Classification of complex local environments in systems of particle shapes through shape-symmetry encoded data augmentation

Shih-Kuang (Alex) Lee,[1] Sun-Ting Tsai,[2] and Sharon C. Glotzer*[2, 3]

[1)] *Department of Material Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA.*

[2)] *Department of Chemical Engineering, University of Michigan, Ann Arbor, MI 48109, USA.*

[3)] *Biointerfaces Institute, University of Michigan, Ann Arbor, MI 48109, USA.*

(*Electronic mail: sglotzer@umich.edu)

(Dated: 25 March 2024)

ABSTRACT

Detecting and analyzing the local environment is crucial for investigating the dynamical processes of crystal nucleation and shape colloidal particle self-assembly. Recent developments in machine learning provide a promising avenue for better order parameters in complex systems that are challenging to study using traditional approaches. However, the application of machine learning to self-assembly on systems of particle shapes is still underexplored. To address this gap, we propose a simple, physics-agnostic, yet powerful approach that involves training a multilayer perceptron (MLP) as a local environment classifier for systems of particle shapes, using input features such as particle distances and orientations. Our MLP classifier is trained in a supervised manner with a shape symmetry-encoded data augmentation technique without the need for any conventional roto-translations invariant symmetry functions. We evaluate the performance of our classifiers on four different scenarios involving self-assembly of cubic structures, 2-dimensional and 3-dimensional patchy particle shape systems, hexagonal bipyramids with varying aspect ratios, and truncated shapes with different degrees of truncation. The proposed training process and data augmentation technique are both straightforward and flexible, enabling easy application of the classifier to other processes involving particle orientations. Our work thus presents a valuable tool for investigating self-assembly processes on systems of particle shapes, with potential applications in structure identification of any particle-based or molecular system where orientations can be defined.

I. INTRODUCTION

Self-assembly is studied extensively in such fields as physics, chemistry, materials science, chemical engineering and biology[1,2]. A fundamental process involving thermodynamics and kinetics, self-assembly refers to the formation of ordered structures from individual building blocks or particles without direction from an external field. In recent years, the development of self-assembled structures from sub-micron-sized particle building blocks has attracted considerable attention due to their potential applications in nanotechnology[3–10]. An important challenge in elucidating and, eventually, engineering assembly pathways to optimize target structures is defining appropriate order parameters that quantify local order in the assembling structures along the pathway[11–15]. Defining suitable local order parameters is particularly challenging when the self-assembling structure is complex (e.g. a large unit cell, possessing chirality, etc.) or when competing polymorphs or pre-nucleation motifs emerge along the pathway[16–18].

Previous studies have attempted to identify suitable order parameters for self-assembly, including using the radial distribution function, the bond-orientational order parameter[15,19–21], and Voronoi tessellation[19,22,23]. While these order parameters have been used extensively and successfully in capturing some aspects of the assembly process, they have limitations. For example, the radial distribution function (RDF) considers only the pairwise (two-point) correlation between particles and does not capture higher-order correlations. The bond-orientational order parameter is sensitive to local structure but is less effective in detecting global ordering. Methods using the Voronoi tessellation can elucidate local structure but are less effective in describing long-range order. In particular, these approaches are insufficient for interrogating, e.g., the self-assembly of a system of truncated tetrahedra into a crystal with 432-particle unit cell[17].

Machine learning (ML) is becoming an increasingly popular approach for discovering order parameters useful in the study of self-assembly[12–14,24–28]. One ML approach uses existing order parameters that combine roto-translation-invariant symmetry functions as input descriptors and approximates optimal order parameters[12,29]. However, an immediate shortcoming of this type of supervised ML method is its limited classification capabilities, which are constrained by its input descriptors. If none of the input descriptors can classify a certain crystal structure, the machine learning methods will fail to classify that crystal structure. Training deep neural networks (NNs) to classify crystal phases is another increasingly common approach[15,21,30–33]. Deep NNs can identify nonlinearity and are especially promising in constructing order parameters from particle features, such as instantaneous positions along a trajectory. Since thermal fluctuations often lead to noisy data, however, applying symmetry functions to encode particle positions is usually inevitable[34]. These pre-engineered symmetry functions can work well for some systems, but finding symmetry functions that encode particle orientations is non-trivial. Finally, graph neural net-

works (GNNs) have also become popular for classifying crystal structures[35–37]. By treating crystal structures as connected graphs, GNNs preserve permutation symmetry. However, they often require additional manipulation to deal with rotational and translational symmetry[29]. Although equivariant GNNs have recently been proposed[38], the question of how to incorporate equivariant properties for particle shapes remains to be addressed. Common to all of these ML approaches is the tendency for increasing complexity in the approach as the building blocks and the structures they form become increasingly complex.

In this work, we show how we can use the most basic NN, a multilayer perceptron (MLP), as a local environment classifier in systems of particle shapes. Because we use the MLP classifier to quantify the local environment around each particle, the classifier is permutation-invariant, precluding the use of more sophisticated network structures such as GNNs[37]. Thus, instead of employing conventional symmetry functions to transform per-particle quantities to input descriptors, we analyze step-by-step the symmetry of a particle's shape and propose a straightforward shape-symmetry encoded data augmentation method that allows our MLP classifier to operate on per-particle features directly. This data augmentation method ensures the roto-translation invariance of input features to the global environment and accounts for the particle shape's symmetry. Importantly, the simplicity of our data augmentation method necessitates minimal manipulation of the raw data, and also leads to a substantial improvement in the classification performance and flexibility when distinguishing different thermodynamic phases in our test systems. In this way, our approach can provide a simple, powerful, physics-agnostic alternative to conventional order parameters when studying self-assembly.

The remainder of this article is organized as follows. In Sec. II A, we introduce per-particle quantities as input features. Next, in Sec. II B, we show how we perform data augmentation on input features based on particle shape symmetry. In Sec. II C, we introduce the MLP model we used as a local environment classifier. In Sec. III, we demonstrate our method on seven different test cases. We first test our classifier's stability in Sec. III A for a simple cubic structure self-assembled in simulation from hard cubes. Subsequently, we test our classifier on six additional self-assembly examples that result in three different categories of final products. In each case, we show how data augmentation improves classification quality. A conclusion is given in Sec. III D. We have also created a repository that grants complete access to all trajectory data, code and scripts, enabling users to reproduce our work[39,40].

II. METHOD

We construct a classifier for the local environment around a particle using an MLP that incorporates continuous, discrete and shape symmetry information. By building an MLP classifier that learns to classify each particle, we ensure that the classification is permutation invariant. We first extract features from each particle's local environment and then use these features to train MLP classifiers to accurately classify particles based on their local environment.

A. Per-particle quantities as input features

To account for translational symmetry and describe the local environment of a target particle in a system of like particles, we employ a set of interparticle quantities defined with respect to the surrounding neighborhood. Specifically, we calculate the **relative positions** $\mathbf{r}_{ij}$ and **relative orientations** defined by quaternion $\mathbf{q}_{ij}$ of a predetermined number of neighboring particles j in relation to the target particle i. These quantities serve as a unique fingerprint of the local environment surrounding the target particle[15,42] and are defined below.

We define the relative position $\mathbf{r}_{ij}$ using spherical coordinates $(r_{ij}, \theta_{ij}, \phi_{ij})$, where the relative distances r_{ij} between a fixed number N_b of neighboring particles denoted by the index j (that is, the size of the neighborhood) and the target particle i contain information about the local density distribution. The angular part (θ_{ij}, ϕ_{ij}) define the local bond angles. Note that for a two-dimensional system, θ_{ij} is always 0. However, despite the invariance of the relative distances r_{ij} to arbitrary translation operations, these distances can still be affected by thermal fluctuations and the length scale of the system, which can impact the transferability of our classifier. Thus, we normalized the relative distances:

$$r_{ij}^{(\mathrm{au})} = \frac{r_{ij}}{\max_{j\in N_b} r_{ij}} \tag{1}$$

where the superscript $^{(\mathrm{au})}$ denotes the augmented features.

The relative quaternion $\mathbf{q}_{ij}$ is commonly used to define the relative orientation of a neighboring particle j with respect to the target particle i. The quaternion plays an important role in both simulating and analyzing crystallization of a system of anisotropic particles[43,44] and in defining the space group of crystals of certain molecular systems. To describe a particle's orientation, a reference orientation first needs to be established using the important symmetry axis of the particle along the orthogonal basis of Cartesian coordinates. We denote the reference orientation by $\mathbf{q}_0 = (1,0,0,0)$, which is equivalent to the identity rotation with respect to the reference orientation. In Fig. 1, we illustrate the particles studied in this paper by placing them in the predefined reference orientation. Given this reference orientation, we can define subsequent orientations of each particle by performing a 3-dimensional spatial rotation, which can be expressed as a rotation quaternion $\mathbf{q} = (C, Su^x, Su^y, Su^z)$ where $(C,S) = (\cos(\theta/2), \sin(\theta/2))$. This $\mathbf{q}$ thus represents a rotation angle θ from the reference orientation about the axis $\mathbf{u}$. The relative orientation between target particle i and its neighborhood j can then be expressed as the rotation quaternion via the conventional rotation from i to j as:

$$\mathbf{q}_{ij} = \mathbf{q}_i^{-1}\mathbf{q}_j \tag{2}$$

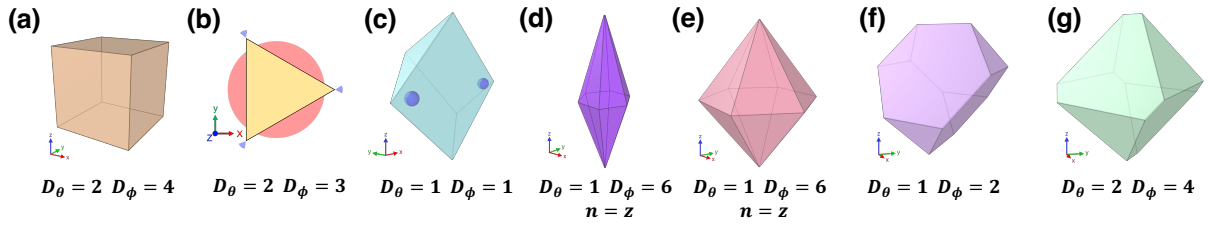

FIG. 1. **Systems of particle shapes** The particle systems studied in this paper: **(a)** cube, **(b)** patchy triangle, **(c)** patchy triangular prism[41], **(d)** hexagonal bipyramid with aspect ratio $\alpha = 3.0$, **(e)** hexagonal bipyramid with $\alpha = 1.28$, **(f)** truncated tetrahedron, and **(g)** truncated octahedron. In (a) and (b), the transparent blue region decorating the particles indicates the attractive patch, while the transparent red region indicates a repulsive patch. Also shown are the shape symmetry-related factors D_θ, D_ϕ, and mirror plane normal vector $\mathbf{n}$ utilized for each system. Each particle pictured here is oriented such that it is in the reference orientation given by the quaternion $\mathbf{q}_0 = (1,0,0,0)$.

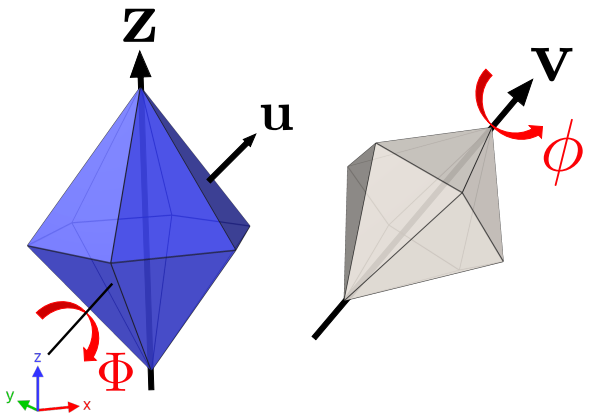

FIG. 2. **Rotation in the local environment** The blue bipyramid represents the target particle, and the white bipyramid represents a neighboring particle.

By utilizing relative position and orientation, we can accurately capture each particle's local environment while maintaining translational invariance. We have yet to discuss the property of invariance to an arbitrary rotation of the system, which can influence the angular parts θ_{ij}, ϕ_{ij} that are used to define relative position and $\mathbf{q}_{ij}$. Conventionally, the rotational invariance is achieved by randomly rotating the training set in each training epoch[36]; this step is referred to as data augmentation. In the next section, we present a more robust approach for training a classifier invariant to an arbitrary system rotation of particle environments.

B. Shape-symmetry encoded data augmentation

In the previous section, we introduced relative positions and quaternions as input features. These input features are designed so they are invariant to the translation of the system. However, arbitrary rotation of the system can also limit the transferability of the classifier, which is traditionally addressed using data augmentation techniques[27]. Note that our data augmentation is different from the data augmentation used in the training of convolution neural networks, where random orientations are applied to generate modified copies of image data to allow the network to recognize different symmetry of the images. For example, in image processing, one duplicates the image but with random rotations to create a training dataset to prevent possible overfitting and enhance the transferability of the classifier. Recent developments in equivariant NNs (ENNs) also allow for input without data augmentation[38,45,46]. For systems of point particles, the final crystal structure can be used to define the reference orientation. Here we exploit particle shape and use the user-defined reference orientation represented by quaternions to define the local environment. To ensure rotational invariance, we independently rotate each target particle and its local environment onto the predefined reference orientation, such that each target particle is in the predefined reference orientation as shown in Fig. 1 prior to calculating the per-particle quantities.

In addition to translational and rotational symmetries associated with the system of particles, the individual particles can possess symmetries. With a sophisticated design, the ENNs can process these additional symmetries during training. However, we propose a simple yet effective way to achieve the same result here. We perform an additional data augmentation that encodes the particle's shape and interaction anisotropy (patchiness). The new angular part of the relative position and relative quaternion after this data augmentation is denoted as $(\theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})})$ and $\mathbf{q}_{ij}^{(\mathrm{au})}$. Through simple geometric reasoning, the augmented angular part can be calculated easily as follows:

$$(\theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})}) = \left(\frac{\theta_{ij}}{\pi} \bmod \frac{1}{D_\theta}, \frac{\phi_{ij}}{\pi} \bmod \frac{2}{D_\phi} \right) \tag{3}$$

where mod is the modulo operator, and D_ϕ and D_θ are discrete integers that assume the value, N of the N-fold rotational symmetry along ϕ and θ-direction with respect to the particles' shape when it is in the reference orientation. In Fig. 1, we show D_ϕ and D_θ for each of the corresponding hard shapes and patchy particles in their reference orientations.

The orientation quaternion $\mathbf{q}_{ij}$ ican be written as a rotation of angle Φ along an axis $\mathbf{u}$, as shown in Fig. 2 It can be proved that the same quaternion can also be written as the rotation of angle ϕ along axis $\mathbf{v}$,

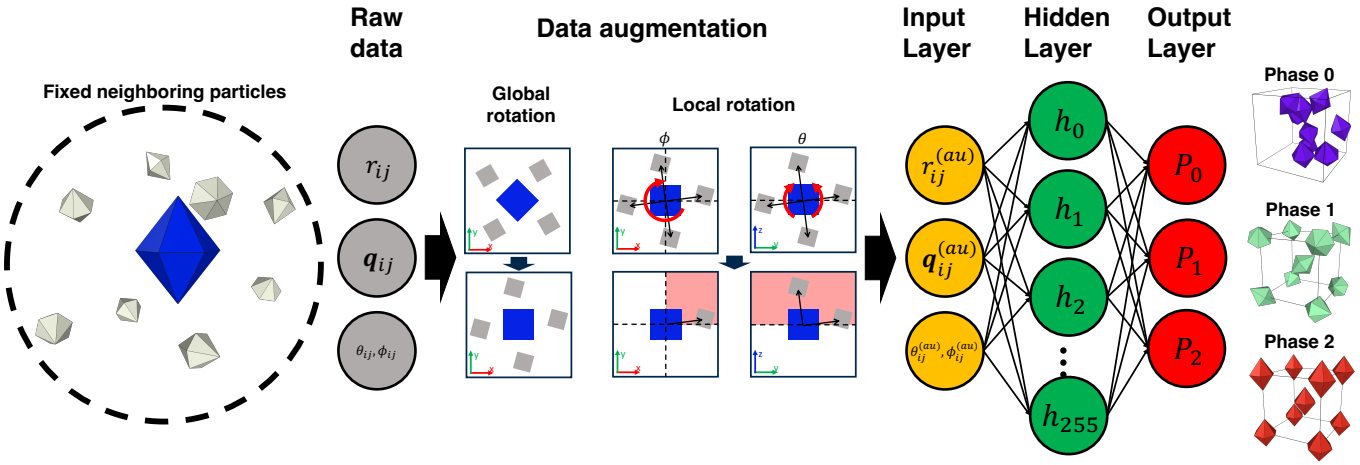

FIG. 3. **The data augmentation and the architecture of Multilayer Perceptron (MLP)** The diagram of the data augmentation and the MLP architecture used to classify local particle environment. The raw local environment data is augmented through sets of rotations and passed to the input layer of MLP. The input layer has $N_b + 4N_b + 2N_b$ units as described in Eq. 9, and the hidden layer contains 256 units. The number of output layers depends on the system. For example, we will present results for a system of particles undergoing a two-steps crystallization transition, where the output layer is composed of three units representing the probability that a given particle belongs to one of three predefined phases.

$$\mathbf{q}_{ij} = e^{i\frac{\Phi}{2}\mathbf{u}} = e^{i\frac{\phi}{2}\mathbf{v}}\mathbf{Q} \tag{4}$$

where $\mathbf{v} = \mathbf{Q}\mathbf{z}\mathbf{Q}^{-1}$ defines the symmetry axis of the particle. The symmetry axis is defined as the z-axis when each particle is in its reference orientation and rotates in the same way the particle rotates. The augmented quaternion is then calculated as

$$\mathbf{q}_{ij}^{(\mathrm{au})} = e^{i\frac{\phi^{(\mathrm{au})}}{2}\mathbf{v}^{(\mathrm{au})}} \tag{5}$$

where $\phi^{(\mathrm{au})} = \phi \mod \frac{2\pi}{D_\phi}$ and

$$\mathbf{v}^{(\mathrm{au})} = \mathbf{v}_{\perp\mathbf{z}} + \cos\theta^{(\mathrm{au})}\mathbf{z} \tag{6}$$

where $\mathbf{v}_{\perp\mathbf{z}}$ is defined as the component perpendicular to $\mathbf{z}$ and $\theta^{(\mathrm{au})} = \arccos(\mathbf{v}\cdot\mathbf{z}) \mod \frac{\pi}{D_\theta}$.

Under the same rule, it is also straightforward to consider mirror symmetry. For the angular part, in general, we can calculate augmented relative positions $\mathbf{r}_{ij}^{(\mathrm{au})}$ before separating relative positions into distance r_{ij} and angular parts (θ_{ij},ϕ_{ij}):

$$\mathbf{r}_{ij}^{(\mathrm{au})} = (\mathbf{r}_{ij})_{\perp\mathbf{n}} + \|\mathbf{r}_{ij}\cdot\mathbf{n}\|\mathbf{n} \tag{7}$$

where $(\mathbf{r}_{ij})_{\perp\mathbf{n}} = \mathbf{n}\times(\mathbf{r}_{ij}\times\mathbf{n})$ is defined as the component perpendicular to $\mathbf{n}$. In practice, since the mirror plane of a hexagonal bipyramid is along the $\mathbf{z}$-axis, we need only to take the absolute value of the $\mathbf{z}$-coordinate before separating relative positions into distance and angular parts. For the quaternion part, we need to separate the symmetry axis into normal and parallel parts with respect to the plane normal vector $\mathbf{n}$:

$$\mathbf{v}^{(\mathrm{au})} = \mathbf{v}_{\perp\mathbf{n}} + \|\mathbf{v}\cdot\mathbf{n}\|\mathbf{n} \tag{8}$$

where $\mathbf{v}_{\perp\mathbf{n}} = \mathbf{n}\times(\mathbf{v}\times\mathbf{n})$ is defined as the component perpendicular to $\mathbf{n}$.

C. Multilayer Perceptron (MLP) as local environment classifier

The local symmetry is broken during a self-assembly process as the local environment around a particle changes. To monitor the change in the local environment, we utilize a fully connected NN, commonly known as a multilayer perceptron (MLP), to classify this local environment, as illustrated in Fig. 3. There are several advantages of using a simple MLP instead of a more advanced GNNs or ENNs. First, an MLP handles large datasets with simplicity and effectiveness. An MLP allows us to train the machine even on a personal laptop, suitable for a quick, on-the-fly test. Additionally, an MLP can be easily extended to incorporate additional features or classify particles in other colloidal systems. Second, a simple MLP can be greatly accelerated by harnessing the power of modern graphical processing units (GPUs), which can facilitate future research, such as using MLPs as order parameters to study the assembly pathways of disparate systems forming the same structure, or in enhanced sampling methods such as umbrella sampling or metadynamics whose algorithms demand efficient calculation of order parameters and their derivatives. Third, despite its simple network structure, a MLP still provides sufficient nonlinearity to build a powerful classifier from fundamental features, e.g., particle coordinates and orientations. Its classification ability is decent enough to map local environmental fingerprints of particle shapes to thermodynamic phases.

The input to the MLP classifier is a set of feature vectors $(r_{ij}^{(\mathrm{au})}, \mathbf{q}_{ij}^{(\mathrm{au})}, \theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})})$, as described in Sec. II A and II B. These feature vectors are arranged and concatenated to a 1-dimensional vector $\mathbf{x}_i$ with the relative position and relative

quaternion sequentially placed in ascending order as follows,

$$\mathbf{x}_i = \text{sort}_{r_{ij}^{(\text{au})}}(..., r_{ij}^{(\text{au})}, \mathbf{q}_{ij}^{(\text{au})}, \theta_{ij}^{(\text{au})}, \phi_{ij}^{(\text{au})}, ...) \tag{9}$$

where j runs over all neighbors and therefore $\mathbf{x}_i \in \mathbb{R}^{N_b+4N_b+2N_b}$. As shown in Fig. 3, the MLP architecture includes one hidden layer consisting of 256 neurons. The input layer of the network takes the feature vectors $\mathbf{x}_i$ and propagates them forward to the hidden layer, which performs a linear operation followed by a non-linear activation function σ:

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x}_i + \mathbf{b}) \tag{10}$$

The activation function comprises a linear operation defined by a layer-wise matrix multiplication with trainable weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$.

The hidden layer $l = 0$ takes input feature vectors $\mathbf{x}_i$ directly, and therefore has dimensions $\mathbf{W} \in \mathbb{R}^{\mathscr{M}\times\mathscr{N}}$ and $\mathbf{b} \in \mathbb{R}^{\mathscr{M}}$, where $\mathscr{M} = 256$ and $\mathscr{N} = N_b + 4N_b + 2N_b$. After this linear operation, we used a rectified linear unit (ReLU) as the activation function σ. The output from the hidden layer is then converted to output nodes $\mathbf{y} \in \mathbb{R}^{\mathscr{C}}$ in the output layer,

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}^{(o)}\mathbf{h} + \mathbf{b}^{(o)}) \tag{11}$$

where $\mathbf{W}^{(o)} \in \mathbb{R}^{\mathscr{M}\times\mathscr{C}}$ and $\mathbf{b}^{(o)} \in \mathbb{R}^{\mathscr{C}}$ are the output weight matrix and output bias, respectively, and $\mathscr{C}$ is the number of predefined classes. The softmax function in eq. 11 is a mathematical function that converts a vector of real numbers into a probability distribution. The MLP classifiers are then trained using the optimizer Adam[47], with a learning rate of 7.5×10^{-4}, and with an error metric given by the cross-entropy loss L, defined as:

$$L = -\sum_{n=1}^{\mathscr{C}} \log \hat{\mathbf{y}}_n \tag{12}$$

The cross-entropy loss L, or log loss, measures the performance of a probabilistic classification model whose output is a probability distribution. For simplicity, all of our MLP classifiers are trained with 30 epochs, where the number of epochs is defined as the number of times the optimization algorithm goes through all training samples. The classifier and training algorithm are both implemented using PyTorch[48].

D. Data preparation

We generated independent training and testing trajectories for all seven test systems. Our training trajectories comprise fully equilibrated phases that were initiated from synthetic structures or self-assembled and annealed at various thermodynamic conditions. Equally spaced snapshots of the training trajectories were used to generate the training sets, from which we randomly drew the validation sets. The partitioning ratio of training and validation sets was fixed at 4:1, and each training set comprised a minimum of 20,000 local environments. The testing trajectories comprised self-assembly runs in which at least one phase transition was observed in each run.

We utilized the Hard Particle Monte Carlo (HPMC) and Molecular Dynamics (MD) modules of HOOMD-Blue[49] to simulate two-dimensional and three-dimensional convex hard particles. Interaction patchiness was implemented using the Just-In-Time (JIT) compilation module under HPMC. We simulated the equilibration, annealing and self-assembly processes within both the canonical (NVT) and isobaric-isothermal (NPT) ensemble. To simulate the hard particles using MD, we employed the anisotropic Weeks-Chandler-Andersen (AWCA) potential in HOOMD-Blue[50].

After we obtained the trajectories, we used the freud analysis package[51] to construct the neighbor list used to calculate per-particle input features. Other analysis functions of freud were also used, such as the radial distribution function (RDF) and Steinhardt order parameters[52]. The quaternion algebra is handled by rowan[44]. Snapshot images are rendered using Ovito[53]. Because our model is a supervised model, the model's ability to classify the phases is largely influenced by the labeling strategy during training. Here, our labeling strategy labels all particles within the same crystal the same. Additionally, we confirmed via calculation of the diffraction pattern and bond order diagram that our training trajectories contain mostly the reference local environments.

However, we note that one can never be 100% sure in any type of supervised ML model that all configurations that appear in the test trajectory will be in the training trajectories. Indeed, there is always a chance that small local fluctuations in particle arrangements will make our labeling less accurate, and thus will be missed by the model. In the systems used as test cases, we know a priori what equilibrium phases form and train on those phases. We then confirmed via the diffraction pattern and bond order diagram that our training dataset contains mostly the reference environments and that any spontaneous fluctuations that create locally the coordination of a different phase do not persist.

Each test case was simulated as follows:

- **Cubes:** For training trajectories, three independent equilibrated HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories equilibrated at packing fractions of 0.244 and 0.751, respectively. The testing trajectory was generated by compressing the system from packing fraction 0.244 to 0.864 using HPMC simulation and then equilibrating.

- **Patchy triangles:** For training trajectories, three independent equilibrated training HPMC trajectories were prepared, with one dense fluid trajectory and two kagome lattice trajectories. The kagome lattice is a two-dimensional crystal composed of corner-sharing triangles that has been discovered to be assembled by triblock Janus particles[54,55]. The fluid phase was equilibrated above the nucleation temperature ($k_BT \gtrapprox 0.105$), while the kagome lattice was initialized from a perfect kagome lattice with randomly placed guest parti-

cles and equilibrated at $k_BT = 0.105$. The testing trajectory was prepared by quenching from $k_BT = 0.3$ to 0.1 and equilibrating using HPMC.

- **Patchy prisms:** For training trajectories, three independent equilibrated HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories. The testing trajectory was generated by equilibrating an initially disordered system. Here, we use the system from Ref.[41] where the distance between the attractive patches and the face center of the prism is 0.8.

- **Hexagonal bipyramids with aspect ratio $\alpha = 1.28$:** For training trajectories, three independent equilibrated training MD trajectories were prepared, with one dense fluid trajectory, one plastic crystal trajectory, and one body-centered tetragonal (BCT) crystal trajectory[56]. The systems were equilibrated at packing fractions 0.464 and 0.569 for the dense fluid and plastic crystal, respectively.

- **Hexagonal bipyramids with aspect ratio $\alpha = 3.0$:** For training trajectories, three independent equilibrated training MD trajectories were prepared, with one dense fluid trajectory, one liquid crystal trajectory, and one triclinic crystal trajectory[56] equilibrated at packing fractions of 0.4, 0.51, and 0.661, respectively. The testing trajectory was prepared by quenching the system from a reduced pressure P^* of 0.5 to 10 and equilibrating it using MD.

- **Truncated tetrahedrons and octahedrons:** For truncated tetrahedrons, three independent HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories. The dense fluid trajectories were equilibrated at packing fractions 0.347 for truncated tetrahedrons and 0.524 for truncated octahedrons. For truncated tetrahedrons, the two training diamond crystal trajectories were equilibrated at packing fractions of 0.561. For the truncated octahedrons, the two training high-pressure lithium crystal trajectories were prepared by slowly annealing a self-assembled crystal and equilibrating at a packing fraction of 0.606. The HPMC testing trajectories were prepared by quenching the systems from a reduced pressure P^* from 0.5 to 10 for truncated tetrahedrons (using NPT) and a packing fraction from 0.14 to 0.62 for truncated octahedrons (using NVT), followed by equilibration.

III. RESULTS AND DISCUSSION

We test the performance of our MLP on seven different systems, beginning with the simplest case of hard cubes that self-assemble into a simple cubic lattice.

A. Test case 1: Simple cubic crystals assembled by hard cubes

As a demonstration of our method, we first show a simple classification test on a simple cubic structure self-assembled from the fluid phase of hard cubes upon an increase in packing fraction. As shown in Fig. 1, the cube exhibits 2-fold and 4-fold rotational symmetries along the θ- and ϕ-directions. Using Eq. 3 and Eq. 5, we can explicitly consider these symmetries in the data augmentation step and generate an appropriate training dataset that accounts for symmetry. From the three snapshots, each representing a different stage in the self-assembly process, in Fig. 4 (a), we see that as the packing fraction increases, the number of local environments classified as locally cubic also increases.

To further quantify the extent to which data augmentation using symmetry improves the MLP's classification abilities, we compare the classification results by the classifier trained with and without data augmentation. In Fig. 4 (b), the particle fraction is defined as the number of particles being classified as a certain local environment divided by the total number of particles in the system. There, we also plot the conventional Steinhardt order parameter Q_l[52] with $l = 4$ for comparison. The order parameter Q_l is defined by averaging $Q_{l,i}$ over each particle i in a system defined as:

$$Q_{l,i} = \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l} Q_{lm,i}Q^*_{lm,i}} \tag{13}$$

$$Q_{lm,i} = \frac{1}{N_b}\sum_{j=1}^{N_b} Y_{lm,\,ij}(\mathbf{r}_{ij}) \tag{14}$$

where $Y_{lm,\,ij}$ is the spherical harmonic calculated by the relative position of the target particle i and its neighbors j. For clarification, Q_l differs from the quaternion $\mathbf{Q}$ in equation 4.

Note that when using Q_4, we need to select a threshold value (manually selected as 0.51 in this case) based on human intuition or visualization, or we can apply another ML method to determine this threshold, such as support vector machine (SVM), which maximizes the margin. While in the MLP classifier, the threshold is determined solely by the MLP. It can be clearly seen that without data augmentation based on symmetry, misclassification occurs primarily in the fluid phase. In Fig. 4 (c), we see from the convergence behavior of the loss values that the augmented dataset generally converges faster and better.

To ensure that the augmented data provides sufficient information for the model to recognize the difference between different local environments and predict all possible local fluctuations as the same label, a sufficiently large number of neighbors must be included to build each training dataset. If the number of neighbors is not large enough, a clear drop of training and validation accuracy is seen, as depicted in Fig. 4 (e).

Fig. 4 (d) and (e) shows the results of accuracy tests performed on two other aspects. First, in Fig. 4(d), we observe that accuracy increases with the number of training epochs. We observe that the accuracy converges at around eight epochs and persists without severe overfitting until 32

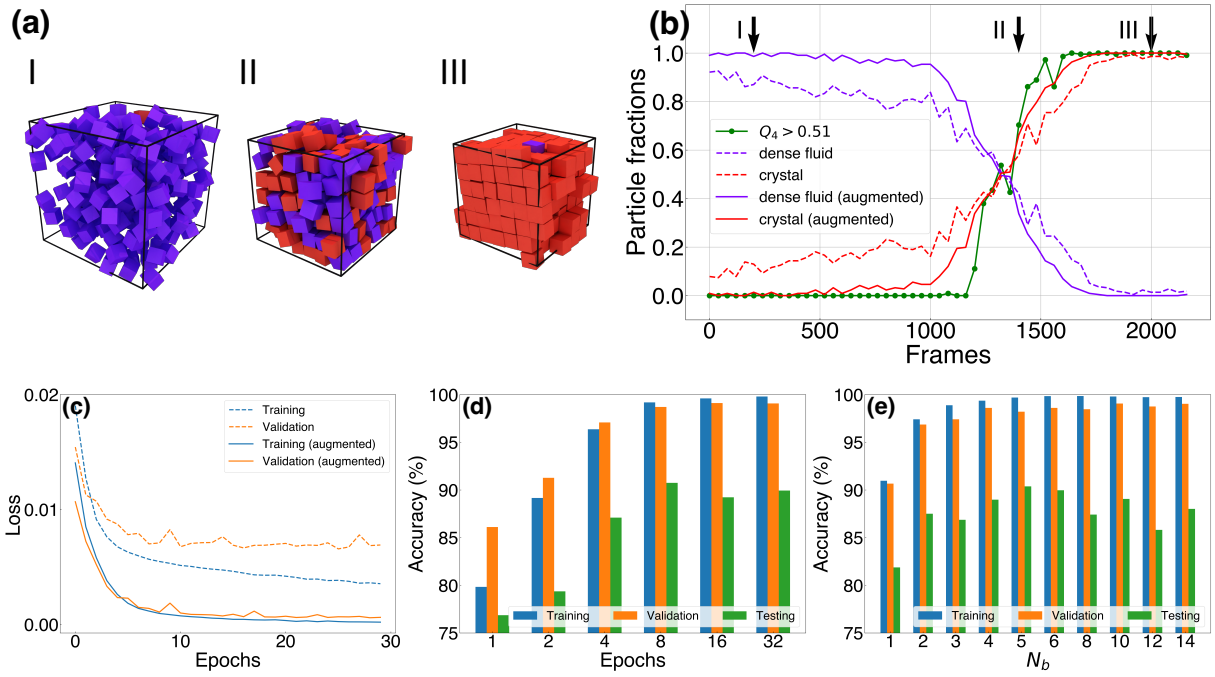

FIG. 4. **Simple cubic lattice consisted of cubes**. Summary of the MLP classifier's classification results, training details, and accuracy tests, in which we use the simple cubic system as a test case. **(a)** The MLP classifier's classification results on three different snapshots. **(b)** The MLP classifier's classification results on the self-assembly trajectory and classification, compared to the Steinhardt order parameter Q_4 (dotted line) calculated as ground truth. For visualization purposes, solid lines represent the MLP classifier trained on the data accounting for symmetry, while dashed lines are for the MLP classifier trained on the data without accounting for symmetry. Annotations I, II, and III indicate the three corresponding snapshots in (a) for the classifier trained on augmented data. **(c)** Learning curve of the MLP classifier used in (a) with (solid line) and without (dashed line) data augmentation **(d)** MLP classification accuracy plotted versus increasing number of training epochs **(e)** MLP classification accuracy plotted versus increasing number of neighbors N_b in the local particle neighborhood. In (a)-(d), we used fixed $N_b = 6$. In (a)-(c) and (e), we used 30 training epochs.

epochs. Second, in Fig. 4 (e), we show the change in accuracy by increasing N_b. The accuracy converges after $N_b = 5$, roughly the number of first nearest neighbors in a cubic lattice. This provides an excellent initial guess for N_b in preparing the training set of our classifier. Furthermore, for the test accuracy, which is defined with respect to the Q_l, 85-90% of the time our model will classify a particle the same as the classification using Q_l. This level of accuracy is sufficient to detect phase transitions also detected by Q_l. Importantly, our model (i) will also detect phase transitions in known systems where Steinhardt OPs (which contain no orientation information) and other commonly used OPs fail, and (ii) process particles' raw positions and orientations with a set of linear operations while Steinhardt OPs use non-linear functions such as spherical harmonics. This linearity in OPs will be important when one would like to apply them to various biased simulations, where gradients of OPs are required. We have also included training and testing time in the Supplementary Information. These tests show that our classifier is robust with appropriate augmentation, training epochs, and N_b. In the next section, we will look at more complicated crystals formed by hard particle shapes with different symmetries.

B. Test cases 2 and 3: Self-assembly of 2D and 3D patchy particles

For this second test case, we employ our classifier to classify 2D and 3D systems of patchy particles. In the 2D case depicted in Fig. 5 (a), each particle is a rigid equilateral triangle (Fig. 1 (b)). The patchiness is realized by decorating each particle with a Kern-Frenkel attractive patch[57] at each of the three vertices. Additionally, we apply three repulsive patches centered on each of the particle's edges to negatively design against undesirable phases. The guest particles inside the kagome lattice make finding a reliable order parameter that distinguishes the guest particles from non-guest kagome particles necessary. Our simulation results show that during self-assembly three distinct local environments emerge corresponding to a fluid-like, guest particle, and kagome lattice environments .

We demonstrate the classification result of our MLP classifier on the test case assembly pathway of the kagome lattice in Figure 5 (b), where the light green color indicates the local environment is classified as a guest particle. It can be seen that the system nucleates and forms a kagome lattice cluster within the fluid phase. Inside the kagome lattice cluster, several guest triangles are enclosed by six surrounding triangles. As the assembly simulation proceeds, we observe the

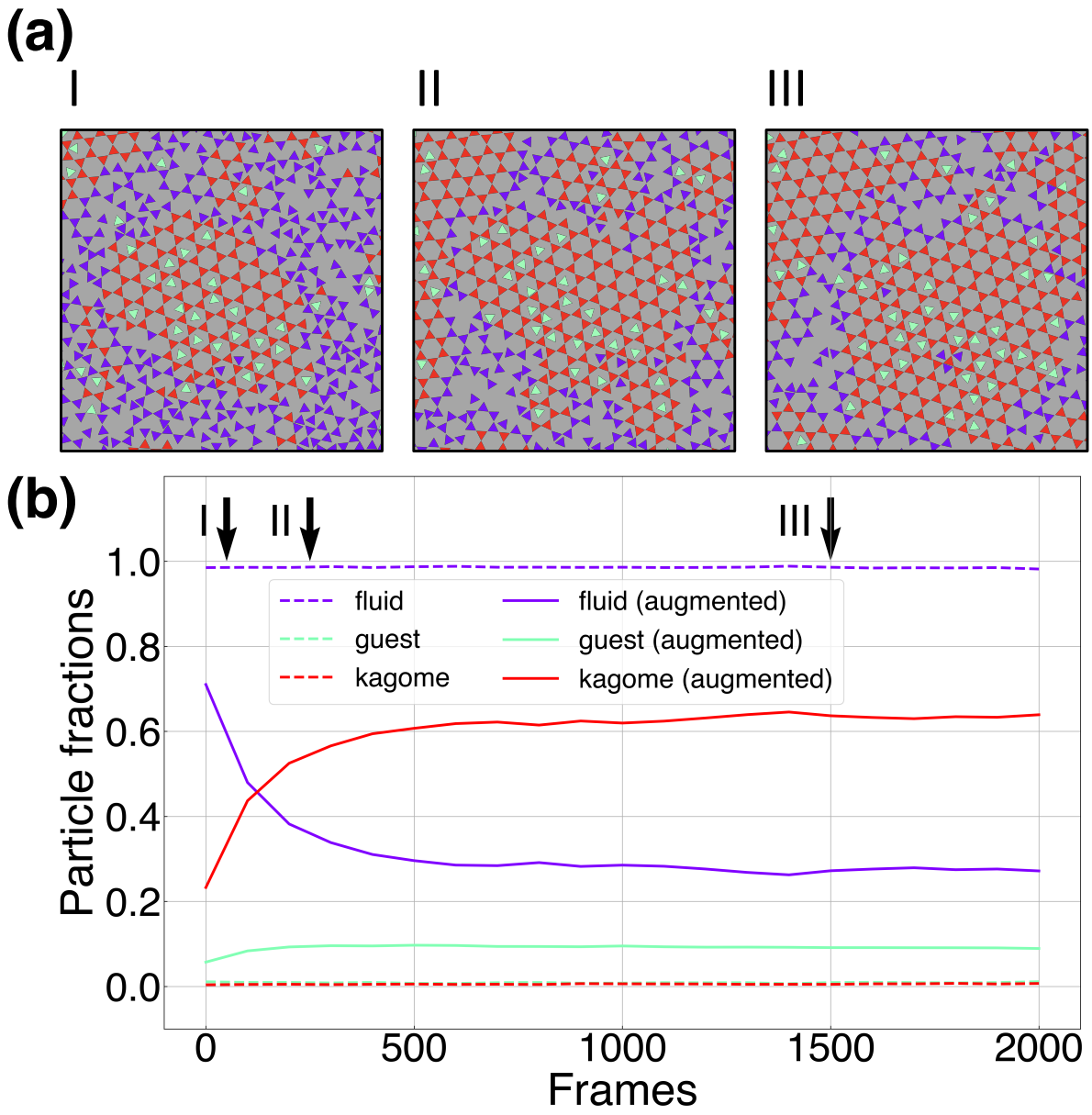

FIG. 5. **Kagome lattice of patchy triangles** Summary of the MLP classifier's classification results on the test assembly trajectory. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III correspond to the three snapshots in (a) for the classifier trained on augmented data.

coalescence of multiple small clusters into a single crystallite. Once the majority of particles in the system are in kagome lattice phase, the number of guest particles remains unchanged, which is also captured by the MLP classifier. It should be noted that some guest particles are misclassified as belonging to the fluid phase when they are too close to the surrounding particles within the kagome lattice. On the other hand, without data augmentation, the classifier only discovered fluid phase. This inability to distinguish local environments without data augmentation is consistent with our observation in hard cubes that when the crystal phase forms, the MLP classifier without data augmentation tends to underestimate the local environments of the ordered phases.

The second test case is the 3-dimensional dimer diamond self-assembled from the fluid phase of patchy triangular prisms[41] (See Fig. 1 (c)). In this test example, the triangular prisms pair up and arrange the pairs into the dimer-diamond structure. Since we treat each triangular prism as a particle, it is insufficient for the MLP classifier to classify only the diamond structure. It is crucial to recognize the pairing motif to classify the crystal phase.

In Fig. 6 (b), we show the trajectories of particle fractions

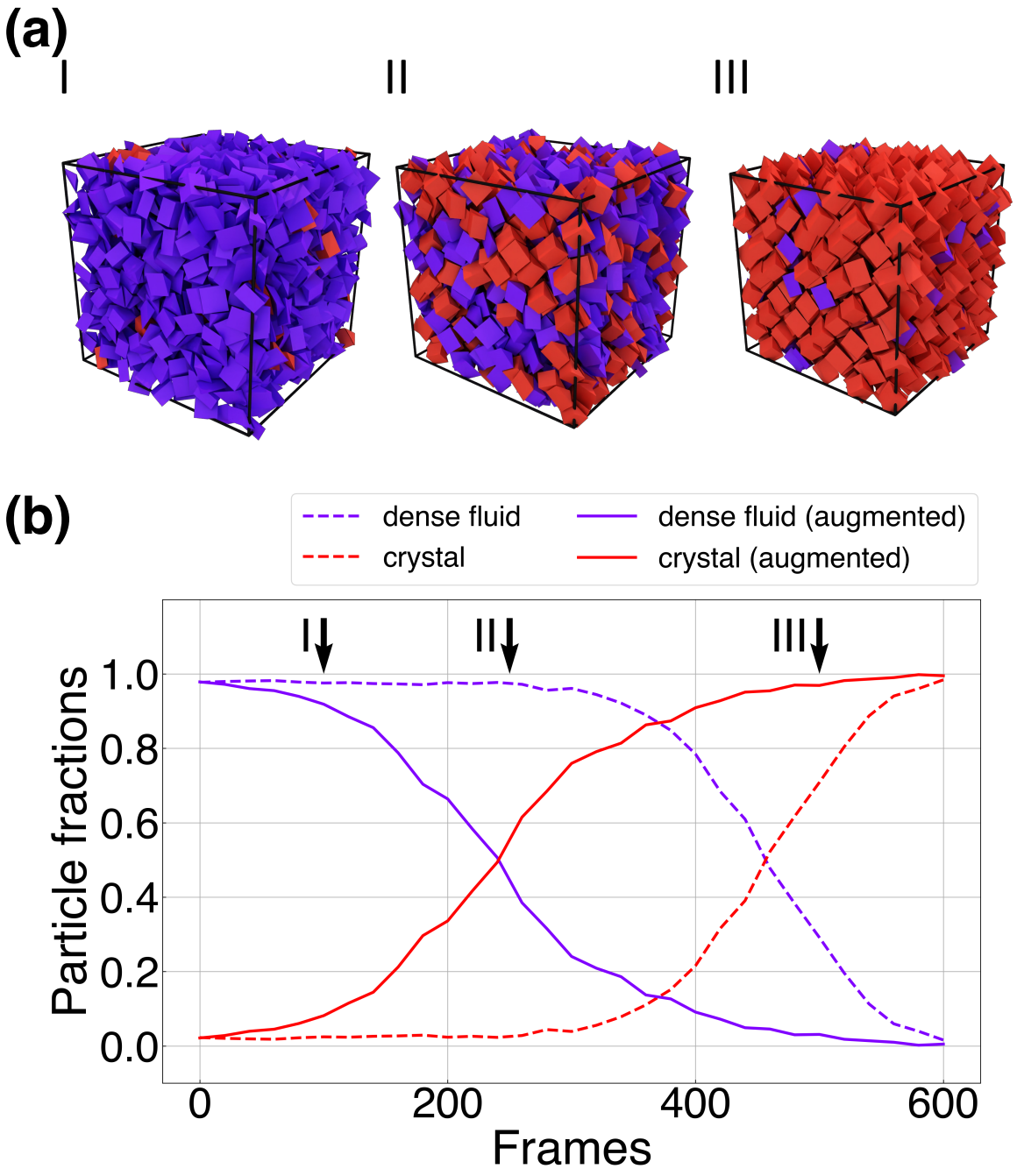

FIG. 6. **Dimer-diamond phase consisted of anti-aligned patchy triangular prism**[41] The MLP classifier's classification results on the testing self-assembly trajectory, in which we identify the crystallization dimer-diamond structure. Each Wyckoff site comprises two anti-aligned patchy prisms forming a gyrobifastigium. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the whole trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotates I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

identified by the classifier trained with or without data augmentation. While the initial and final particle fractions are now the same, the classifier trained with or without data augmentation exhibits different transition behaviors: the non-augmented classifier identifies half of the crystal particles much later than the augmented classifier.

C. Test cases 4 and 5: self-assembly of prolate hexagonal bipyramids

In the previous section, we showed two different patchy particle systems for which distinguishing the different local environments during assembly is crucial for observing that, in both 2D and 3D, these patchy shapes follow similar assembly pathways. In this section, we focus on two different assembly pathways using geometrically similar building blocks. Both crystals are self-assembled entropically by hard hexagonal bipyramids, but the bipyramid's aspect ratio ($\alpha = 1.28$ vs. 3.0), defined as the ratio of the particle's height h to its circumcircle diameter of base d, i.e., $\alpha = h/d$, greatly influences the intermediate and final products. For $\alpha = 3.0$ orientational order develops prior to translational order, while for $\alpha = 1.28$ we observe the opposite. According to a recent study[56], the self-assembly pathway from a fluid to the final crystal under slow compression involves an intermediate phase – a plastic crystal BCT phase for $\alpha = 1.28$ and a liquid crystal phase for $\alpha = 3.0$. The final products are BCT and triclinic phases for $\alpha = 1.28$ and $\alpha = 3.0$, respectively.

Because only the aspect ratio, and not symmetry, is different for the two shapes, we prepared only one classifier trained on six different synthetically prepared phases of hexagonal bipyramids for the two aspect ratios. We labeled the disordered phase of both hexagonal bipyramid systems as the same dense fluid phase. Because we consider only the particles' relative positions and quaternions as input, there is no difference in the form of the feature vectors except that they describe different local environments.

We first show the classification results of the MLP classifier on the test case trajectory of hexagonal bipyramids with $\alpha = 3.0$ in Fig. 7 (a) and (b). The dense fluid, liquid crystalline, and triclinic crystal phases are labeled purple, light blue, and light green, respectively. We also calculated the RDF for the three snapshots of Fig. 7 (a). In Fig. 7 (b), our MLP classifier reveals that the system starts to transform into the liquid crystalline phase immediately, followed by slow growth of the triclinic phase. This can also be seen in snapshots I, II, and III of Fig 7 (a). By comparing snapshots I and II, it is evident that the first transition involves only the orientational, and not yet the translational, ordering of particles, leading to a small difference between the two RDFs. Only after the liquid crystalline phase is sufficiently developed does the system order translationally to produce the triclinic crystal; this subsequent behavior is supported by snapshots II and III, as well as their RDFs.

Furthermore, our MLP classifier trained on the augmented data for the hexagonal bipyramids with $\alpha = 1.28$ was used to detect a similar two-step transition, and the outcomes are illustrated in Fig. 7 (c) and (d). The plastic crystal and BCT phases are indicated by orange and red colors, respectively. In this instance, the translational ordering of the particles occurs before the orientational ordering, which corresponds to a transition from the dense fluid phase to a plastic crystal phase, followed by a transition to the final BCT phase. It is worth noting that both snapshots IV and V of Fig. 7 (c) possess nearly identical RDFs, meaning the local density and structure are very similar. Despite this, our MLP classifier captures a significant difference in the particle fraction corresponding to dense fluid and plastic crystal. Without data augmentation, the MLP classifier cannot identify particle fractions that match our observed snapshots. In particular, the non-augmented MLP classifier underestimates the particle fractions of the final crystalline phases in both cases. Since the final crystalline phases have both translational and orientational ordering, the MLP classifier performs better with augmented data.

D. Test cases 6 and 7: Self-assembly of truncated polyhedra

As a final test of our classifier, we consider systems that are particularly challenging to identify using order parameters. As an example, we consider self-assembled systems of truncated tetrahedrons. Damasceno et al.[58] showed that tetrahedrons with varying degrees of truncation self-assemble into a wide range of complex crystal structures, including diamond structures and high-pressure lithium phases. Here, we investigate two different truncated tetrahedron systems with intermediate and large amounts of truncation. Because the tetrahedron gradually transforms to an octahedron with increasing vertex truncation, we refer to the system with intermediate truncation as the truncated tetrahedron and the system with high truncation as the truncated octahedron, to avoid confusion. Our simulations show results consistent with those of Damasceno et al. When we compressed the systems to a packing fraction between 0.5 and 0.6, they self-assembled into cubic diamond structures and high-pressure lithium phases for truncated tetrahedrons and truncated octahedrons, respectively.

In Fig. 8 (a) and (b), we demonstrate the classification results of the MLP classifier for the truncated tetrahedrons. In Fig. 8 (b), during the early simulation stage, the classifier trained on augmented data identifies an abrupt increase in the number of particles classified as belonging to a cubic diamond local environment. Subsequently, the diamond structure grows rapidly and remains stable. However, the classifier trained on the data without augmentation reports a slow growth of the diamond structure and fails to identify the phase transition.

In Fig. 9 (b), our MLP classifier trained on augmented data first detected a progressively increasing high-pressure lithium-like local environment before the simulation reached frame12000. After that, our classifier identified that the high-pressure lithium phase reached the critical nucleus size, followed by rapid crystal growth. Thus the MLP classifier discovered a homogeneous nucleation process followed by crystal growth. This nucleation process can also be seen in Fig. 9 (a). From the classification-based coloring in snapshot I, we observed many small sub-critical crystal nuclei form and dissolve in the early stages. In snapshot II, shortly before the rapid growth of the crystal, the MLP classifier identified a noticeable amount of crystal-like local environment in the lower part of the box, which we expect to be the critical nucleus. At the end of the simulation, we can see that the crystal has stabilized in the lower right corner of the box in snapshot III,

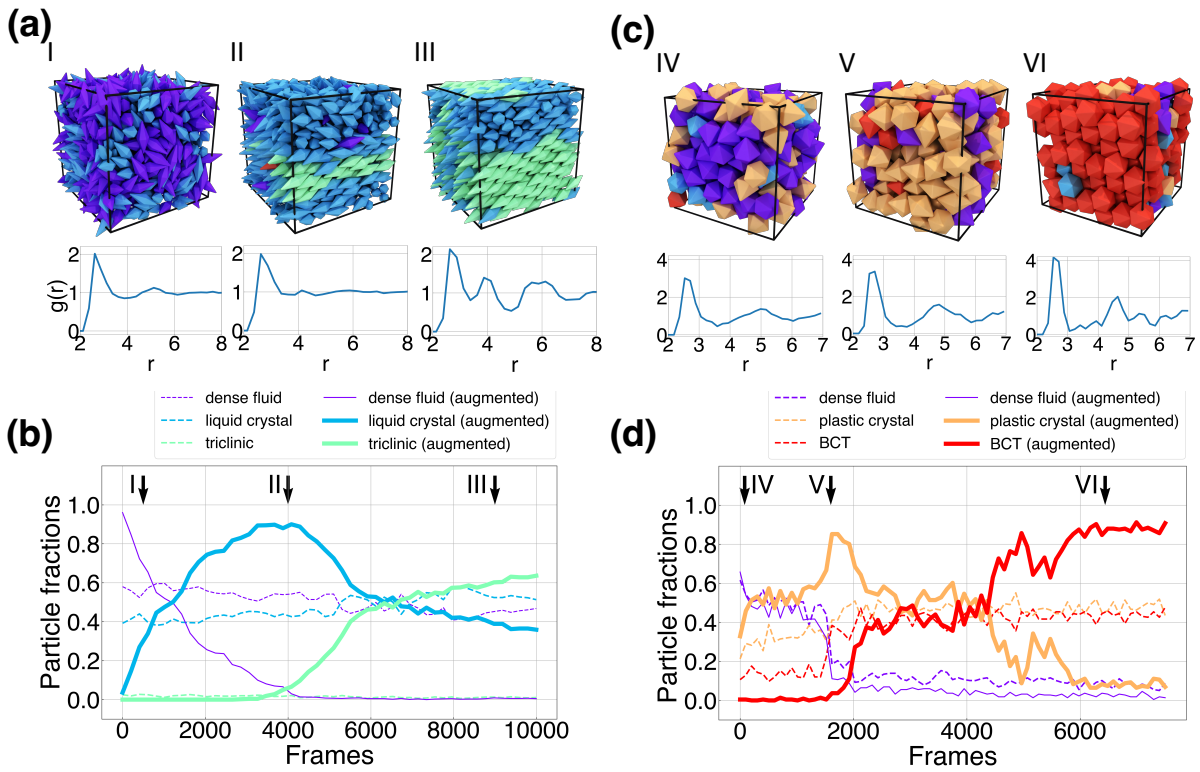

FIG. 7. **Hexagonal bipyramid systems** The MLP classifier's classification results on the assembly trajectories for two systems of hexagonal bipyramids. Both systems exhibit a two-step transition when crystallizing from an initial disordered fluid phase. **(a)** and **(c)** The MLP classifier's classification on six snapshots of the hexagonal bipyramid systems with (a) $\alpha = 3.0$ and (b) $\alpha = 1.28$, as well as the corresponding radial distribution functions at the bottom. **(b)** and **(d)** The MLP classifier's classification results on the entire trajectories. The annotations I, II, III, and IV, V, VI correspond to the snapshots in (a) and (b), respectively, for the classifier trained on augmented data. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively.

where it is in coexistence with a dense fluid phase.

For comparison, we demonstrate the classifier's classification ability without data augmentation. As expected, the MLP classifier trained on the data without augmentation failed to recognize the rapid growth of the crystal. The inability of the classifier to identify the formation of complex crystals is similar to the case of truncated tetrahedrons, as shown in Fig. 8 (a) and (b), and thus for both test cases we see that data augmentation highly improves the performance of the MLP classifier. We can rationalize this performance difference between training on augmented vs. non-augmented data by examining the information contributed by each component of the particle feature vector. Since we started both simulations from very dense fluid phases, there are no significant changes in densities that are encoded in $r_{ij}^{(au)}$ during the formation of crystals. Therefore, the information provided by $r_{ij}^{(au)}$ is insufficient to recognize the formation of crystals. As a result, whether the MLP classifier can correctly learn to identify the formation of crystals largely depends on whether θ_{ij}, ϕ_{ij} and $\mathbf{q}_{ij}$ are augmented.

CONCLUSIONS

To summarize, we have developed a simple, yet powerful, physics-agnostic local environment classifier specifically designed for systems of particle shapes utilizing a multilayer perceptron. As demonstrated in this paper, our method is applicable to a range of enthalpically and entropically patchy particle systems. Importantly, our MLP classifier does not need conventional roto-translation invariant symmetry functions to transform per-particle quantities to input descriptors. Instead, it directly takes particle positions and orientations as input features, complemented by shape-encoded data augmentation. To demonstrate robustness and flexibility, our classifier's performance was assessed on a variety of self-assembling systems, including hard cubes, 2D and 3D patchy shapes, hexagonal bipyramids with two different aspect ratios, and two different truncated shapes. The data augmentation method we used is straightforward and easily transferable to other systems involving particle orientations, such as molecular or coarse-grained systems. As a result, our approach should be useful in classifying different polymorphs formed by molecules by properly defining orientation through quaternions. Furthermore, due to the simplicity and the promising classification performance, our method may be applied

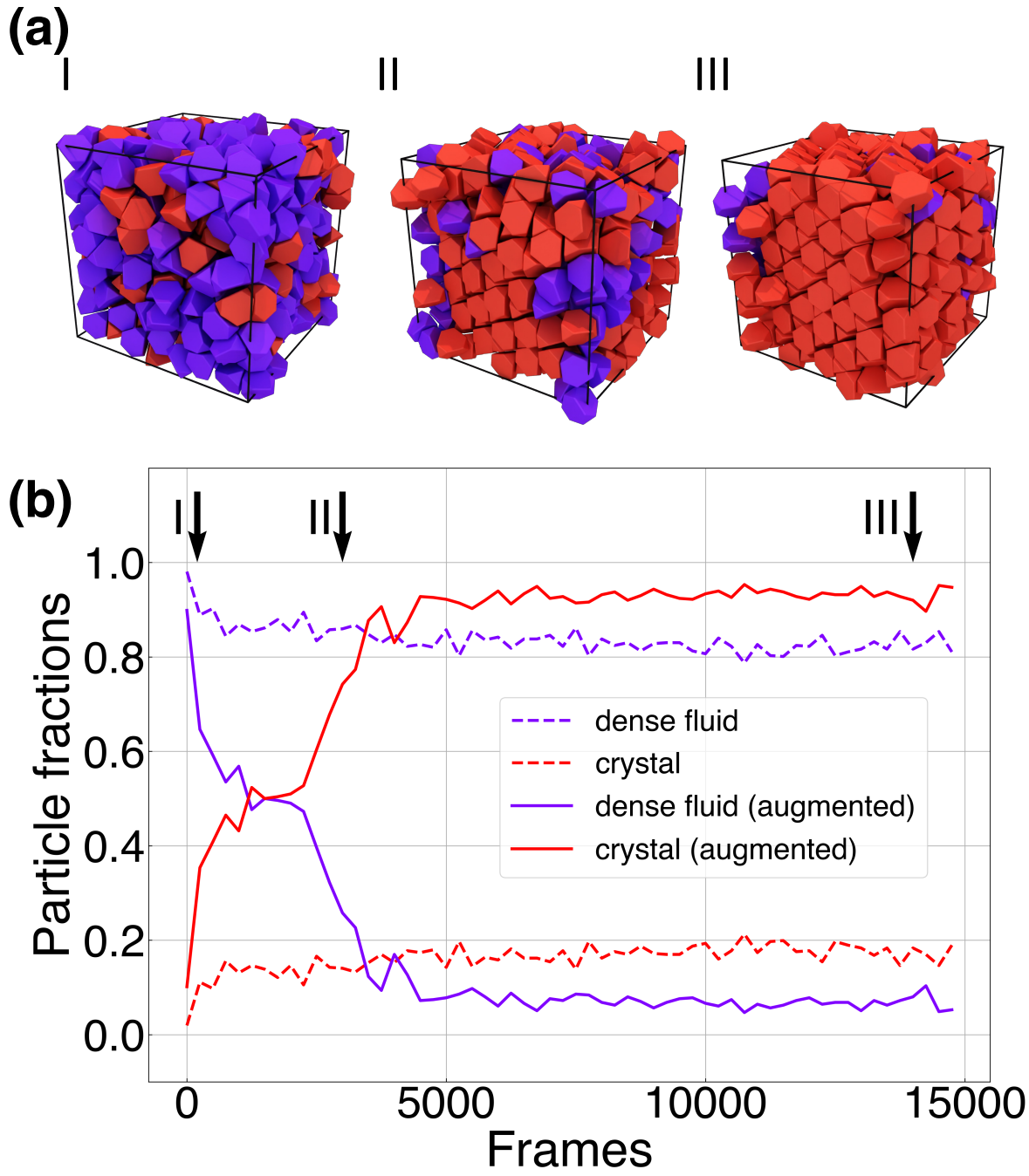

FIG. 8. **Diamond structure of truncated tetrahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the diamond structure from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on snapshots taken at three points along the trajectory. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

to study nucleation pathways of colloidal systems with biased simulation techniques, such as umbrella sampling and metadynamics[36]. Such applications will be explored in future works.

We emphasize that we do not intend for our model to be used on an unknown dataset. Likewise, Steinhardt and other order parameters (OP) commonly used in soft matter studies are not used on unknown datasets. Rather, they are used to automate the detection of phase transitions in systems where the initial and final phases are known. In our paper, we are attempting to show that our supervised ML model is able to detect phase transitions in known systems nearly as well as

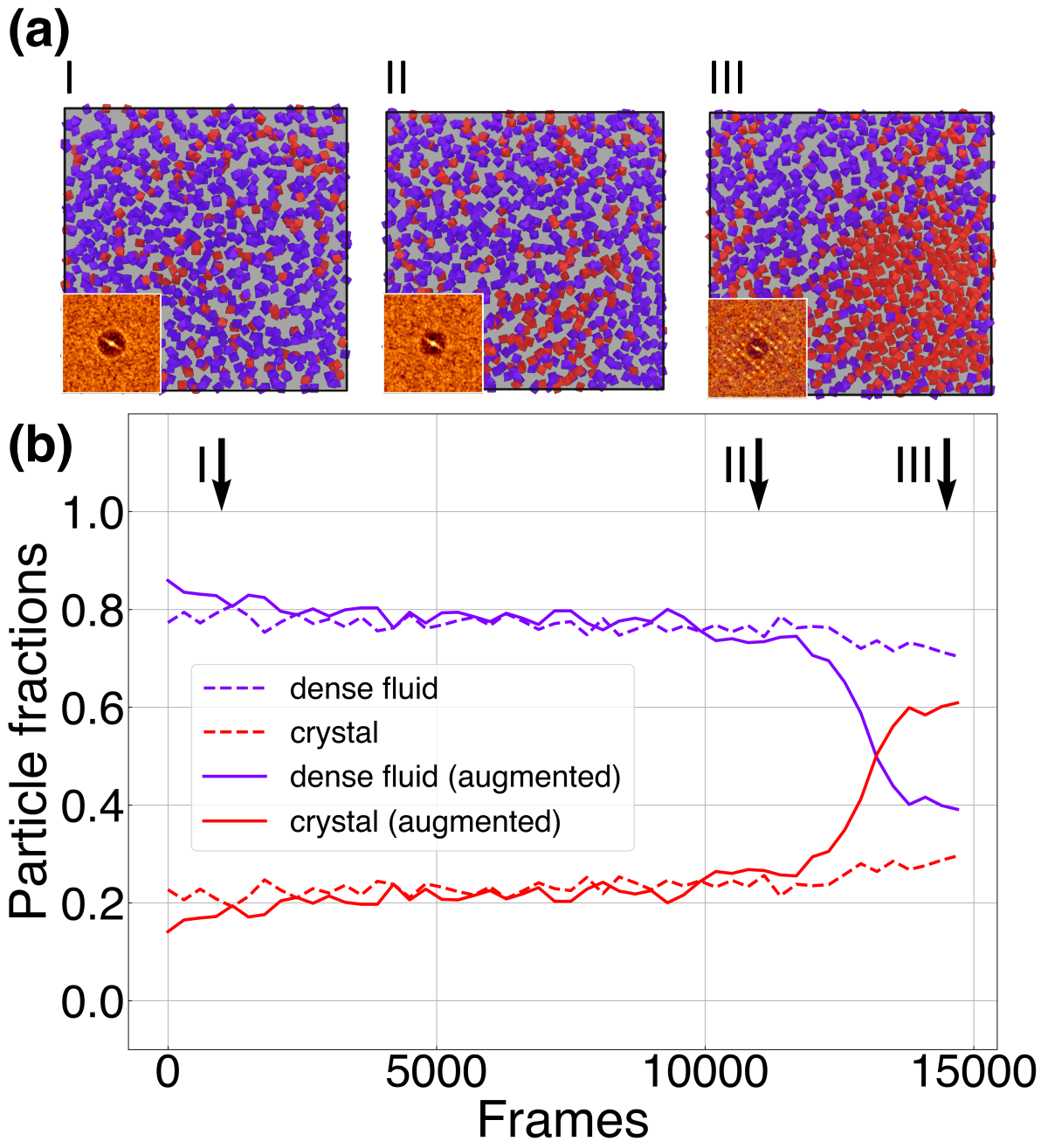

FIG. 9. **High-pressure lithium phase of truncated octahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the high-pressure lithium phase from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

those order parameters.

Again, as with any supervised ML model, the model will fail when applied to an unknown dataset that contains information not contained in the training set. Our supervised model is best used to quickly analyze many (often, hundreds or thousands of) trajectories forming crystals that they are known a priori to form (for example, to sample many statistically similar pathways, a common workflow in studies of nucleation and crystallization). Our ML model facilitates this workflow significantly.

The generalizability of our approach to unknown datasets can, in principle, be extended by training over all possible intermediate (e.g. metastable) and equilibrium crystal structures. Such an undertaking is beyond the scope of this paper

but would be worth doing in the future.

IV. CONFLICTS OF INTEREST

There are no conflicts to declare.

V. SUPPLEMENTARY MATERIAL

See supplementary material for the additional computational time data.

VI. ACKNOWLEDGEMENTS

This research was supported by a CDS&E grant from the National Science Foundation (NSF), Division of Materials Research Award No. DMR 2302470. This work used resources from the Extreme Science and Engineering Discovery Environment (XSEDE), supported by the same aforementioned NSF grant. Computational resources and services were supported by Advanced Research Computing at the University of Michigan, Ann Arbor. The authors would like to thank Yuan Zhou for providing data on patchy particle systems and for helpful discussion on the symmetry of shapes; Michael Engel for providing the crystal data for truncated octahedrons; Thi Vo's help in setting up the MD simulations; and Brandon Butler, Ziyue Zou, and Yihang Wang for their helpful suggestions.

DATA AVAILABILITY

The data that supports the findings of this study as well as information showcasing the data preparation, training and testing of our model are available online on GitHub at `https://github.com/shihkual/mlp_crystal_classifier.git` and at the Deep Blue Data at `https://doi.org/10.7302/w13t-2177` offered by the University of Michigan Library.

[1]M. A. Boles, M. Engel, and D. V. Talapin, "Self-assembly of colloidal nanocrystals: From intricate structures to functional materials," Chemical reviews **116**, 11220–11289 (2016).

[2]G. M. Whitesides and M. Boncheva, "Beyond molecules: Self-assembly of mesoscopic and macroscopic components," Proceedings of the National Academy of Sciences **99**, 4769–4774 (2002).

[3]Z. Li, M. Liu, L. Wang, J. Nangreave, H. Yan, and Y. Liu, "Molecular behavior of dna origami in higher-order self-assembly," Journal of the American Chemical Society **132**, 13545–13552 (2010).

[4]T. Tørring, N. V. Voigt, J. Nangreave, H. Yan, and K. V. Gothelf, "Dna origami: a quantum leap for self-assembly of complex structures," Chemical Society Reviews **40**, 5636–5646 (2011).

[5]J. Wang, X. Zhao, J. Li, X. Kuang, Y. Fan, G. Wei, and Z. Su, "Electrostatic assembly of peptide nanofiber–biomimetic silver nanowires onto graphene for electrochemical sensors," ACS Macro Letters **3**, 529–533 (2014).

[6]Q. Li, Y. Jia, L. Dai, Y. Yang, and J. Li, "Controlled rod nanostructured assembly of diphenylalanine and their optical waveguide properties," ACS nano **9**, 2689–2695 (2015).

[7]S. Palchoudhury, Z. Zhou, K. Ramasamy, F. Okirie, P. E. Prevelige, and A. Gupta, "Self-assembly of p22 protein cages with polyamidoamine dendrimer and inorganic nanoparticles," Journal of Materials Research **32**, 465–472 (2017).

[8]M. Uchida, K. McCoy, M. Fukuto, L. Yang, H. Yoshimura, H. M. Miettinen, B. LaFrance, D. P. Patterson, B. Schwarz, J. A. Karty, *et al.*, "Modular self-assembly of protein cage lattices for multistep catalysis," ACS nano **12**, 942–953 (2018).

[9]K. McCoy, M. Uchida, B. Lee, and T. Douglas, "Templated assembly of a functional ordered protein macromolecular framework from p22 virus-like particles," ACS nano **12**, 3541–3550 (2018).

[10]L. Wang, C. Gong, X. Yuan, and G. Wei, "Controlling the self-assembly of biomolecules into functional nanomaterials through internal interactions and external stimulations: A review," Nanomaterials **9**, 285 (2019).

[11]J. Wang and A. Ferguson, "Nonlinear machine learning in simulations of soft and biological materials," Molecular Simulation **44**, 1090–1107 (2018).

[12]M. Spellings and S. C. Glotzer, "Machine learning for crystal identification and discovery," AIChE Journal **64**, 2198–2206 (2018).

[13]C. S. Adorf, T. C. Moore, Y. J. Melle, and S. C. Glotzer, "Analysis of self-assembly pathways with unsupervised machine learning algorithms," The Journal of Physical Chemistry B **124**, 69–78 (2019).

[14]R. Van Damme, G. M. Coli, R. Van Roij, and M. Dijkstra, "Classifying crystals of rounded tetrahedra and determining their order parameters using dimensionality reduction," ACS nano **14**, 15144–15153 (2020).

[15]G. M. Coli and M. Dijkstra, "An artificial neural network reveals the nucleation mechanism of a binary colloidal ab13 crystal," ACS nano **15**, 4335–4346 (2021).

[16]E. G. Teich, G. van Anders, and S. C. Glotzer, "Identity crisis in alchemical space drives the entropic colloidal glass transition," Nature communications **10**, 64 (2019).

[17]S. Lee, E. G. Teich, M. Engel, and S. C. Glotzer, "Entropic colloidal crystallization pathways via fluid–fluid transitions and multidimensional prenucleation motifs," Proceedings of the National Academy of Sciences **116**, 14843–14851 (2019).

[18]J. E. Carpenter and M. Grünwald, "Pre-nucleation clusters predict crystal structures in models of chiral molecules," Journal of the American Chemical Society **143**, 21580–21593 (2021).

[19]W. Mickel, S. C. Kapfer, G. E. Schröder-Turk, and K. Mecke, "Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter," The Journal of chemical physics **138**, 044501 (2013).

[20]W. Lechner and C. Dellago, "Accurate determination of crystal structures based on averaged local bond order parameters," The Journal of chemical physics **129**, 114707 (2008).

[21]R. Mao, J. O'Leary, A. Mesbah, and J. Mittal, "A deep learning framework discovers compositional order and self-assembly pathways in binary colloidal mixtures," JACS Au **2**, 1818–1828 (2022).

[22]A. Stukowski, "Structure identification methods for atomistic simulations of crystalline materials," Modelling and Simulation in Materials Science and Engineering **20**, 045021 (2012).

[23]P. M. Larsen, S. Schmidt, and J. Schiøtz, "Robust structural identification via polyhedral template matching," Modelling and Simulation in Materials Science and Engineering **24**, 055007 (2016).

[24]A. W. Long and A. L. Ferguson, "Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms," The Journal of Physical Chemistry B **118**, 4228–4244 (2014).

[25]W. F. Reinhart, A. W. Long, M. P. Howard, A. L. Ferguson, and A. Z. Panagiotopoulos, "Machine learning for autonomous crystal structure identification," Soft Matter **13**, 4733–4745 (2017).

[26]A. L. Ferguson, "Machine learning and data science in soft materials engineering," Journal of Physics: Condensed Matter **30**, 043002 (2017).

[27]R. S. DeFever, C. Targonski, S. W. Hall, M. C. Smith, and S. Sarupria, "A generalized deep learning approach for local structure identification in molecular simulations," Chemical science **10**, 7503–7515 (2019).

[28]W. F. Reinhart, "Unsupervised learning of atomic environments from simple features," Computational Materials Science **196**, 110511 (2021).

[29]Q. Kim, J.-H. Ko, S. Kim, and W. Jhe, "Gcicenet: a graph convolutional network for accurate classification of water phases," Physical Chemistry Chemical Physics **22**, 26340–26350 (2020).

[30] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, "Insightful classification of crystal structures using deep learning," Nature communications **9**, 2775 (2018).
[31] N. Charest, M. Tro, M. T. Bowers, and J.-E. Shea, "Latent models of molecular dynamics data: Automatic order parameter generation for peptide fibrillization," The Journal of Physical Chemistry B **124**, 8012–8022 (2020).
[32] A. Chakraborty and R. Sharma, "A deep crystal structure identification system for x-ray diffraction patterns," The Visual Computer **38**, 1275–1282 (2022).
[33] A. F. Lizano-Villalobos and X. Tang, "Convolutional neural network-based colloidal self-assembly state classification," Soft Matter (2023).
[34] P. Geiger and C. Dellago, "Neural networks for local structure detection in polymorphic systems," The Journal of chemical physics **139**, 164105 (2013).
[35] Z. Zou and P. Tiwary, "Enhanced sampling of crystal nucleation with graph representation learnt variables," arXiv preprint arXiv:2310.07927 (2023).
[36] F. Dietrich, X. R. Advincula, G. Gobbo, M. Bellucci, and M. Salvalaglio, "Machine learning nucleation collective variables with graph neural networks," (2023).
[37] S. Banik, D. Dhabal, H. Chan, S. Manna, M. Cherukara, V. Molinero, and S. K. Sankaranarayanan, "Cegann: Crystal edge graph attention neural network for multiscale classification of materials environment," npj Computational Materials **9**, 23 (2023).
[38] V. G. Satorras, E. Hoogeboom, and M. Welling, "E (n) equivariant graph neural networks," in *International conference on machine learning* (PMLR, 2021) pp. 9323–9332.
[39] S.-K. Lee, "Github repository," `https://github.com/shihkual/mlp_crystal_classifier.git` (2024).
[40] S.-K. Lee, S.-T. Tsai, and S. C. Glotzer, "Deep blue data," `https://doi.org/10.7302/w13t-2177` (2024).
[41] Y. Zhou, R. K. Cersonsky, and S. C. Glotzer, "A route to hierarchical assembly of colloidal diamond," Soft Matter **18**, 304–311 (2022).
[42] A. S. Keys, C. R. Iacovella, and S. C. Glotzer, "Characterizing complex particle morphologies through shape matching: Descriptors, applications, and algorithms," Journal of Computational Physics **230**, 6438–6463 (2011).
[43] A. Haji-Akbari and S. C. Glotzer, "Strong orientational coordinates and orientational order parameters for symmetric objects," Journal of Physics A: Mathematical and Theoretical **48**, 485201 (2015).
[44] V. Ramasubramani and S. C. Glotzer, "rowan: A python package for working with quaternions," Journal of Open Source Software **3**, 787 (2018).
[45] M. Finzi, M. Welling, and A. G. Wilson, "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups," in *International Conference on Machine Learning* (PMLR, 2021) pp. 3318–3328.
[46] D. Chen, J. Tachella, and M. E. Davies, "Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022) pp. 5647–5656.
[47] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)* (San Diega, CA, USA, 2015).
[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019) pp. 8024–8035.
[49] J. A. Anderson, J. Glaser, and S. C. Glotzer, "Hoomd-blue: A python package for high-performance molecular dynamics and hard particle monte carlo simulations," Computational Materials Science **173**, 109363 (2020).
[50] V. Ramasubramani, T. Vo, J. A. Anderson, and S. C. Glotzer, "A mean-field approach to simulating anisotropic particles," The Journal of Chemical Physics **153**, 084106 (2020).
[51] V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, "freud: A software suite for high throughput analysis of particle simulation data," Computer Physics Communications **254**, 107275 (2020).
[52] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond-orientational order in liquids and glasses," Physical Review B **28**, 784 (1983).
[53] A. Stukowski, "Visualization and analysis of atomistic simulation data with ovito–the open visualization tool," Modelling and simulation in materials science and engineering **18**, 015012 (2009).
[54] Q. Chen, S. C. Bae, and S. Granick, "Directed self-assembly of a colloidal kagome lattice," Nature **469**, 381–384 (2011).
[55] L. Y. Rivera-Rivera, T. C. Moore, and S. C. Glotzer, "Inverse design of triblock janus spheres for self-assembly of complex structures in the crystallization slot via digital alchemy," Soft Matter **19**, 2726–2736 (2023).
[56] Y. Lim, S. Lee, and S. C. Glotzer, "Engineering the thermodynamic stability and metastability of mesophases of colloidal bipyramids through shape entropy," ACS nano (2023).
[57] N. Kern and D. Frenkel, "Fluid–fluid coexistence in colloidal systems with short-ranged strongly directional attraction," The Journal of chemical physics **118**, 9882–9889 (2003).
[58] P. F. Damasceno, M. Engel, and S. C. Glotzer, "Crystalline assemblies and densest packings of a family of truncated tetrahedra and the role of directional entropic forces," Acs Nano **6**, 609–614 (2012).