

Inverse Design of Crystals and Quasicrystals in a Non-Additive Binary Mixture of Hard Disks

Edwin A. Bedolla-Montiel,¹ Jochem T. Lange,¹ Alberto Pérez de Alba Ortiz,^{2,1} and Marjolein Dijkstra¹

¹*Soft Condensed Matter & Biophysics, Debye Institute for Nanomaterials Science, Utrecht University, Princetonplein 1, 3584 CC Utrecht, Netherlands.*

²*Computational Soft Matter Lab, Computational Chemistry Group and Computational Science Lab, van 't Hoff Institute for Molecular Science and Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, Netherlands.*

(*Electronic mail: m.dijkstra@uu.nl, e.a.bedollamontiel@uu.nl)

(Dated: 25 March 2024)

The development of new materials typically involves a process of trial and error, guided by insights from past experimental and theoretical findings. The inverse design approach for soft-matter systems has the potential to optimize specific physical parameters such as particle interactions, particle shape, or composition and packing fraction. This optimization aims to facilitate the spontaneous formation of specific target structures through self-assembly. In this study, we expand upon a recently introduced inverse design protocol for monodisperse systems to identify the required conditions and interactions for assembling crystal and quasicrystal phases within a binary mixture of two distinct species. This method utilizes an evolutionary algorithm to identify the optimal state point and interaction parameters, enabling the self-assembly of the desired structure. Additionally, we employ a convolutional neural network (CNN) that classifies different phases based on their diffraction patterns, serving as a fitness function for the desired structure. Using our protocol, we successfully inverse design two-dimensional crystalline structures, including a hexagonal lattice, and a dodecagonal quasicrystal, within a non-additive binary mixture of hard disks. Finally, we introduce a symmetry-based order parameter that leverages the encoded symmetry within the diffraction pattern. This order parameter circumvents the need for training a CNN, and is used as a fitness function to inverse design an octagonal quasicrystal.

I. INTRODUCTION

The self-assembly of colloidal particles is a pivotal mechanism for fabricating nanostructured materials. Colloidal systems, consisting of particles ranging from nanometer to micrometer sizes suspended in a fluid medium, inherently possess the ability to spontaneously organize themselves into structured arrangements due to interparticle forces and thermodynamic conditions. These systems show promise for potential applications due to their photonic,^{1,2} magnetic and electronic properties.³ However, the structure of the assembly of such materials depends on the building blocks, namely the interaction between the species composing the system, and the thermodynamic conditions, such as temperature, pressure, density, or composition. Understanding the relationship between these building blocks, thermodynamic state, and self-assembled structures is crucial for leveraging them for materials design. This relationship is fundamental, as the physical properties of each material are intrinsically intertwined with its structure.

In the forward design approach, a specific colloidal particle system is chosen as the foundational building blocks for a material with desired properties. Subsequently, the interaction parameters and thermodynamic conditions are systematically changed until achieving the desired material. The forward design approach can quickly become unfeasible, because the number of possible building block combinations and conditions needed to assemble the required structures is intractably large.

Recently, there has been a growing interest in developing frameworks for the inverse design of self-assembled structures and materials.⁴ The inverse design approach directly de-

termines the parameters and thermodynamic conditions necessary to attain a target structure with specific properties.⁵ Inverse design protocols have been successfully applied to the search of both crystalline^{6,7} and quasicrystalline structures.⁷⁻¹³ One category of methods involves adjusting intermolecular interactions to target the radial distribution function of the desired phase. The objective is to precisely match the two-body structural correlations using a maximum entropy optimization scheme.^{14,15}

Recently, machine learning methods have been used for the analysis, identification, and formation of soft-matter systems.⁴ Coli *et al.* developed an inverse design protocol¹⁰ employing supervised machine learning techniques. The authors used a convolutional neural network (CNN) as a classifier to differentiate structures based on their diffraction patterns. Subsequently, target structures are achieved by modifying the interaction parameters and thermodynamic state points through an evolutionary strategy optimization algorithm, which uses a CNN-based fitness. This exploration helps identifying the thermodynamic conditions and interactions necessary for stabilizing the desired structures. The methodology proposed by Coli *et al.* facilitates the self-assembly of quasicrystals, as the symmetry of the structure is encoded within the diffraction patterns. Recently, Lieu and Yoshinaga proposed an alternative framework in which reinforcement learning is used along with patchy particles to promote the self-assembly of dodecagonal quasicrystals.¹⁶ Using an optimization method tailored for reinforcement learning, they automatically tuned the cooling schedule such that a critical temperature is identified at which the quasicrystal is formed.

In this work, we expand upon the protocol of Coli *et al.*

to encompass non-additive binary mixtures of hard disks, a system more akin to the quasicrystals observed in the experiments on monolayers of inorganic nanoparticles.¹⁷ The phase behavior of a similar system, *i.e.* binary mixtures of hard spheres, has been explored by computer simulations.^{18,19} We employ a CNN to classify different structures using the diffraction patterns of one of the species within the mixture. Subsequently, we use an evolutionary strategy algorithm to optimize the thermodynamic conditions and interaction parameters, facilitating the self-assembly of binary crystals and quasicrystals. However, this protocol encounters severe challenges when the training data for the CNN lacks diffraction patterns of the target structure, hindering its effectiveness in inverse designing the desired system. To address this limitation, we introduce a novel symmetry-based order parameter. This order parameter eliminates the necessity for training a CNN and bypasses the requirement for diffraction patterns of the structures to be inverse-designed. We observe that this order parameter, which measures the order of symmetry of the diffraction pattern, can successfully be used as a fitness function in the evolutionary strategy algorithm to optimize the physical parameters for the self-assembly of the desired structures.

This paper is organized as follows. In Sec. II, we introduce our model for the binary mixture of hard disks. The inverse design protocol is presented in Sec. III. This protocol comprises simulations for sampling, both a CNN and a symmetry-based order parameter for fitness evaluation, and an evolutionary strategy for the optimization of the parameters. In Sec. IV, we present the main results, illustrating the self-assembly of crystalline and quasicrystalline structures via the CNN approach. Following this, we employ the symmetry-based order parameter to successfully inverse design an octagonal quasicrystal. We present our conclusions in Sec. V.

II. MODEL

In our inverse-design approach for self-assembly of binary mixtures, we aim to produce trajectories that start in a disordered, low-density phase and finish in the target crystalline or quasicrystalline phase. Producing such trajectories requires time propagation of particle motions, which we compute via molecular dynamics (MD) simulations.²⁰ We consider a two-dimensional non-additive binary mixture of hard disks of two different sizes. The large (L) species have a diameter σ_L and the small (S) species a diameter σ_S . To perform MD simulations of hard-disk systems, it is necessary to employ a continuous interaction potential. Using the Extended Law of Corresponding States as formulated by Noro and Frenkel,²¹ and assuming its applicability extends to repulsive potentials, Báez *et al.*²² argued that by using a re-parametrization of the intermolecular potential proposed by Jover *et al.*,²³ it becomes feasible to map the hard-core interaction onto a continuous potential. This approach yields accurate results, particularly when the continuous potential precisely reproduces the second virial coefficient of the true hard-particle potential. Such a re-parametrization of the continuous hard-core po-

tential $u_{\alpha\beta}(r)$ between species $\alpha = L, S$ and species $\beta = L, S$ reads

$$u_{\alpha\beta}(r) = \begin{cases} A \varepsilon \left[\left(\frac{\sigma_{\alpha\beta}}{r} \right)^\lambda - \left(\frac{\sigma_{\alpha\beta}}{r} \right)^{\lambda-1} \right] + \varepsilon & , r < \sigma_{\alpha\beta} B \\ 0 & , r \geq \sigma_{\alpha\beta} B \end{cases}, \quad (1)$$

with

$$A = \lambda \left(\frac{\lambda}{\lambda-1} \right)^{\lambda-1}; \quad B = \left(\frac{\lambda}{\lambda-1} \right), \quad (2)$$

where r denotes the center-of-mass distance between the two particles, $\sigma_{\alpha\beta}$ represents the hard-core diameter between species α and β , and ε denotes the effective strength of the interaction. We set the hard-core diameters $\sigma_{LL} = \sigma_L$, $\sigma_{SS} = \sigma_S$, and introduce a non-additivity parameter Δ for the contact distance between the large and small species

$$\sigma_{LS} = \frac{\sigma_S + \sigma_L}{2} (1 - \Delta). \quad (3)$$

A non-linear equation in terms of the reduced temperature $T^* = k_B T / \varepsilon$ with k_B Boltzmann's constant must be solved in order to determine the temperature at which the difference between the second virial coefficient of both interaction potentials becomes zero.

In this study, the parameter λ is held constant at $\lambda = 50$ in Eq. (1), as well as the temperature, which remains constant across all simulations. For the two-dimensional systems studied here, the temperature is set to $T^* = 1.4671$ following the results of Báez *et al.*²² for a two-dimensional system. By fixing λ , the values of A and B are also fixed in Eq. (2).

The thermodynamic state of our binary mixture is defined by four design parameters:

- the size ratio $q = \sigma_S / \sigma_L$,
- the small species composition $x_S = N_S / N$ with N_S the number of particles for the small species,
- the non-additivity parameter Δ ,
- and the packing fraction $\eta = (N_S \sigma_S^2 + N_L \sigma_L^2) \pi / 4A$ with A the area defined by the simulation box.

III. INVERSE DESIGN PROTOCOL

Our aim is to optimize the interaction parameters as well as the thermodynamic state point to favor the self-assembly of specific target phases in a non-additive binary hard-disk mixture. In this work, we build upon the inverse design protocol introduced in Ref. 10. This approach combines the covariance matrix adaptation evolution strategy (CMA-ES) to sample and optimize the set of design parameters, using a CNN to evaluate the fitness of each sample. The method consists of three steps. In the first step of the inverse design process,

a fixed number of parameter sets are drawn from a multivariate Gaussian distribution. The dimension of the multivariate Gaussian distribution is set by the number of free design parameters that need to be tuned. For each set of parameters, also called sample, we perform MD simulations of the system from which equilibrated configurations will be collected to compute diffraction patterns. In the second step, we evaluate the fitness of the samples by classifying their diffraction patterns using the CNN or the symmetry-based order parameter. Samples more likely to be classified as the target phase receive a higher fitness value. In the final step, we update the mean and covariance matrix of the Gaussian distribution to move towards regions within the parameter space where the fittest samples are located. Below, we describe the three steps in more detail.

A. Simulations

We perform MD simulations of the non-additive binary hard-disk mixture in a square simulation box with periodic boundary conditions in all directions using the LAMMPS simulation code version 28 Mar 2023.^{24–27} Each system consists of $N = N_S + N_L = 512$ particles. We initialize the system by randomly placing the particles within the simulation box at an initial packing fraction of $\eta = 0.3$. Subsequently, the simulation box is linearly compressed to a higher target packing fraction, determined by CMA-ES, over a duration of $10^8 \tau$, where $\tau = \sigma_L \sqrt{m/\epsilon}$ denotes the simulation time unit, and m represents the mass of a particle, which is kept identical for both species. This box deformation scheduling is used to reach the higher packing fractions where the crystals and quasicrystals can be self-assembled. A time step of $\tau = 0.001$ is employed for all simulations. Upon reaching the target packing fraction, an equilibration phase follows wherein the positions of the particles are propagated for at least $10^7 \tau$ to reach equilibrium. However, for systems exhibiting quasicrystal phases, at least $10^8 \tau$ was required for equilibration, which we determine by monitoring the time evolution of the energy of the system and the radial distribution function. Longer runs are needed in order to reduce the number of defects in the system.^{28,29} Subsequently, a production phase of at least $10^7 \tau$ is performed to collect a large number of independent configurations of the system. Configurations were collected every $10^4 \tau$, resulting in a total of 10^3 configurations. The various thermodynamic state points are explored using simulations in the canonical (NVT) ensemble with canonical thermostatting, employing the method introduced by Bussi *et al.*³⁰ The thermostat relaxation time is set to 100τ .

B. Convolutional neural network as classifier

The inverse design protocol used in Ref. 10 employs a CNN as a classifier. The output of the CNN provides a probability indicating how closely a given configuration resembles the target phase, effectively serving as a fitness function. Subsequently, a derivative-free optimization method is employed

to optimize the fitness of the samples. The CNN takes as input the diffraction patterns extracted from the configurations gathered during the simulations. The aim of employing a CNN is to classify the various phases within a non-additive binary hard-disk mixture based on their respective diffraction patterns. We anticipate this two-dimensional mixture to stabilize various phases, including, for example, an isotropic fluid phase (FLUID), hexagonal (HEX) and square (SQ) crystal phases, and quasicrystal phases such as the decagonal (QC10), dodecagonal (QC12), and octadecagonal (QC18) quasicrystal phases.

To train the CNN,¹⁰ we need diffraction patterns corresponding to these phases. As the aim of this paper is to inverse design these phases within a binary mixture of hard disks, these diffraction patterns are currently unavailable. However, data is available from a single-component system of particles interacting with a hard-core square-shoulder potential.¹⁰ This monodisperse system exhibits crystalline and quasicrystalline phases sharing similar translational and rotational symmetries as the anticipated phases in the binary mixture.^{10,28,31} In this study, we investigate the possibility of training the CNN using diffraction patterns from a monodisperse two-dimensional system interacting via a hard-core square-shoulder potential for inverse designing these phases in a binary mixture.^{10,28}

To generate training data, Monte Carlo simulations of the two-dimensional hard-core square-shoulder model were performed for a total of 5×10^5 Monte Carlo sweeps, where each sweep represents an attempt to randomly displace all particles in the system. The equilibration phase is followed by 1×10^6 sweeps, during which we save a configuration every 10^3 sweeps, resulting in 10^3 independent configurations. This is repeated for 10 different state points for each of the six considered phases, FLUID, HEX, SQ, QC12, QC10, and QC18. To ensure identification of phases regardless of their orientation, every training configuration is rotated by a random angle before evaluating its diffraction pattern following the procedure outlined in Ref. 10. The diffraction patterns for these one-component systems are obtained by computing the two-dimensional structure factor,

$$S(\mathbf{k}) = \frac{1}{N} \left| \sum_{j=1}^N e^{i\mathbf{k} \cdot \mathbf{r}_j} \right|^2, \quad (4)$$

where $\mathbf{k} = 2\pi(n_x, n_y)/L$ represents the wavevector with n_x and n_y denoting two integers, L the box length, and \mathbf{r}_j the center-of-mass position of particle j . We determine the diffraction pattern for each training configuration on a 150×150 grid, followed by a size reduction using a max pooling filter to achieve a final size of 33×33 . The primary goal of max pooling is to reduce the dimensions of the data while preserving important input features. Consequently, the max pooling step reduces the computational time and memory during training.

To classify and evaluate structures in our binary mixture, we consider only the large species in computing the diffraction patterns. The CNN used in this work is comprised of two convolutional layers for feature extraction and two fully connected layers for classification as depicted schematically in Fig. 1. Each convolutional layer performs two operations: fea-

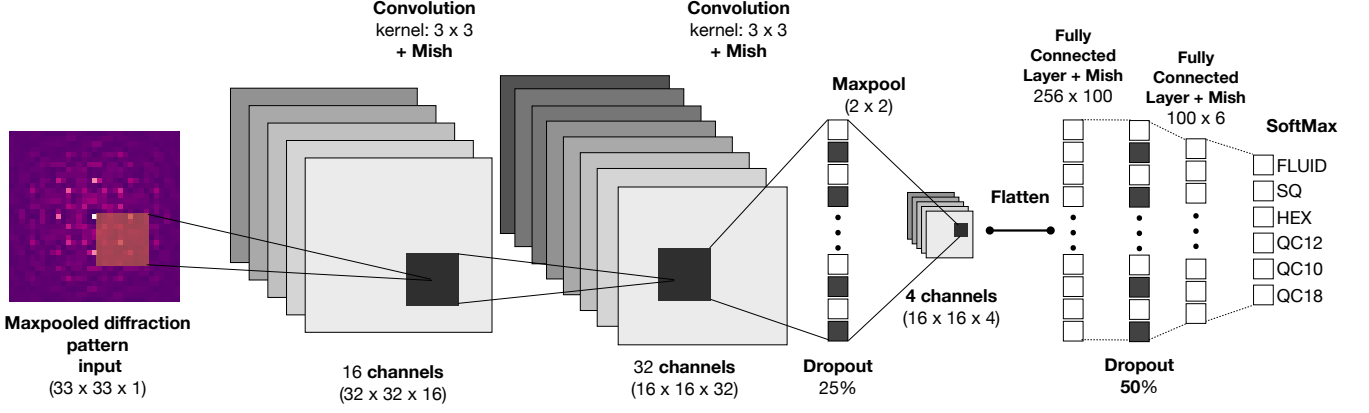


FIG. 1. **Schematic representation of the CNN used as a classifier in this work.** The network consists of two convolutional layers for feature extraction, followed by a MaxPooling layer to reduce resolution and two fully connected layers before a SoftMax function for final classification. All details about kernels, layer size, and activation functions are also shown. Each layer uses a non-linear *Mish* activation function. Note that between the fully connected layers there is a dropout layer to avoid overfitting.

ture extraction through convolutional filters and a non-linear transformation using the *Mish* activation function, defined as $f(x) = x \tanh[\log(1 + e^x)]$.³² The first convolutional layer has one input channel, corresponding to the maxpooled diffraction pattern, and sixteen output channels, which are the extracted features. The convolutional layer uses kernels of size 3×3 , with a stride of $s = 1$ and a padding of $p = 1$. The second convolutional layer takes as input the sixteen channels and outputs thirty-two channels in order to increase the features extracted from the images. The kernels in this layer have the same size, padding and stride as the first convolutional layer. We use a downsampling step after the channels through a layer with a 2×2 max pooling filter with a stride of $s = 2$. To enhance the robustness of the neural network, a dropout probability of $P = 0.25$ is introduced following the downsampling of the second convolutional layer. Dropout is a regularization technique that selectively eliminates feature detectors with a probability P , while the remaining weights are trained as usual.³³ The dropout layer is subsequently stacked, flattened, and prepared as input for classification, where two fully connected layers are used to classify the features extracted from the convolutional layers. The first fully connected layer consists of 256 units, and the second layer consists of 100 units. The *Mish* activation function is applied to all units in the layers. To further enhance robustness and introduce regularization, a dropout probability of $P = 0.5$ is applied between the two fully connected layers. The output layer contains six units to align with the available training classes, followed by a SoftMax function to determine classification probabilities.

We train the CNN by minimizing the cross-entropy loss while implementing L_2 regularization as weight decay of 10^{-5} .³⁴ This regularization of weights enhances robustness of the network, making it less prone to overfitting. The Adam optimizer³⁵ is used with a learning rate of 10^{-3} . To prevent overfitting, early-stopping is employed. If the loss remains unchanged between consecutive epochs, the training is halted and the optimization is assumed to have reached convergence within a maximum of fifty epochs. The dataset comprises a

total of 48000 samples, divided into 8000 samples per class, split in an 80 – 20% ratio for training and test sets. The test set, containing samples not used during training, is employed to evaluate the performance of the classifier, which reaches a final classification accuracy in the test set of at least 99.9%. The CNN implementation and training protocol are carried out using PyTorch and PyTorch Lightning.^{36,37}

C. Symmetry-Based Order Parameter

It is important to note that training the CNN relies on having the diffraction patterns for the target structures. However, when we aim to reverse engineer a structure without access to its diffraction pattern, this inverse design method will not succeed. To overcome this challenge, we introduce a symmetry-based order parameter leveraging the unique rotational symmetry within diffraction patterns, eliminating the need for training a CNN, and serving as a fitness function for inverse design protocols. To analyze the symmetry present in a diffraction pattern, we ascertain the count of detectable symmetry axes within the aforementioned pattern. More specifically, we identify the lines of symmetry that cut the diffraction pattern into two identical mirror images. To discern lines of symmetry, we posit a metric to quantify the degree of inverse reflectivity exhibited by a given axis. Mathematically, this reads

$$S(\theta) = \sum_{\mathbf{r} \in A} |I(Q(\theta) \cdot \mathbf{r}) - I(\mathbf{r})|, \quad (5)$$

where $A = \{(x, y) : x^2 + y^2 \leq R^2\}$ represents all points $\mathbf{r} = (x, y)$ lying within a circle of radius R centered around the origin of the diffraction pattern. The angle θ denotes the angle formed between the reflection axis and the x -axis as illustrated in Fig. 2(a). The peak intensity within the diffraction pattern at location \mathbf{r} is defined as $I(\mathbf{r})$. Finally, $Q(\theta)$ represents the reflection matrix across the reflection axis at an angle θ with

the x -axis

$$Q(\theta) = \begin{pmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{pmatrix}. \quad (6)$$

By inverse reflectivity we mean that a high value of $S(\theta)$ indicates a lack of reflection symmetry for that particular angle θ , i.e. there is no mirror symmetry between the two opposite sections of the diffraction pattern. Conversely, a low value of $S(\theta)$ suggests high reflection symmetry for that specific angle θ .

Applying the floor operation to the expression $Q(\theta) \cdot \mathbf{r}$ in Eq. (5) is essential to ensure the use of real pixel locations when dealing with the raw data from diffraction patterns. Fig. 2(a) provides a schematic representation of this method, illustrating the circle of radius R in the diffraction pattern. We note that as the method operates with real information from diffraction pattern images, it can readily be extended to experimental setups.

We calculate $S(\theta)$ for a uniform grid of M angles in the range of $\theta \in [0, \pi)$. We show $S(\theta)$ as a function of θ in Fig. 2(b), where we find an oscillatory pattern in $S(\theta)$ as a function of θ . The oscillation frequency of $S(\theta)$ correlates with the number of reflection axes of the diffraction pattern. To evaluate the number of symmetry axes, we compute the power spectral density $P(m)$. First, we subtract the mean from $S(\theta)$ in order to normalize the symmetry score across different samples. Subsequently, the power spectral density $P(m)$ is computed according to the equation

$$P(m) = \frac{1}{M} \left| \sum_{k=0}^{M-1} S(\theta_k) e^{-i2\pi m k \Delta\theta} \right|^2, \quad (7)$$

where $\theta_k = \theta_0 + k\Delta\theta$ represents the discrete angles used to compute the discrete symmetry scores $S(\theta_k)$, with $\theta_0 = 0$ and $k \in \{0, 1, 2, \dots, M-1\}$, i denotes the imaginary unit, $m \in \{0, 1, 2, \dots, M-1\}$ represents the m -fold rotational symmetry, and $|\cdot|^2$ denotes the norm of the complex numbers. Essentially, the power spectral density $P(m)$ decomposes the signal into its Fourier components such that the underlying rotational symmetries are easier to detect. A representative plot of the power spectral density $P(m)$ is shown in Fig. 2(c), where the high peaks correspond to the rotational symmetries obtained from the signal decomposition of $S(\theta)$. For an m -fold rotationally symmetric diffraction pattern, we expect to find a high value at the m value of the power spectral density $P(m)$. For instance, a QC8 exhibits a 8-fold rotational symmetry in its diffraction pattern, and the power spectral density $P(m)$ exhibits the 8-fold symmetry by having the largest value at $m = 8$, as shown in Fig. 2(c). It is essential to note that signal aliasing might occur, leading to high values at frequencies that are multiples of the sampling frequency, like the 16-fold and 24-fold symmetry for an 8-fold diffraction pattern (see the small peaks in Fig. 2(c)).³⁸

We can now define a symmetry-based order parameter f_m , serving as a fitness function

$$f_m = P(m) - \sum_{\substack{i=0, \\ i \neq m}}^{M-1} P(i), \quad (8)$$

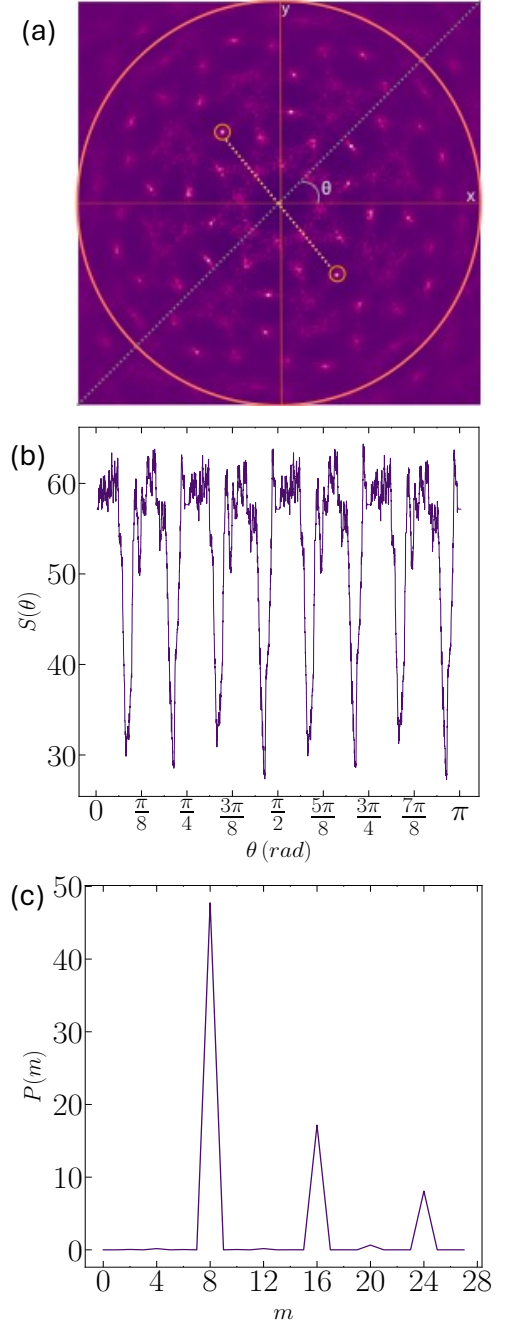


FIG. 2. **Symmetry-based order parameter.** (a) A grey line represents the reflection axis, forming an angle θ with the x -axis. Intensities of points from one half of the circle are subtracted point-wise from the intensity of their reflection points across the reflection axis on the other half of the circle. An example of intensities are depicted by small circles and connected by a dotted line in the image. The absolute sum of all intensity differences quantifies the degree of inverse reflectivity $S(\theta)$. (b) $S(\theta)$ as a function of θ . (c) The power spectral density $P(m)$ of $S(\theta)$ as a function of m , with m denoting the m -fold rotational symmetry.

where we anticipate a high value for $P(m)$ when the diffraction pattern displays m -fold symmetry. By subtracting all rotational symmetries that are not equal to m (the second term in Eq. (8)), we penalize symmetries that might induce aliasing effects of the m -fold symmetry. Subsequently, we can optimize f_m defined in Eq. (8) using an evolution strategy algorithm, similar to the fitness value as determined by the CNN. In this way, we promote the self-assembly of structures with m -fold symmetry. It is important to note here that f_m is not bounded from above, as opposed to the CNN, so the highest value of the peak is unknown beforehand.

D. Covariance Matrix Adaptation Evolution Strategy as Optimizer

In this study, we employ the covariance matrix adaptation evolution strategy (CMA-ES) for optimizing the fitness function, which provides a measure of how close the system is to the target phase. Here we present a short summary of the method. For further explanations, we direct the reader to the references cited in the following. The CMA-ES represents a stochastic zero-order optimization technique designed to optimize a real-valued, non-convex, and nonlinear function, operating without requiring gradient information of the function.³⁹ CMA-ES is an iterative algorithm which samples from a multivariate Gaussian distribution and updates the mean and the covariance matrix of the Gaussian distribution at each iteration. This process continues until a convergence criterion is met. In each iteration, often referred to as a *generation*, the algorithm draws n samples from a d -dimensional multivariate Gaussian distribution, where d represents the number of design parameters we wish to optimize. The fitness function is then evaluated for these n samples, and the outcomes are arranged in descending order. Subsequently, the top k samples from this particular generation are chosen as the best candidates, forming the elements of the set \mathbf{X} . In each generation, the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$ are updated to adjust the Gaussian distribution to reach the optimum.

Various proposals for updating these parameters have been introduced,⁴⁰ but here we focus on the separable CMA-ES (sepCMA-ES).⁴¹ The sepCMA-ES version constrains the covariance matrix to remain diagonal, enabling linear space and time complexity. The sepCMA-ES version outperforms the traditional CMA-ES algorithm in many optimization problems.⁴²

We start from an initial mean vector $\boldsymbol{\mu}$ sampled uniformly within the feasible parameter space, and an initial covariance matrix $\mathbf{C} = \mathbf{I}$, with \mathbf{I} the identity matrix with shape $d \times d$. The sepCMA-ES generates a new candidate solution using the

following equations

$$\begin{aligned} \mathbf{z}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{for } i = 1, \dots, n \\ \mathbf{x}_i &= \boldsymbol{\mu} + \sigma \mathbf{B} \mathbf{D} \mathbf{z}_i \\ \boldsymbol{\mu} &= \sum_{i=1}^k w_i \mathbf{x}_{i:k} \\ \mathbf{z} &= \sum_{i=1}^k w_i \mathbf{z}_{i:k} \end{aligned} \quad (9)$$

where $\mathbf{x}_{i:k}$ denotes the i -th best individual out of the k samples. The elements of \mathbf{z}_i are standard normal random variables. The matrix \mathbf{B} has the orthogonal eigenvectors of \mathbf{C} as columns, whereas the matrix \mathbf{D} has the eigenvalue square roots as diagonal elements. The parameter σ is the initial standard deviation used at the start of the optimization method. The distribution of weights is defined as

$$w_i = \frac{\log(k+1) - \log(i)}{\sum_{j=1}^k \log(k+1) - \log(j)}, \quad (10)$$

where i is the rank index of sample x , with $i = 1$ indicating the configuration with the highest f value. These equations update the candidate solution as well as the mean vector used to sample the multivariate normal distribution.

We now turn our attention to updating the covariance matrix, using the following equations

$$\begin{aligned} p_\sigma &= (1 - c_\sigma) p_\sigma + \sqrt{\mu_W c_\sigma (2 - c_\sigma)} \mathbf{B} \mathbf{z} \\ p_c &= (1 - c_c) p_c + H_\sigma \sqrt{\mu_W c_c (2 - c_c)} \mathbf{B} \mathbf{D} \mathbf{z} \\ \mathbf{C} &= (1 - c_{cov}) \mathbf{C} + \frac{1}{\mu_{cov}} c_{cov} p_c p_c^T \\ &\quad + c_{cov} \left(1 - \frac{1}{\mu_{cov}} \right) \sum_{i=1}^k w_i \mathbf{B} \mathbf{D} \mathbf{z}_{i:k} (\mathbf{B} \mathbf{D} \mathbf{z}_{i:k})^T, \end{aligned} \quad (11)$$

where the Heaviside function $H_\sigma = 1$ when the condition $\frac{\|p_\sigma\|}{\sqrt{1 - (1 - c_\sigma)^{2g}}} < (1.4 + \frac{2}{d+1}) E(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)$ is met. This is the condition that happens most often during the evolution path, but the other case $H_\sigma = 0$ can happen as well, which indicates that the evolution path is stalled. The d -dimensional vectors p_σ control the amplitude of the covariance matrix, while the d -dimensional vector p_c manipulates the directionality. Additionally, $\langle \|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \rangle$ represents the average length of a vector sampled from a standard multivariate normal distribution. This quantity is used for step-size control. All the other parameters are free parameters which are constant throughout the optimization procedure, and the values chosen are explained below.

Finally, we update the step size σ along with the covariance matrix and its decomposition, using the following equations

$$\begin{aligned} \sigma &= \sigma \exp \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma\|}{E(\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)} - 1 \right) \right) \\ \mathbf{D}^2 &= \text{diag}(\mathbf{C}), \end{aligned} \quad (12)$$

where $\text{diag}(\mathbf{C})$ is a diagonal matrix with the same diagonal elements as \mathbf{C} .

In this work, we use $n = 24$ and $k = 12$ for all our results. The value of the initial standard deviation changes for each optimization problem, since it depends on the bounds of the d -dimensional search space $[a, b]^d$. In all results presented here, $\sigma = 0.3(b - a)$ where a and b might be different depending on the phase to inverse design. The initial values for p_σ and p_c in Eq. (11) are null vectors. All free parameters c_i , μ_W and μ_{cov} in Eq. (11) and Eq. (12) are set to the default values as outlined in the original work⁴¹ and in the Python implementation by Nomura and Shibata⁴³ that we use in this work. Finally, the stopping criterion of the optimization scheme is determined by setting the number of generations to fifty generations. This is a common choice for stopping the optimization scheme in numerical optimization experiments, where the number of function evaluations is fixed.⁴⁴

IV. RESULTS

A. Inverse design of a hexagonal lattice

We start our investigation by reverse-engineering the hexagonal crystal (HEX) phase in a non-additive binary mixture of hard disks. Our goal is to identify the optimal set of design parameters that facilitates the self-assembly of a hexagonal lattice within this mixture. To accomplish this, we use the size ratio q , the small species composition x_S , the non-additivity parameter Δ , and the packing fraction η , as design parameters. The values of these parameters are sampled from a multivariate normal distribution in each generation according to the equations of sepCMA-ES.

To specifically target the hexagonal lattice, we employ the output of the trained CNN, which represents the probability that the diffraction pattern of a given configuration is classified as a hexagonal lattice, denoted as P_{HEX} , as the fitness function, i.e. $f = \bar{P}_{\text{HEX}}$. Here, the "bar" indicates that the input diffraction pattern for the classifier is an averaged diffraction pattern of fifty different configurations collected during the production phase of the simulation. It is important to note that the CNN is trained using diffraction patterns of fluid, crystalline, and quasicrystalline phases obtained from a monodisperse system interacting with a hard-core square-shoulder potential. We thus investigate whether the classification of the CNN trained with diffraction patterns of a monodisperse system is general enough to inverse design target phases in a binary system.

The results of the reverse-engineering process are presented in Fig. 3. The algorithm successfully finds the target hexagonal lattice in approximately eight generations, evident from the rapid rise in the mean fitness to a high value. By the tenth generation, the algorithm consistently samples state points that, on average, facilitate the self-assembly of the hexagonal lattice. Additionally, Fig. 3(c) shows a representative snapshot along with its diffraction pattern computed for the large species, exhibiting clear hexagonal symmetry.

The evolution of the design parameters is shown in Fig. 3(b). The algorithm quickly converges to the range of design parameters where the hexagonal lattice self-assembles.

Comparing these findings to those obtained in Feyen *et al.*¹⁸ it is expected that the inverse design protocol would favor the self-assembly of this phase as the stability region of the hexagonal lattice is extensive in the phase diagram. In our work, this is evident from the wide spread of samples with high fitness values, which gradually narrows as the evolution progresses and the optimization method converges. Subsequently, the probability distribution narrows around the estimated mean. This behavior is attributed to the rugged landscape of f , with multiple minima, allowing the hexagonal lattice to self-assemble across a broad range of design parameters. It is important to note that once the optimizer has focused on a local minimum, other samples from other minima get discarded due to the selection mechanism choosing the best samples within each generation. These challenges can be overcome by using different exploration mechanisms which could explore other minima during optimization, even after discovering a local minimum of f .^{42,45}

B. Inverse design of a dodecagonal quasicrystal

We now target a dodecagonal quasicrystal (QC12) phase in a non-additive binary hard-disk mixture, recently predicted in simulations.^{18,19} This quasicrystal comprises a random tiling of squares and equilateral triangles, exhibiting distinct 12-fold rotational symmetry. This square-triangle random tiling has been investigated extensively,⁴⁶ and observed in computer simulations of monodisperse systems.²⁸ Given that data for a one-component 12-fold quasicrystal (QC12) is already included in the training data of the CNN, it can readily be used to target this phase within the binary mixture.

We define the fitness function as $f = \bar{P}_{\text{QC12}}$, where P_{QC12} represents the probability that the diffraction pattern of a given configuration is classified as a QC12 by the CNN. Similar to the protocol used for the hexagonal lattice, the "bar" indicates that the input diffraction pattern for the classifier is an averaged diffraction pattern of fifty different configurations collected during the production phase of the simulation.

We present the results of the inverse design protocol in Fig. 4. We clearly observe that the optimization of the quasicrystal phase requires more generations to identify a local minimum than in the HEX case. However, once found, the evolution of parameters remains stable, eventually reaching a plateau. A clear indicator of the inverse design protocol approaching the target phase is monitoring the packing fraction η , which shows a consistent increase as the protocol converges. Higher packing fractions are favored, since the QC12 and other crystalline phases only self-assemble at high pressures and high packing fractions. In Fig. 4(a), we observe that after approximately ten generations, η starts to increase consistently toward values exceeding $\eta > 0.8$. Subsequently, the packing fraction remains close to $\eta \approx 0.84$.

The behavior observed in the other parameters, specifically q and x_S , contrasts with that of the packing fraction η . The size ratio q displays extensive variations within the first ten generations, as shown by the large spread in the samples in Fig. 4(b). During the first ten generations, at least 90% of the

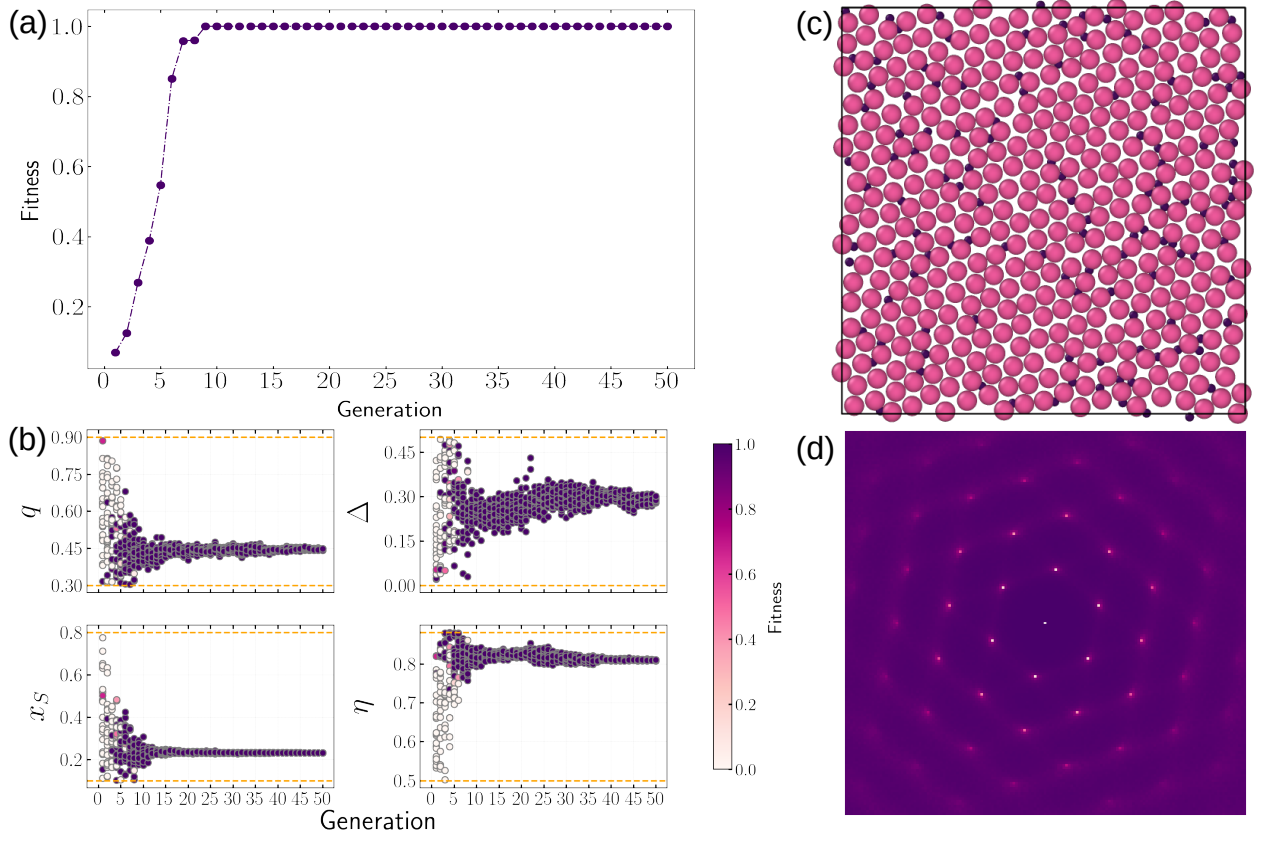


FIG. 3. **Inverse design of the hexagonal lattice in a non-additive hard-disk mixture using the CNN as order parameter.** (a) Evolution of the mean fitness during the inverse design protocol. (b) Evolution of the design parameters, size ratio $q = \sigma_S/\sigma_L$, non-additivity parameter Δ , small species composition x_S , and packing fraction η , for the inverse design of the hexagonal lattice. Each dot represents a sample for a given generation. The dotted lines are the bounds imposed for the optimization problem. (c) Exemplary configuration snapshot of a hexagonal lattice obtained during the last generation. (d) Diffraction pattern of the configuration in (c).

total range of the size ratio q is explored.

During the exploration phase, corresponding to the first fifteen generations of the sepCMA-ES, attempts are made to sample a wide region of the search space, until a high fitness state point is found, subsequently directing the optimizer toward that region. However, for both q and x_S , the region explored remains quite large, covering at least 60% of the total range, until it reaches a plateau at around generation number 30. The extensive exploration is required since the region of self-assembly of the octagonal quasicrystal is small compared to the size of the design parameter space. Still, the average values found for q and x_S agree well with Ref. 18.

Regarding the non-additivity parameter Δ , low values are obtained, which is expected since the small particles stabilize the random square-triangle tiling. Larger values of Δ would imply significant overlap between small and large particles, a scenario the optimizer has steered away from within the search space.

We would like to emphasize once more that the QC12 self-

assembled in the binary mixture comprises a random square-triangle tiling for the large species. The data used to train the CNN originates from a single-component system that also exhibits a square-triangle tiling.⁴⁶ It is important to note that the current training data may not be sufficient to facilitate the inverse design of other quasicrystals observed in simulations.^{28,29}

C. Inverse design of octagonal quasicrystal

So far, we have successfully reverse-engineered hexagonal crystals and dodecagonal quasicrystals within a non-additive binary hard-disk mixture using a CNN trained solely with diffraction patterns from a single-component system. However, in cases where we aim to inverse design structures within a binary mixture that are not available in the one-component system, we face a challenging recursive loop. The sought-after structure is necessary to train the CNN, while a

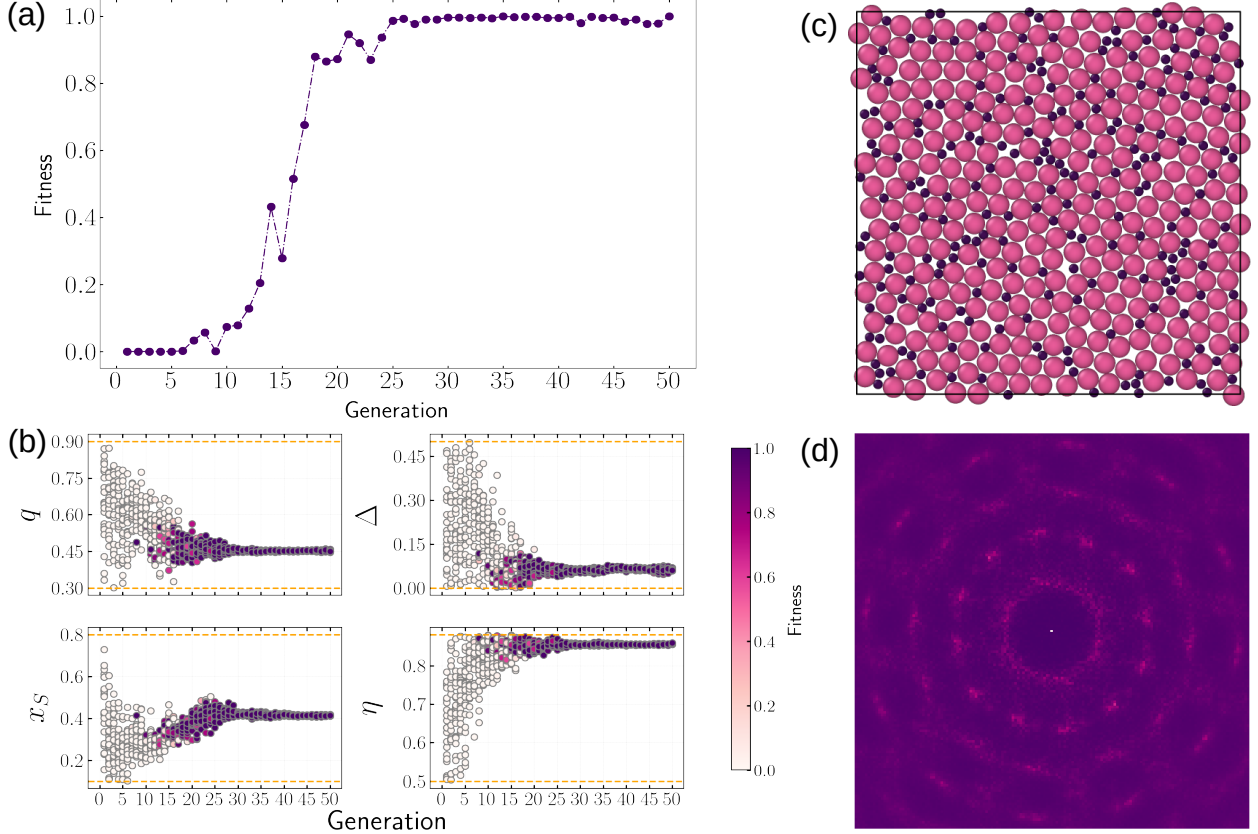


FIG. 4. **Inverse design of the QC12 in a non-additive hard-disk mixture using the convolutional neural network as order parameter.** (a) Evolution of the mean fitness during the inverse design protocol. (b) Evolution of the design parameters, size ratio $q = \sigma_S/\sigma_L$, non-additivity parameter Δ , small species composition x_S , and packing fraction η , for the inverse design of the QC12. Each dot represents a sample for a given generation. The dotted lines are the bounds imposed for the optimization problem. (c) Exemplary configuration snapshot of a QC12 obtained during the last generation. (d) Diffraction pattern of the configuration in (c).

trained CNN is crucial to inverse design that specific structure. An example of such a structure is the octagonal quasicrystal (QC8),¹⁸ for which diffraction patterns from a single-component system are absent in our training data. This absence poses a challenge when attempting to employ the CNN for the inverse design of QC8 within the binary mixture. We therefore resort to our symmetry-based order parameter, which bypasses the requirement for training a CNN and can be used as a fitness function in our inverse design protocol.

Our focus is now on achieving the octagonal quasicrystal structure using the fitness function presented in Section III C. Consistent with our previous endeavours, our objective is to identify the optimal set of design parameters that facilitates the self-assembly of QC8s. We focus on the following design parameters, the size ratio q , the small species composition x_S , and the packing fraction η . These parameters are again sampled from a multivariate Gaussian distribution at each generation. We compute the non-additivity parameter as a function of the size ratio q , using the expression $\Delta(q) =$

$(2\sqrt{q})/(1+q) - 1$ to facilitate a comparison with Ref. 18.

The fitness function is now defined as \bar{f}_8 , using the expression from Eq. (8) for f . In this case, the CNN is no longer used and the symmetry-based order parameter serves as the fitness function steering the optimization. Similar to the previous sections, the "bar" in the fitness function indicates that the input diffraction pattern for the classifier is an averaged diffraction pattern of fifty different configurations collected during the production phase of the simulation.

In Fig. 5(b), we present the evolution of the parameters for the QC8, displaying a plateau in the evolution of the fitness after twenty generations as shown in Fig. 5(a). The packing fraction η undergoes extensive exploration but quickly localizes at high values, as expected since the QC8 can only self-assemble at high packing fractions. Additionally, the size ratio q and small species composition x_S converge to values with high fitness after fifteen generations. These values align with Ref. 18. A representative snapshot and diffraction pattern are shown in Fig. 5(c) and Fig. 5(d), respectively, highlighting the

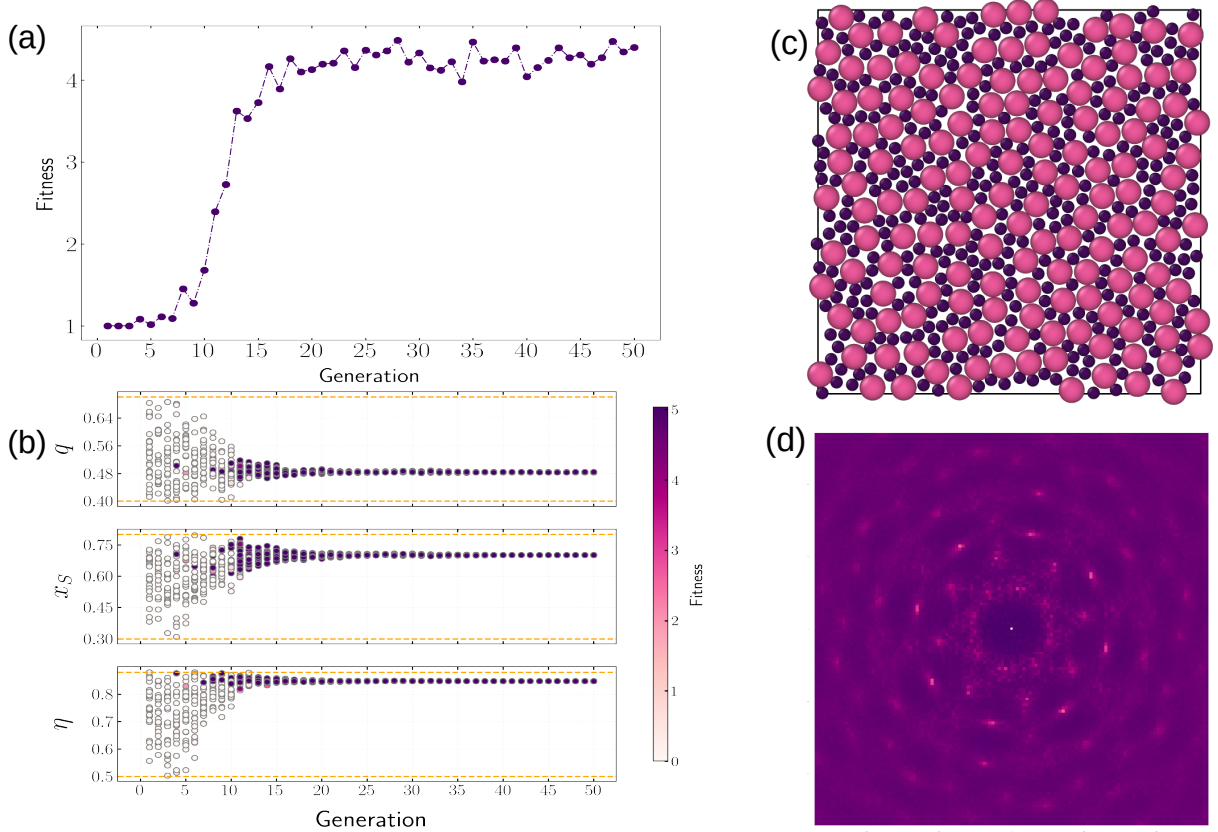


FIG. 5. **Inverse design of the octagonal quasicrystal in a non-additive hard-disk mixture using the symmetry-based order parameter.** (a) Evolution of the mean fitness during the inverse design protocol. (b) Evolution of the design parameters, size ratio $q = \sigma_S/\sigma_L$, small species composition x_S , and packing fraction η , for the inverse design of the QC8 using the symmetry-based fitness function. Each dot represents a sample from a specific generation. The dotted lines are the bounds imposed for the optimization problem. (c) Typical configuration snapshot of a octagonal quasicrystal obtained during the last generation. (d) Diffraction pattern of the configuration in (c).

octagonal symmetry.

Our results thus show that the fitness, defined in terms of the symmetry of the diffraction pattern, functions as intended. From Fig. 5(a), we observe that the evolution of the fitness function appears noisy in the later generations, a characteristic not observed in our results using the CNN. This could stem from the fitness function not being bounded but instead relying on obtaining peaks in the power spectral density $P(m)$ as large as possible. However, as these values can fluctuate due to penalties imposed on other frequencies, the maximum values of the peaks are not known in advance. One way to solve this issue could involve computing multiple scores simultaneously using different sets of diffraction patterns, or averaging the total fitness value during the evolution of the design parameters.

V. CONCLUSIONS

In conclusion, we have extended a recently introduced inverse design protocol to reverse-engineer crystalline and quasicrystalline structures within a two-dimensional non-additive binary hard-disk mixture. This method employs a CNN to characterize and classify diffraction patterns as similar or dissimilar to a target phase. This information can be used together with an optimizer, e.g. an evolutionary algorithm, to find the optimal state point and interaction parameters that facilitate the assembly of the target phase. More importantly, we show that we have successfully inverse designed hexagonal crystals and dodecagonal quasicrystals within a binary mixture using a CNN trained on solely diffraction patterns from a single-component system.

When dealing with unknown phases, the neural network cannot be trained since there is no readily available information about the system. To address this limitation, a symmetry-based order parameter was introduced, enabling the inverse

design of crystalline structures and quasicrystals with new symmetries. This order parameter determines the symmetry of the diffraction patterns by identifying the number of reflection axes. More specifically, this approach bypasses the necessity of training a CNN and the requirement for diffraction patterns of the structures to be reverse-engineered. By only changing the fitness function in our inverse design protocol, we successfully inverse designed octagonal quasicrystals within a binary mixture.

This study demonstrates that data from single-component systems is sufficient for training neural networks and reverse engineering structures in binary mixtures, provided that these phases are represented in the training data and large quantities can be obtained. While the neural network performs well when data of the specific structure is available, the symmetry-based order parameter serves as a valuable tool when the target structure data remains elusive.

ACKNOWLEDGMENTS

M.D., A.P.A.O. and E.A.B.M acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. ERC-2019-ADG 884902 SoftML). A.P.A.O. and E.A.B.M thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius.

DATA AVAILABILITY

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

- ¹A.-P. Hynninen, J. H. J. Thijssen, E. C. M. Vermolen, M. Dijkstra, and A. van Blaaderen, "Self-assembly route for photonic crystals with a bandgap in the visible region," *Nature Materials* **6**, 202–205 (2007).
- ²M. He, J. P. Gales, É. Ducrot, Z. Gong, G.-R. Yi, S. Sacanna, and D. J. Pine, "Colloidal diamond," *Nature* **585**, 524–529 (2020).
- ³K. Deguchi, S. Matsukawa, N. K. Sato, T. Hattori, K. Ishida, H. Takakura, and T. Ishimasa, "Quantum critical state in a magnetic quasicrystal," *Nature Materials* **11**, 1013–1016 (2012).
- ⁴M. Dijkstra and E. Luijten, "From predictive modelling to machine learning and reverse engineering of colloidal self-assembly," *Nature Materials* **20**, 762–773 (2021).
- ⁵Z. M. Sherman, M. P. Howard, B. A. Lindquist, R. B. Jadrich, and T. M. Truskett, "Inverse methods for design of soft materials," *Journal of Chemical Physics* (2020), 10.1063/1.5145177.
- ⁶M. C. Rechtsman, F. H. Stillinger, and S. Torquato, "Optimized Interactions for Targeted Self-Assembly: Application to a Honeycomb Lattice," *Physical Review Letters* **95**, 228301 (2005).
- ⁷U. T. Lieu and N. Yoshinaga, "Formation and fluctuation of two-dimensional dodecagonal quasicrystals," *Soft Matter* **18**, 7497–7509 (2022).
- ⁸M. Zu, P. Tan, and N. Xu, "Forming quasicrystals by monodisperse soft core particles," *Nature Communications* **8**, 2089 (2017).
- ⁹Y. Ma and A. L. Ferguson, "Inverse design of self-assembling colloidal crystals with omnidirectional photonic bandgaps," *Soft Matter* **15**, 8808–8826 (2019).
- ¹⁰G. M. Coli, E. Boattini, L. Filion, and M. Dijkstra, "Inverse design of soft materials via a deep-learning-based evolutionary strategy," *arXiv:2106.14615 [cond-mat]* (2021), [arxiv:2106.14615 \[cond-mat\]](https://arxiv.org/abs/2106.14615).
- ¹¹U. T. Lieu and N. Yoshinaga, "Inverse design of two-dimensional structure by self-assembly of patchy particles," *The Journal of Chemical Physics* **156**, 054901 (2022).
- ¹²N. Yoshinaga and S. Tokuda, "Bayesian modeling of pattern formation from one snapshot of pattern," *Physical Review E* **106**, 065301 (2022).
- ¹³E. G. Noya, C. K. Wong, P. Llombart, and J. P. Doye, "How to design an icosahedral quasicrystal through directional bonding," *Nature* **596**, 367–371 (2021).
- ¹⁴R. B. Jadrich, J. A. Bollinger, B. A. Lindquist, and T. M. Truskett, "Equilibrium cluster fluids: Pair interactions via inverse design," *Soft Matter* **11**, 9342–9354 (2015).
- ¹⁵W. D. Piñeros, B. A. Lindquist, R. B. Jadrich, and T. M. Truskett, "Inverse design of multicomponent assemblies," *The Journal of Chemical Physics* **148**, 104509 (2018).
- ¹⁶U. T. Lieu and N. Yoshinaga, "Dynamic control of self-assembly of quasicrystalline structures through reinforcement learning," (2023), [arxiv:2309.06869 \[cond-mat\]](https://arxiv.org/abs/2309.06869).
- ¹⁷D. V. Talapin, E. V. Shevchenko, M. I. Bodnarchuk, X. Ye, J. Chen, and C. B. Murray, "Quasicrystalline order in self-assembled binary nanoparticle superlattices," *Nature* **461**, 964–967 (2009).
- ¹⁸E. Feyen, M. Impéror-Clerc, L. Filion, G. Foffi, and F. Smalenburg, "Self-assembly of dodecagonal and octagonal quasicrystals in hard spheres on a plane," *Soft Matter* **19**, 2654–2663 (2023).
- ¹⁹E. Feyen, A. Jagannathan, G. Foffi, and F. Smalenburg, "Infinite-pressure phase diagram of binary mixtures of (non)additive hard disks," *The Journal of Chemical Physics* **152**, 204901 (2020).
- ²⁰D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications* (Academic Press San Diego, 2002).
- ²¹M. G. Noro and D. Frenkel, "Extended corresponding-states behavior for particles with variable range attractions," *The Journal of Chemical Physics* **113**, 2941–2944 (2000).
- ²²C. A. Báez, A. Torres-Carbajal, R. Castañeda-Priego, A. Villada-Balbuena, J. M. Méndez-Alcaraz, and S. Herrera-Velarde, "Using the second virial coefficient as physical criterion to map the hard-sphere potential onto a continuous potential," *The Journal of Chemical Physics* **149**, 164907 (2018).
- ²³J. Jover, A. J. Haslam, A. Galindo, G. Jackson, and E. A. Müller, "Pseudo hard-sphere potential for use in continuous molecular-dynamics simulation of spherical and chain molecules," *The Journal of Chemical Physics* **137**, 144505 (2012).
- ²⁴A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Comp. Phys. Comm.* **271**, 108171 (2022).
- ²⁵P. J. in 't Veld, S. J. Plimpton, and G. S. Grest, "Accurate and efficient methods for modeling colloidal mixtures in an explicit solvent using molecular dynamics," *Comput. Phys. Commut.* **179**, 320–329 (2008).
- ²⁶J. M. Monti, J. T. Clemmer, I. Srivastava, L. E. Silbert, G. S. Grest, and J. B. Lechman, "Large-scale frictionless jamming with power-law particle size distributions," *Phys. Rev. E* **106** (2022).
- ²⁷T. Shire, K. J. Hanley, and K. Stratford, "DEM simulations of polydisperse media: Efficient contact detection applied to investigate the quasi-static limit," *Computational Particle Mechanics* (2020).
- ²⁸T. Dotera, T. Oshiro, and P. Ziherl, "Mosaic two-lengthscale quasicrystals," *Nature* **506**, 208–211 (2014).
- ²⁹H. G. Schoberth, H. Emmerich, M. Holzinger, M. Dulle, S. Förster, and T. Gruhn, "Molecular dynamics study of colloidal quasicrystals," *Soft Matter* **12**, 7644–7654 (2016).
- ³⁰G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," *The Journal of Chemical Physics* **126**, 014101 (2007).
- ³¹H. Pattabhiraman and M. Dijkstra, "Phase behaviour of quasicrystal forming systems of core-corona particles," *The Journal of Chemical Physics* **146**, 114901 (2017).
- ³²D. Misra, "Mish: A self regularized non-monotonic activation function," (2020), [arXiv:1908.08681 \[cs.LG\]](https://arxiv.org/abs/1908.08681).
- ³³P. Baldi and P. J. Sadowski, "Understanding dropout," in *Advances in Neural Information Processing Systems*, Vol. 26, edited by C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (Curran Associates, Inc., 2013).

- ³⁴I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) <http://www.deeplearningbook.org>.
- ³⁵D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” (2017), arXiv:1412.6980 [cs.LG].
- ³⁶A. 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, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019) pp. 8024–8035.
- ³⁷W. Falcon and The PyTorch Lightning team, “PyTorch Lightning,” (2019).
- ³⁸P. Stoica, R. L. Moses, *et al.*, *Spectral analysis of signals*, Vol. 452 (Pearson Prentice Hall Upper Saddle River, NJ, 2005).
- ³⁹N. Hansen, “The CMA Evolution Strategy: A Tutorial,” arXiv:1604.00772 [cs, stat] (2016), arxiv:1604.00772 [cs, stat].
- ⁴⁰K. Varelas, A. Auger, D. Brockhoff, N. Hansen, O. A. ElHara, Y. Semet, R. Kassab, and F. Barbaresco, “A comparative study of large-scale variants of cma-es,” in *Parallel Problem Solving from Nature – PPSN XV*, edited by A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley (Springer International Publishing, Cham, 2018) pp. 3–15.
- ⁴¹R. Ros and N. Hansen, “A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity,” in *Parallel Problem Solving from Nature – PPSN X*, Lecture Notes in Computer Science, edited by G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni (Springer, Berlin, Heidelberg, 2008) pp. 296–305.
- ⁴²N. Hansen, “Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed,” in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, GECCO ’09 (Association for Computing Machinery, New York, NY, USA, 2009) pp. 2389–2396.
- ⁴³M. Nomura and M. Shibata, “cmaes : A simple yet practical python library for cma-es,” (2024), arXiv:2402.01373 [cs.NE].
- ⁴⁴L. M. Rios and N. V. Sahinidis, “Derivative-free optimization: a review of algorithms and comparison of software implementations,” *Journal of Global Optimization* **56**, 1247–1293 (2013).
- ⁴⁵A. Auger and N. Hansen, “A restart CMA evolution strategy with increasing population size,” in *2005 IEEE Congress on Evolutionary Computation*, Vol. 2 (2005) pp. 1769–1776 Vol. 2.
- ⁴⁶M. Oxborrow and C. L. Henley, “Random square-triangle tilings: A model for twelfold-symmetric quasicrystals,” *Physical Review B* **48**, 6966–6998 (1993).