encoder and the results of the clustering performed in this space are shown in Fig. 9(b), while the final classification is presented in Fig. 9(a). In agreement with the standard order parameters used in Ref. 43, the unsupervised learning method identifies two distinct particle environments, corresponding to the fluid and SC crystal phases. From the autoencoder analysis, we found that the most relevant BOPs for this system are $\bar{q}_6$ (RI ~ 48%), $\bar{q}_4$ (RI ~ 36%) and $\bar{q}_8$ (RI ~ 16%).

D. Binary mixture

After having considered single-component systems with both isotropic and anisotropic interactions, we now examine a binary mixture of large (L) and small (S) spheres of diameters σ_L and σ_S, respectively, with a size ratio σ_S/σ_L = 0.78 and a stoichiometry $N_s/N = \frac{2}{3}$. The particles in this system interact via the WCA potential, $U_{\alpha\gamma}(r)$, between species $\alpha = L, S$ and $\gamma = L, S$,

$$\beta U_{\alpha\gamma}(r) = \begin{cases} 4\beta\varepsilon\left[\left(\frac{\sigma_{\alpha\gamma}}{r}\right)^{12} - \left(\frac{\sigma_{\alpha\gamma}}{r}\right)^{6} + \frac{1}{4}\right], & \frac{r}{\sigma_{\alpha\gamma}} \le 2^{1/6} \\ 0, & \frac{r}{\sigma_{\alpha\gamma}} > 2^{1/6}, \end{cases} \tag{11}$$



(a)

(b)

FIG. 8. Analysis of a snapshot of a system with grain boundaries. (a) Classification of the snapshot under analysis. The RGB color of each particle is a linear combination of the colors of the two phases identified with the associated membership probabilities as coefficients. (b) Projection of the vectors $\mathbf{Q}(i)$ onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering.

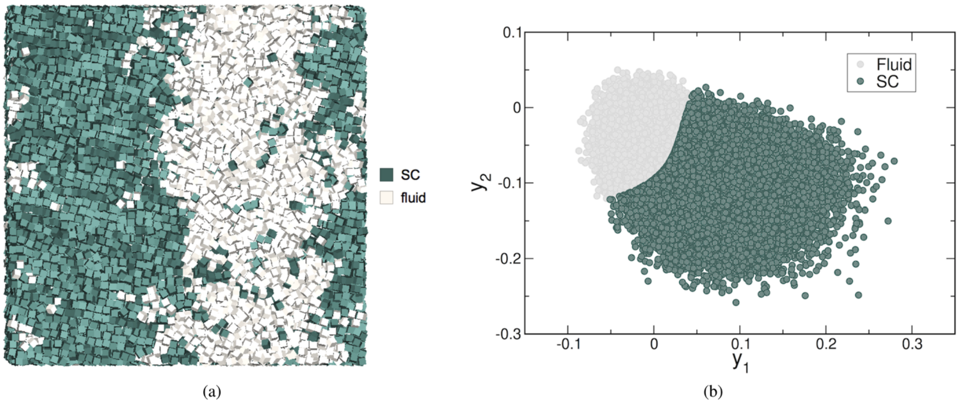


FIG. 9. Analysis of a snapshot of hard cubes showing a coexistence between the fluid and the SC crystal phases. (a) Classification of the snapshot under analysis. The RGB color of each particle is a linear combination of the colors of the two phases identified with the associated membership probabilities as coefficients. (b) Projection of the vectors **Q**(i) onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering.

where $\sigma_{\alpha\gamma} = (\sigma_\alpha + \sigma_\gamma)/2$ and $\beta\varepsilon$ is the energy scale. The snapshot considered here, depicted in Fig. 10(a), was obtained from a MC simulation in the isothermal-isobaric ensemble (constant number of particles N, pressure P, and temperature T) with $N = 8073$ particles, $\beta\varepsilon = 5$, and $\beta P\sigma_L^3 = 24.2$, and shows a coexistence between the MgCu$_2$ Laves phase and the fluid phase. More details about the simulation can be found in Ref. 44.

Recall that, since we are dealing with a binary mixture, we describe the local environment of each particle i in the system with a 16-dimensional vector of BOPs, $\mathbf{Q}(i) = (\{\bar{q}_l(i)\}, \{\bar{q}_l^{ss}(i)\})$, where the first set of 8 BOPs, $\{\bar{q}_l(i)\}$ with $l = 1, \ldots, 8$, is evaluated considering all the nearest neighbors of particle i, while the second set of 8 BOPs, $\{\bar{q}_l^{ss}(i)\}$, is evaluated considering only the nearest neighbors of the same species ($s = L$ or $s = S$) as particle i.

The results of the unsupervised learning classification are summarized in Fig. 10. Specifically, Fig. 10(b) shows the two-dimensional projection of the vectors $\mathbf{Q}(i)$ found by the encoder and the results of the clustering performed in this space. Our method identifies three distinct particle environments, corresponding to the large (L) and small (S) particles in the MgCu$_2$ Laves phase and the particles in the fluid phase. Note that since large and small particles in the fluid phase appear equally disordered, the algorithm groups them together in the same cluster. This identification is used to color the particles in Fig. 10(a).

The relative importance of the BOPs obtained from the autoencoder analysis is shown in Fig. 11. Interestingly, the BOPs evaluated considering all the nearest neighbors of the particles are not found to be relevant for this system, with the only exception of $\bar{q}_6$ (RI ~ 2%). The largest variations in the environments are instead found in three of the BOPs evaluated considering only the nearest neighbors of the same species as the reference particle, specifically $\bar{q}_6^{ss}$ (RI ~ 70%), $\bar{q}_8^{ss}$ (RI ~ 15%), and $\bar{q}_4^{ss}$ (RI ~ 11%).

E. Nucleation and crystal growth

Finally, we examine the nucleation and crystal growth of a single-component system of particles with diameter σ, interacting via the WCA potential [Eq. (10)]. Note that the nucleation and phase

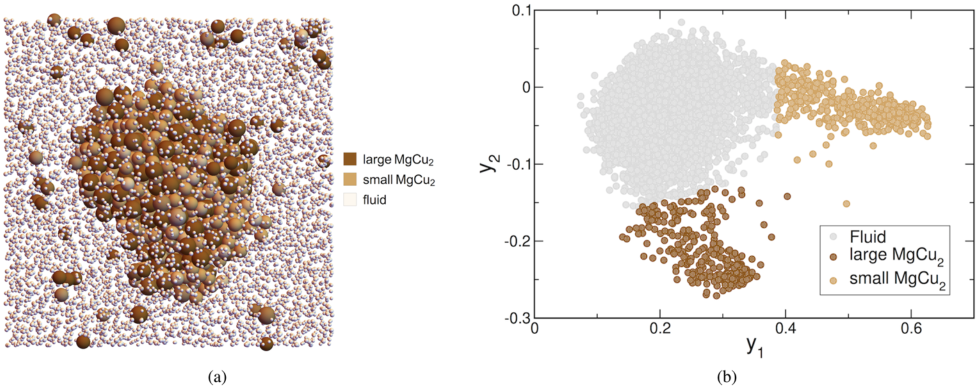


FIG. 10. Analysis of a snapshot of a binary mixture showing a coexistence between the fluid phase and the MgCu$_2$ Laves phase. (a) Classification of the snapshot under analysis. The RGB color of each particle is a linear combination of the colors of the three local environments identified with the associated membership probabilities as coefficients. Fluidlike particles are displayed at 1/4 of their actual size. (b) Projection of the vectors **Q**(i) onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering.

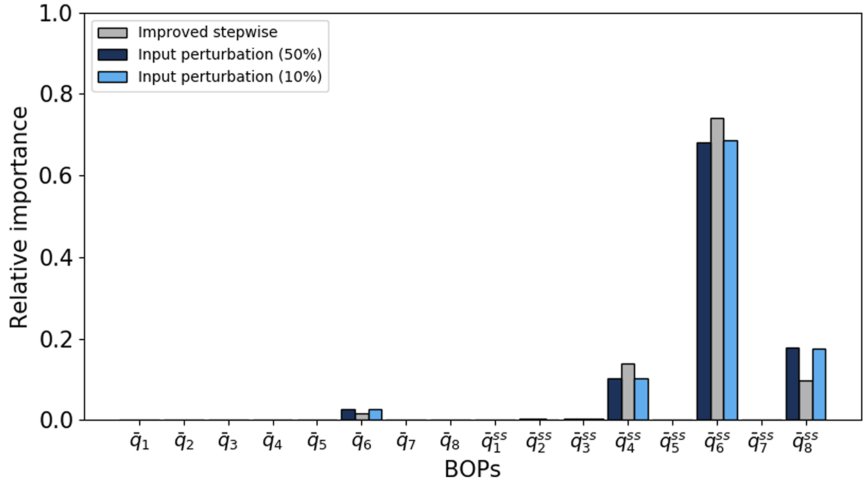


FIG. 11. Relative importance of the BOPs for the system in Fig. 10 assessed with the input perturbation and the improved stepwise methods. The input perturbation method is applied with two different amounts of white noise: 10% and 50% of the input, respectively.

behavior of this system have been extensively studied in Ref. 45. Here, we performed a MC simulation in the isothermal-isobaric ensemble with $N = 2048$ particles, $\beta\varepsilon = 40$ and $\beta P\sigma^3 = 30$. Six snapshots from this simulation, showing the transition from the fluid to the crystal phase, are depicted in Fig. 12.

We started our structural analysis by considering all snapshots together. For all particles, we evaluated the vectors $\mathbf{Q}(i)$, which we used as the input of the unsupervised learning algorithm. Performing a single analysis for all the snapshots in Fig. 12 guarantees that (i) sufficient examples of the two environments, i.e., fluid and

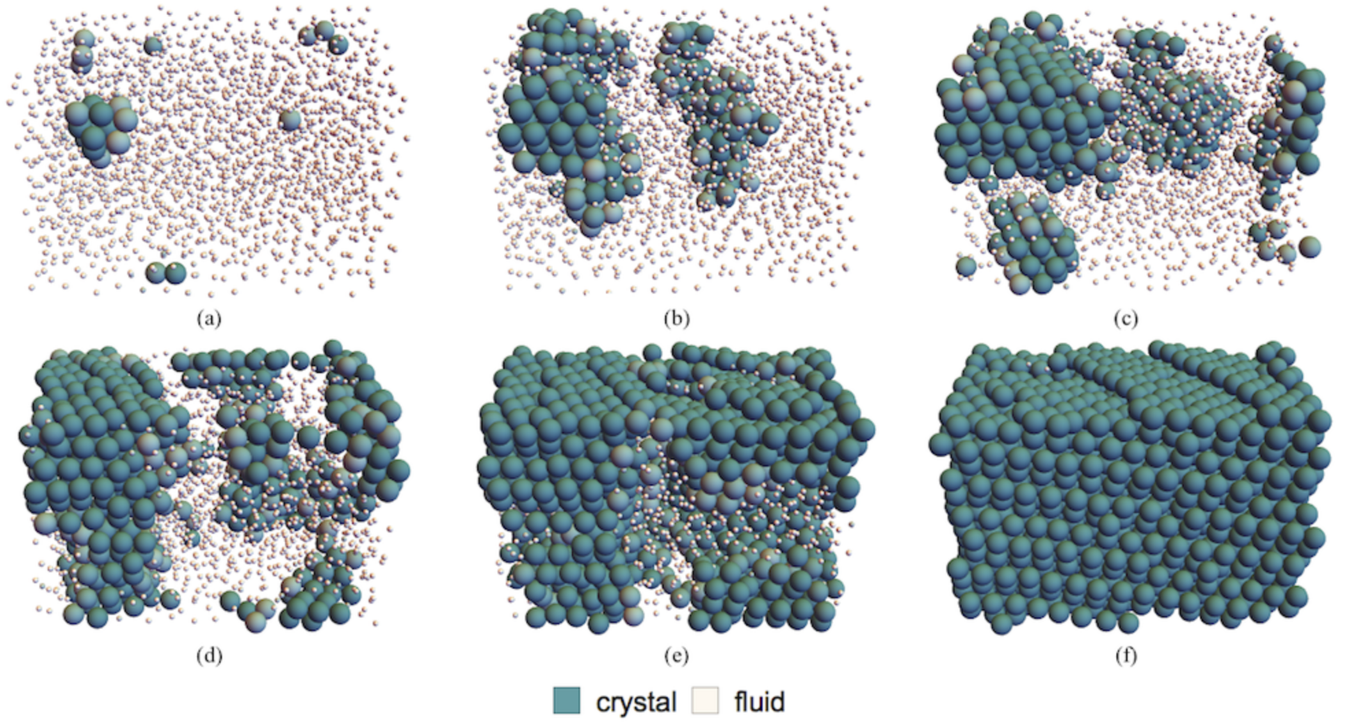


FIG. 12. Classification of snapshots of a MC simulation with $N = 2048$ particles interacting via the WCA potential, showing a transition from the fluid to the crystal phase. The RGB color of each particle is a linear combination of the colors of the two phases identified with the associated membership probabilities as coefficients. Fluidlike particles are displayed at 1/4 of their actual size.

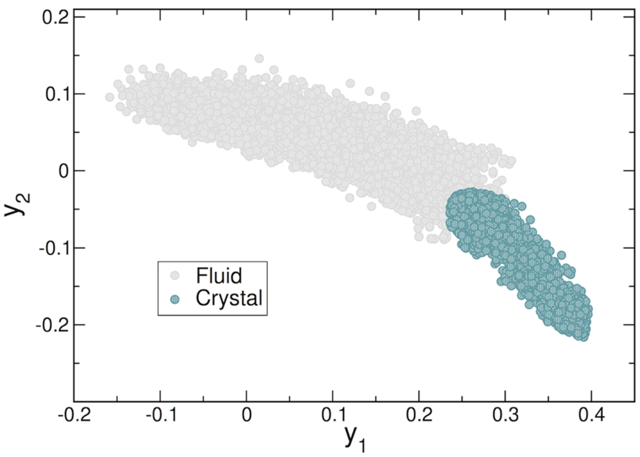


FIG. 13. Projection of the vectors $\mathbf{Q}(i)$ evaluated from the six snapshots in Fig. 12 onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering.

crystalline, are included in the analysis and (ii) the same clustering is applied to each snapshot, allowing a quantitative comparison between them.

The two-dimensional projection found by the encoder and the results of the clustering performed in this space are shown in Fig. 13. This identification is used to color the particles in Fig. 12. As expected, the unsupervised learning method identifies two distinct particle environments, corresponding to the fluid and crystal phases. In order to quantitatively compare the results with the standard classification method presented in Sec. III A 2, we evaluated the fraction of crystalline particles identified in the six snapshots. Note that, since we employ a soft clustering technique, this requires first to assign each particle to the cluster (crystal or fluid) with the largest membership probability. The results obtained with the two methods are presented in Table I and are in excellent agreement.

One of the reasons behind performing a single analysis for the six snapshots in Fig. 12 was to include sufficient examples of both the fluid and crystalline environments. Our unsupervised learning method consists of a two-step analysis: first, the autoencoder finds a low-dimensional projection encoding the features with the largest variations within the input data, and then, the clustering algorithm identifies distinct environments based on the density distribution of the data in this low-dimensional space. In the method presented in Sec. III A 2, instead, a separation between the distinct phases is chosen based on the corresponding distributions of the relevant BOPs obtained from separate simulations. By including sufficient examples of both phases in the input dataset, the results of the unsupervised classification tend to (or at least are very similar to) those of the method in Sec. III A 2, as demonstrated by the excellent agreement we found. However, what "sufficient" means strongly depends on the system under study.

TABLE I. Comparison between the fractions of crystalline particles identified in the six snapshots in Fig. 12 with our unsupervised algorithm and with the standard method presented in Sec. III A 2.

	Snapshot					
Method	(a)	(b)	(c)	(d)	(e)	(f)
Unsupervised (%)	2	12	21	30	64	98
Standard (%)	2	13	23	31	67	98

To obtain a better estimate of what "sufficient" means for this system, we performed a separate analysis of each of the snapshots in Fig. 12. For snapshots (b), (c), (d), and (e), we found very similar results to the previous analysis, while, as expected, for snapshots (a) and (f), where only a very small fraction of particles is in one of the phases, we found a different classification. As an example, we report in Fig. 14 the results obtained for snapshot (a). Specifically, in Fig. 14(b), we show the two-dimensional projection found by the encoder and the results of the clustering performed in this space. Note that this projection is different from the one presented in Fig. 13 as the autoencoder is trained on a different data set with different characteristics. Additionally, in Fig. 14(a), particles are colored according to this classification. Again, the unsupervised learning identified two distinct environments, which we associate with crystalline and fluidlike particles. However, the amount of crystalline order is much larger compared to the previous classification: about

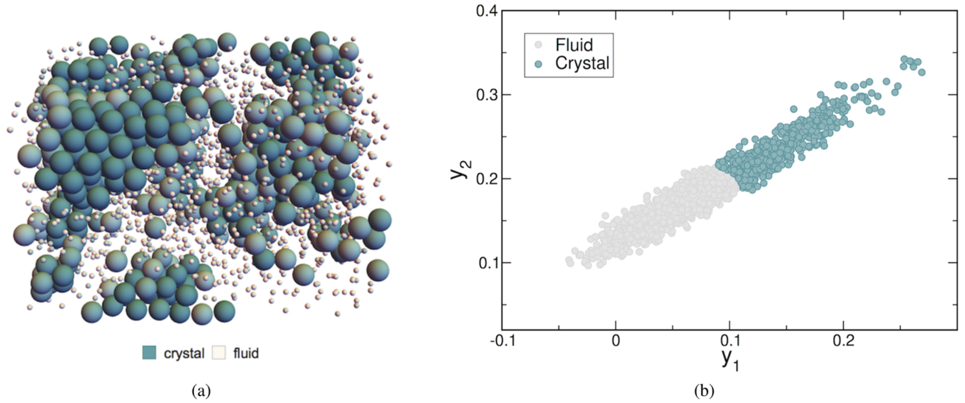


FIG. 14. Separate analysis of the snapshot in Fig. 12(a). (a) Classification of the snapshot under analysis. The RGB color of each particle is a linear combination of the colors of the two environments identified with the associated membership probabilities as coefficients. Fluidlike particles are displayed at 1/4 of their actual size. (b) Projection of the vectors $\mathbf{Q}(i)$ onto the 2-dimensional space found by the encoder. Colors represent the distinct environments identified by the clustering.

20% of the particles are classified as crystalline (in the previous analysis, it was only 2%). Interestingly, if we look closer at the snapshot in Fig. 14(a), we find that the particles recognized as crystalline have indeed a higher local order than in a standard disordered fluid, meaning that the unsupervised learning classification is still reasonable. This has been seen before in nucleation studies. For instance, in Ref. 46, where hard-sphere nucleation was studied at a pressure of $\beta P\sigma^3 = 17$, slightly different tuning of the order parameter resulted in significantly different results for the size of the largest crystalline cluster in the system (from ~30 to 100). Nonetheless, the different order parameters all predicted essentially the same nucleation rate.

IV. CONCLUSIONS

In summary, we have introduced a simple, fast, and easy to implement unsupervised learning algorithm for recognizing local structural motifs in colloidal systems. This algorithm makes use of standard BOPs to describe local environments, an autoencoder for dimensionality reduction, and GMMs for clustering the results. We have applied it to a wide variety of systems, ranging from simple isotropically interacting systems (hard spheres, WCA particles) to binary mixtures, and even anisotropic hard cubes. In all cases, the algorithm performed very well, and we were able to identify local environments to a similar precision as "standard"—manually tuned and system-specific—order parameters.

Moreover, we exploited the analytical mapping defined by the autoencoder to extract extra information on the systems analyzed. Specifically, in all cases, we could identify the relevant symmetries underlying the main differences among the distinct particle environments found in the system. Interestingly, in the binary system we studied, this analysis revealed that the same species BOPs were the most important for distinguishing the different particle environments. Finally, in the last example on nucleation and crystal growth, we also explored the possible difficulties one can encounter when only a very small fraction of particles is in one specific environment, and we showed how including more snapshots, which covered a more balanced distribution of local environments, could benefit the results of the analysis in such cases.

ACKNOWLEDGMENTS

We would like to thank Frank Smallenburg, Gabriele Maria Coli, and Berend van der Meer for providing us with some of the configurations we analyzed in this paper. We would also like to thank Sudeep Punnathanam, Frank Smallenburg, Alfons van Blaaderen, and Sela Samin for many useful discussions.

L.F. and E.B. gratefully acknowledge funding from The Netherlands Organisation for Scientific Research (NWO) (Grant No. 16DDS004).

REFERENCES

[1] S. Auer and D. Frenkel, Nature **409**, 1020 (2001).
[2] U. Gasser, E. R. Weeks, A. Schofield, P. N. Pusey, and D. A. Weitz, Science **292**, 258 (2001).
[3] M. Hermes, E. C. M. Vermolen, M. E. Leunissen, D. L. J. Vossen, P. D. J. van Oostrum, M. Dijkstra, and A. van Blaaderen, Soft Matter **7**, 4623 (2011).
[4] E. Sanz, C. Valeriani, T. Vissers, A. Fortini, M. E. Leunissen, A. Van Blaaderen, D. Frenkel, and M. Dijkstra, J. Phys.: Condens. Matter **20**, 494247 (2008).
[5] A. van Blaaderen and P. Wiltzius, Science **270**, 1177 (1995).
[6] B. van der Meer, M. Dijkstra, and L. Filion, Soft Matter **12**, 5630 (2016).
[7] T. O. E. Skinner, D. G. A. L. Aarts, and R. P. A. Dullens, Phys. Rev. Lett. **105**, 168301 (2010).
[8] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, Phys. Rev. B **28**, 784 (1983).
[9] P. R. Ten Wolde, M. J. Ruiz-Montero, and D. Frenkel, Phys. Rev. Lett. **75**, 2714 (1995).
[10] P. Rein ten Wolde, M. J. Ruiz-Montero, and D. Frenkel, J. Chem. Phys. **104**, 9932 (1996).
[11] W. Lechner and C. Dellago, J. Chem. Phys. **129**, 114707 (2008).
[12] J. D. Honeycutt and H. C. Andersen, J. Phys. Chem. **91**, 4950 (1987).
[13] D. Faken and H. Jónsson, Comput. Mater. Sci. **2**, 279 (1994).
[14] N. Duff and B. Peters, J. Chem. Phys. **135**, 134101 (2011).
[15] W. Lechner, C. Dellago, and P. G. Bolhuis, Phys. Rev. Lett. **106**, 085701 (2011).
[16] P. Geiger and C. Dellago, J. Chem. Phys. **139**, 164105 (2013).
[17] C. Dietz, T. Kretz, and M. H. Thoma, Phys. Rev. E **96**, 011301 (2017).
[18] E. Boattini, M. Ram, F. Smallenburg, and L. Filion, Mol. Phys. **116**, 3066 (2018).
[19] R. S. DeFever, C. Targonski, S. W. Hall, M. Smith, and S. Sarupria, Chem. Sci. **10**, 7503 (2019).
[20] W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson, and A. Z. Panagiotopoulos, Soft Matter **13**, 4733 (2017).
[21] W. F. Reinhart and A. Z. Panagiotopoulos, Soft Matter **14**, 6083 (2018).
[22] M. Spellings and S. C. Glotzer, AIChE J. **64**, 2198 (2018).
[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press, Cambridge, MA, USA, 1986), Vol. 1, pp. 318–362.
[24] M. A. Kramer, AIChE J. **37**, 233 (1991).
[25] M. Scholz and R. Vigário, in ESANN, 2002.
[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (The MIT Press, 2016).
[27] C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Inc., New York, NY, USA, 1995).
[28] A. Dempster, N. Laird, and D. Rubin, J. R. Stat. Soc.: Ser. B **39**, 1 (1977).
[29] G. Schwarz, Ann. Stat. **6**, 461 (1978).
[30] J. Baudry, A. Raftery, G. Celeux, K. Lo, and R. Gottardo, J. Comput. Gr. Stat. **19**, 332 (2010).
[31] W. Rycroft, Chaos **19**, 041111 (2009).
[32] J. A. van Meel, L. Filion, C. Valeriani, and D. Frenkel, J. Chem. Phys. **136**, 234107 (2012).
[33] X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256.
[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature **323**, 533 (1986).
[35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, in *Proceedings of the 30th International Conference on Machine Learning* (JMLR, 2013), pp. 1139–1147.
[36] W. Chen, A. R. Tan, and A. L. Ferguson, J. Chem. Phys. **149**, 072312 (2018).
[37] S. Salvador and P. Chan, in *16th IEEE International Conference on Tools with Artificial Intelligence* (IEEE, 2004), p. 576.
[38] J. Yao, N. Teng, H.-L. Poh, and C. L. Tan, J. Inf. Sci. Eng. **14**, 843 (1998).
[39] M. Scardi and L. W. Harding, Ecol. Modell. **120**, 213 (1999).
[40] M. Gevrey, I. Dimopoulos, and S. Lek, Ecol. Modell. **160**, 249 (2003).
[41] J. D. Olden, M. K. Joy, and R. G. Death, Ecol. Modell. **178**, 389 (2004).
[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, J. Mach. Learn. Res. **12**, 2825 (2011).
[43] F. Smallenburg, L. Filion, M. Marechal, and M. Dijkstra, Proc. Natl. Acad. Sci. U. S. A. **109**, 17886 (2012).
[44] T. Dasgupta, G. M. Coli, and M. Dijkstra, preprint arXiv:1906.10680 (2019).
[45] L. Filion, R. Ni, D. Frenkel, and M. Dijkstra, J. Chem. Phys. **134**, 134901 (2011).
[46] L. Filion, M. Hermes, R. Ni, and M. Dijkstra, J. Chem. Phys. **133**, 244115 (2010).