arXiv:2404.08275v1 [cond-mat.soft] 12 Apr 2024

Machine Learning Assisted Sorting of Active Microswimmers

Abdolhalim Torrik^{1,*} and Mahdi Zarif^{1,†}

¹Department of Physical and Computational Chemistry, Shahid Beheshti University, Tehran 19839-9411, Iran (Dated: April 15, 2024)

Active matter systems being in a non-equilibrium state, exhibit complex behaviors such as selforganization and giving rise to emergent phenomena. There are many examples of active particles with biological origins, including bacteria and spermatozoa, or with artificial origins, such as selfpropelled swimmers and Janus particles. The ability to manipulate active particles is vital for their effective application e.g. separating motile spermatozoa from nonmotile and dead ones, to increase fertilization chance. In this study, we proposed a mechanism – an apparatus – to sort and demix active particles based on their motility values (Pèclet number). Initially, using Brownian simulations we demonstrated the feasibility of sorting self-propelled particles. Following this, we employed machine learning methods, supplemented with data from comprehensive simulations that we conducted for this study, to model the complex behavior of active particles. This enabled us to sort them based on their Pèclet number. Finally, we evaluated the performance of the developed models and showed their effectiveness in demixing and sorting the active particles. Our findings can find applications in various fields, including physics, biology, and biomedical science, where the sorting and manipulation of active particles play a pivotal role.

INTRODUCTION

Active matter systems – living and non-living – are defined by their ability to consume energy, generate forces, and show synchronized self-organization [1, 2]. These systems are intrinsically nonequilibrium and show complex behaviors and emergent phenomena [3–6]. They exist in various forms and scales, and can be biological or artificial, macroscopic or microscopic in nature. Examples of biological active systems with macroscopic origin include: flocks of birds [7], schools of fish [8], swarms of insects [9, 10], and herds of mammals [11]. Microscopic examples to highlight the dynamics of biological active matter systems include: bacteria [12, 13], sperm cells [14, 15], amoeba cell clusters [16–18], microbial biofilms [19], marine algae [20], and zooplanktons [21]. Self-propelled artificial swimmers (catalytic, magnetic and chemically driven micro-motors) [22–32], Janus particles [33–36], and robotic swarms [37] are examples of artificial active matter systems. All of the preceding examples are only a portion of the spectrum of active matter systems.

Despite advancements in understanding these systems, there's still a challenge: the effective sorting of active particles based on their motility values (Pèclet number). Manipulating active particles is a vital part of their effective application, there are many ways to achieve this, examples of such cases are: separation and rectification of swimming bacteria [38–40], using microswimmers for separating a mixture of colloids [41], aggregation and segregation of microswimmers based on their motilities [42, 43], using the confinement to manipulate microswimmers [44–48], separation [49] and sorting [50, 51] of microswimmers based on their chirality, motility [52] and separating motile spermatozoa [53, 54]. However, a

mechanism that can sort active particles based on their motility values, such as the Pèclet value, is yet to be developed.

Machine learning algorithms learn from the data and establish connections between the data to generate decisions/predictions for previously unobserved data without explicit programming [55–58].

In machine learning, there are three main branches: supervised, unsupervised, and semi-supervised learning [58]. In supervised learning, algorithms are trained to learn from input data and corresponding output data, input is processed to map it to new data based on expected outputs. This algorithms are used on both regression and classification tasks. Unsupervised learning algorithms are different, learning is done without any output data for the given input data. examples of such algorithms are clustering, PCA, etc. Semi-supervised learning falls between supervised and unsupervised learning, in this algorithm, a combination of small labeled data and a large amount of unlabeled data is used to train the model [59].

Machine learning has applications across many scientific and also practical disciplines. Within the fields of medicine and biology, machine learning example use cases are: drug discovery [60–65], neuroscience [66], biomedicine [67-70], cancer research [71-75], medical imaging [75–84]. Machine learning algorithms are also being used for material design [85–94], polymer science [95-101], crystallography [102-105], fluid dynamics [106-116], condensed matter physics [117–119]. Active mater physics also takes the advantage of machine learning [120–122], example applications include: detecting various phase transitions using Principal Component Analysis (PCA) [123, 124], active particle control using neural networks [125, 126], identifying anomalous diffusion using recurrent neural networks [127], forecasting dynamics of active nematics using neural networks [128–130], extracting the effective interaction of particles from the system configuration [131], optimal navigation strategies for active particles [132–140], characterizing motility induced regimes [141, 142], tracking Janus particles in three dimensions [143], and applying deep learning algorithms to plankton ecology [144].

In their study, Galajda et al. [145], developed an array of funnel-shaped barriers with wide and narrow openings which concentrated bacterial (E.coli) suspensions with motility on the side with narrow openings. This was because motile bacteria could swim into the funnels through wide openings while their ability to do the reverse through narrow openings was severely limited. Paoluzzi et al. Kumar et al. [146] explored the trapping phase transition of motile polar rods in the presence of a V-shaped obstacle. They found that when the trap angle of the obstacle was below a certain threshold, a trapping transition would occur, while above the threshold angle, all motile particles would escape. Ribeiro et al. [147] considered the influence of spatially periodic potentials on the trapping and sorting of motile active particles. Using simulations they showed that different diffusion regimes and trapping states occur which was a result of the noise value of the active particles and the density of the system. In their work, Ai et al. [148] investigated the mixing and demixing of binary mixtures of polar chiral (clockwise and anticlockwise) active particles with polar velocity alignment. They showed that when chirality difference and the polar velocity alignment of active particles compete with each other (neither is dominant), demixing occurs in which particles with clockwise rotation aggregate into one cluster and particles with anticlockwise chirality aggregate into another cluster. Thus using chirality was a means to demix particles. Miska et al. [149] proposed a method incorporating an acoustofluidic setup for selecting particles based on their motility, where particles with high motility would escape from the acoustic trap. They demonstrated their method using both simulations and also experiments with Janus particles and human sperm and proposed that using this method highly motile sperm could be selected for medically assisted reproduction.

Paoluzzi et al. [150] investigated the narrow escape time of active particles from circular domains using numerical simulations, they showed that narrow escape time undergoes a crossover between two asymptotic regimes with control parameters being the ratio of persistence length of the active particles and the length scale of the circular domain, they suggested the possibility of sorting active particles based on motility parameters. Based on this, we propose a mechanism – an apparatus – to sort (demix) active particles based on their motility. We use Brownian dynamics simulations and also incorporate machine learning methods to model the active particles using the data from an extensive number of comprehensive simulations that we conducted for this study.

METHODS

We consider a rigid circular apparatus with radius r_s which is made up of monomers on its circumference each having a radius of d_{mono} . The apparatus is inside a planar square box, which also contains active Brownian particles each with a radius a. We also pin the apparatus from its center of mass to the center of the box. Thus it can rotate with angular velocity ω , but prevented from moving. The angular velocity value can be varied in different simulations but has a constant value in each. We study our system in 2D, which is convenient to reduce computational resources while being effective in reproducing the key features of active models.

We use the over-damped Langevin equations to describe the motion of particles as time (t) progresses:

l

$$\dot{\mathbf{r}}_i = V_s \mathbf{n}_i - \mu_T \frac{\partial U}{\partial \mathbf{r}_i} + \sqrt{2D_T} \,\boldsymbol{\eta}_i(t), \qquad (1)$$

$$\dot{\theta}_i = \omega + \sqrt{2D_R} \zeta_i(t), \qquad (2)$$

here, $\mathbf{r}_i(t) = (x_i(t), y_i(t))$ is the position of the particle, V_s is the self-propulsion of the particle, and $\mathbf{n}_i =$ $(\cos \theta_i, \sin \theta_i)$ is direction of motion with $\theta_i(t)$ as orientation angles. μ_T is the translational mobility of particles, $\mathbf{f_i} \equiv -\partial U / \partial \mathbf{r}_i$ is the force acting on each particle, from interactions between the ith particle and all other particles. $D_T = \mu_T k_{\rm B} T$ is the translational diffusion coefficient of particles ($k_{\rm B}$ is the Boltzmann constant and T is the absolute temperature). ω is the angular velocity of particles. D_R is the rotational diffusion coefficient of particles which satisfies the relation $D_R = 3D_T/\sigma^2$ in low Reynold's regime. The symbols $\eta_i(t)$ and $\zeta_i(t)$ denote independent translational and rotational noises for each particle. They are characterized by white Gaussian distributions with zero mean, such that $\langle \eta_i^{\alpha}(t) \rangle =$ $\langle \zeta_i(t) \rangle = 0$. Additionally, they exhibit two-point correlations given by $\langle \eta i^{\alpha}(t) \eta j^{\beta}(t') \rangle = i j \delta \alpha \beta \delta(t-t')$ and $\langle \zeta_i(t)\zeta_j(t')\rangle = \delta(t-t')$, where i and j represent active particle labels, and α and β refer to the Cartesian coordinates x and y. The steric potentials, denoted as $V(\mathbf{r}_{ij})$, govern the interactions among active particles and the inclusions. These potentials are derived from modified versions of the Weeks-Chandler-Andersen potential (WCA):

$$V(\mathbf{r}_{ij}) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma_{\text{eff}}}{|\mathbf{r}_{ij}|} \right)^{12} - 2 \left(\frac{\sigma_{\text{eff}}}{|\mathbf{r}_{ij}|} \right)^{6} + 1 \right] & : |\mathbf{r}_{ij}| \le \sigma_{\text{eff}}, \\ 0 & : |\mathbf{r}_{ij}| > \sigma_{\text{eff}}, \end{cases}$$
(3)

The $P\acute{e}clet$ number (or rescaled self-propulsion strength), Pe_s , defined as

$$Pe_s = \frac{aV_s}{D_T} = \frac{3V_s}{4D_R a},\tag{4}$$

where we have used $D_R = 3D_T/4a^2$ for no-slip spheres in the low-Reynolds-number (Stokes) regime [151].

Using Brownian dynamics methods, we discretized the simulations over sufficiently small time steps $\Delta \tilde{t}$ to solve the equations (1) and (2) numerically.

RESULTS

Demixing

Fig 1 shows an example of demixing of four particles with different Pe_s , with system having $r_s = 10, 20, 30$ and 40, with $d_{mono} = 1.0$, $l_{pore} = 2.0$, and $k_B T = 1.0$, with Figs 1a and 1b being the initial and the final configurations, respectively. Demixing of particles in this figure occurs because of the system configuration and Pe_s values of particles that were chosen. Determining the optimal parameters for successful demixing necessitated a process of trial and error. Thus, with the aim of the occurrence of demixing and the ambiguity of the relationship between system parameters and the onset of demixing, we employ a grid search across various parameter combinations and system configurations. By leveraging machine learning methods we aim to gain a better understanding of the demixing phenomenon in active matter systems.

To model the demixing of particles from the apparatus, more than 75,000 simulations were done to collect data points with various parameters, namely: r_s , ω , d_{mono} , l_{pore} , and k_BT . Data were obtained using Brownian dynamics, simulations aimed to obtain Pe_s values for active particles at which they start going outside of the apparatus. The criteria were escaping 5% of particles after 500000 steps with a timestep of dt = 0.0001. This was the main component to have demixing for particles/apparatus with different properties. After obtaining the data from simulations, we proceeded with different machine learning algorithms to model active particle motility (Pèclet), each algorithm is considered in the following sections.

Machine Learning Models

Artificial Neural Network

In this section, we use ANN to model our systems. Typical workflow for modeling using ANN involves several steps, starting with collecting data and preprocessing it, preparing the model, training it, and evaluating the performance of the model. For collecting data, as stated earlier, 2500 relevant data points from a set of 75000 simulations were used. Input parameters for the data was r_s , ω , d_{mono} , l_{pore} , and k_BT with target values being



FIG. 1. Example of demixing of 4 particles with different Pe_s . System defined have $r_s = 10, 20, 30$ and $40, d_mono = 1.0, l_{pore} = 2.0, and k_BT = 1.0$

 Pe_s . Then we preprocessed our data using the Scikitlearn package [152], this step transforms the data into a format that is more convenient to train ANN.

Using Keras from TensorFlow, we created a dense (fully connected network), meaning that each of the neurons in a layer is connected to every neuron in the preceding and subsequent layers, this interconnectivity would make the ANN capable of modeling complex and nonlinear relationships between data. The architecture of our ANN which was designed for regression is comprised of (Fig 2): an input layer, a dense layer where each unit is connected to every feature in the input data, and a Rectified Linear Unit (ReLU) activation function. This was followed by a Batch Normalization Layer, we included this layer to normalize the activation of the previous layer to enhance the training stability and convergence speed. A Dropout Layer was added to "drop out" (set to zero) a fraction of the neurons from the previous layer at random while training the network. Using this we prevent the model from relying heavily on a neuron or a feature. Thus, making the network more general and not specialized and overly complex, in simple words prevents overfitting. This was succeeded with another dense layer and a ReLU activation function. This layer processes information from the input and learns to recognize complex patterns in the data. This layer also was followed by Batch Normalization and Dropout layers. A subsequent hidden layer with ReLU activation was included, followed by a dropout layer. Finally, we have the output layer which contains a single unit, appropriate for regression tasks where the goal is to predict a continuous value.

Following defining the ANN architecture, hyperparameters of the model were optimized to enhance the predictive performance, optimized hyperparameters are listed in Table I. We trained the model utilizing a custom Adam optimizer with the determined learning rate. The loss function we chose was MSE. We also utilized early stop-



FIG. 2. ANN architecture containing the input, dense layer with 512 neurons, a bath normalization layer, a dropout layer, dense layer with 128 neurons, a bath normalization layer, a dropout layer, dense layer with 64 neurons, a dropout layer, and the output layer

ping while training the model, this technique monitors MSE on a validation set and when the model has no improvement over a defined number of epochs, training stops. This prevents learning noise and aids generalization to unseen data.

After completion of training the ANN model, we assessed it on the data that we used to train the model and also the test set. We also used another external test set, which was not used or seen by the neural net, and we assessed it after getting the final model to see its predictive ability. We employed R-squared (R^2) , Mean Absolute Error (MAE), Root Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) as evaluation metrics, these showcase the predictive capabilities of the model, and are defined in the following:

$$R^{2} = 1 - \frac{\sum_{i=1}^{N} (y_{i} - \hat{y}_{i})^{2}}{\sum_{i=1}^{N} (y_{i} - \bar{y})^{2}}$$
(5)

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}$$
(6)

$$RMSE = \sqrt{\frac{\sum\limits_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}$$
(7)

$$MAPE = \frac{\sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|}{N} \times 100\%$$
(8)

N is the total number of data points that are getting evaluated, y_i , \hat{y}_i , and *bary* are the real, predicted, and

mean values of the entity being evaluated, respectively. For ANN model evaluation metrics are presented in Table II.

Support Vector Regression

In this section, we discuss the utilization of Support Vector Regression (SVR) to model our systems. The modeling steps for SVR are similar to those for ANN: data collection, data preprocessing, model preparation, model training, and model evaluation performance. Data collection and preprocessing are the same as before. To implement the SVR model, we used the scikit-learn library, specifically Epsilon-Support Vector Regression.

We used the Radial Basis Function (RBF) kernel in our SVR model. It is defined by the formula: $K(x, x') = e^{-\gamma \cdot ||x-x'||^2}$. Here, x and x' are the input vectors, $||x - x'||^2$ is the squared distance between them, and gamma is a parameter that controls the width of the Gaussian distribution. The RBF kernel can effectively capture complex, non-linear relationships in the data by transforming the data using the preceding formula. Gamma is the hyperparameter that needs to be tuned for optimal model performance.

Other hyperparameters for implementing this SVR model include C (regularization parameter) and epsilon (insensitivity parameter). Hyperparameter C controls how the model treats outliers. A low value of C implies that more outliers are allowed, while a high value of C implies that fewer outliers are tolerated. Hyperparameter epsilon is given by $L(y, f(x), \epsilon) = \max(0, |y - f(x)| - \epsilon)$, where y represents the true target value, f(x) is the predicted target value and ϵ denotes the width of the insensitive zone, controlling the tolerated error. Optimum hyperparameters and evaluation metrics for our SVR model are listed in Tables I, II, respectively.

Kernel Ridge Regression

Similar to the SVR, Kernel Ridge Regression (KRR) transforms the input data into a higher-dimensional space using a nonlinear kernel function. This makes it possible for the algorithm to learn complex relationships between the input data and the outputs. In this section, we will introduce and use KRR to model our systems.

The objective function in KRR is defined by the equation:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha}.$$
(9)

KRR is designed to fit the training data while controlling overfitting through regularization. The objective is to find the vector $\hat{\alpha}$ that minimizes the squared Euclidean norm between **y** and **K** α plus a regularization term ($||\mathbf{y}-$ $\mathbf{K}\boldsymbol{\alpha}\|_{2}^{2} + \lambda \boldsymbol{\alpha}^{T}\mathbf{K}\boldsymbol{\alpha}$). Here, $\boldsymbol{\alpha}$ is the coefficients (weights) assigned to each training sample in the kernelized feature space which is getting optimized. \mathbf{y} is the real (target) value. \mathbf{K} is the kernel matrix where each element is the result of applying a kernel function to the pair of input samples and $\mathbf{K}\boldsymbol{\alpha}$ is the transformed input data. The squared Euclidean norm between \mathbf{y} and $\mathbf{K}\boldsymbol{\alpha}$ measures the deviation between real and predicted values. The regularization term is $\lambda \boldsymbol{\alpha}^{T}\mathbf{K}\boldsymbol{\alpha}$, where λ is a constant parameter that specifies the strength of the regularization to prevent overfitting.

We implemented the KRR using the scikit-learn library and performed a grid search to obtain the optimal values of the hyperparameters. The hyperparameters included in the optimization were regularization strength, kernel type (Linear, polynomial, RBF, and sigmoid), gamma, which denotes the kernel coefficient (the width of the kernel) that is specific to each kernel being used, and the polynomial degree for the polynomial kernel (ignored by other kernels). Best performing hyperparameters obtained for the KRR model and the evaluation metrics for the KRR model are listed in Tables I, II, respectively.

Gaussian Process Regression

Gaussian Process Regression (GPR) is a probabilistic modeling technique used to capture non-linear relationships in data to perform regression tasks. Unlike conventional methods, GPR treats the target variable as a distribution over functions, rather than yielding singlepoint estimates. This allows for making predictions with associated uncertainty.

To perform a regression task using GPR, initially, we establish a prior belief, characterized by a mean function m(x) and a covariance function (kernel) (k(x, x')), before any data is observed. This represents the initial understanding of the function space. Combining the prior belief with the observed data results in a posterior distribution (Bayes' rule) which is Gaussian. The posterior is a refined understanding of the function space. This procedure is iterative, and with each new data point, the model's accuracy increases.

During training, we compute the posterior mean μ and covariance Σ using the following equations:

$$\mu(X_*, X, \mathbf{y}) = K(X_*, X) [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y}$$
(10)

$$\Sigma(X_*, X) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*)$$
(11)

K is the kernel matrix with $K_{ij} = k(x_i, x_j)$. This captures the pairwise similarities between input points. X_* represents new input points, **y** is the vector of observed values, σ_n^2 is the variance of the noise, and I is the identity matrix. When making predictions for new inputs, the posterior mean μ is the predicted value, while covariance Σ indicates the associated uncertainty.

We used the scikit-learn library to implement GPR for modeling our systems. Specifically, we used the method therein called GaussianProcessRegressor with a custom kernel function which is a combination of the product of a constant kernel (C) by an RBF kernel plus a white kernel to account for noise in data plus a "DotProduct" kernel with power four to act as a four-degree polynomial. Both the RBF kernel and DotProduct kernel (with power 4) account for the non-linear relationship between the input data and the targets. We used default hyperparameters for our GPR, listed in Table I, the reason being that all of these hyperparameters are optimized during the training by the implementation of GPR in the scikit-learn library. Evaluation metrics for our GPR model are also listed in Table II.

TABLE I. ML models hyperparameters

Model	Hyperparameter	Details
ANN	optimizer	Adam
	max_iter	10000
	early stopping patience	2000
	learning_rate	0.001
	batch_size	512
	hidden neurons	512, 128, 64
	activations	relu
	loss	mse
SVR	kernel	rbf
	С	100000
	epsilon	0.9
	gamma	0.05
KRR	kernel	polynomial
	degree	4
	λ	0.5
	gamma	0.2
GPR	kernel	Constant*rbf+
		WhiteKernel+ $DotProduct^4$
	alpha	0.1
	$n_restarts_optimizer$	10

Comparing the ML methods

Here, we compare the overall performance of all models using evaluation metrics listed in Table II. The R^2 values for all models are high, approaching unity, across all three datasets, suggesting a strong correlation between the variables and the target values. We also note that a high R^2 value for the training set would indicate overfitting, considering the table, R^2 values for test and external test sets are high and near the values of R^2 for their respective training set, this would mitigate the overfitting problem, thus, validating the generalizability of our models. The MAPE, MAE, and RMSE evaluation metrics too, further showcase consistent values across all three datasets, with minor variations, this too denotes the generalizability and consistency in the predictive power of our models.

TABLE II. Evaluation Metrics for Different Models

Model	Eval.	\mathbf{R}^{2}	MAPE	MAE	RMSE
NN	Train	0.9997	1.9539	1.1007	1.5084
	Test	0.9997	2.1825	1.2077	1.5331
	Ext.	0.9997	1.9725	1.2286	1.6334
SVR	Train	0.9996	1.4968	1.1768	1.8964
	Test	0.9995	1.6141	1.3136	2.0397
	Ext.	0.9993	1.7029	1.4192	2.4055
KRR	Train	0.9994	2.1512	1.5838	2.4108
	Test	0.9992	2.0918	1.6233	2.4612
	Ext.	0.9991	2.2392	1.7889	2.7267
GPR	Train	0.9995	2.7235	1.5265	2.2082
	Test	0.9994	2.6323	1.5563	2.2458
	Ext.	0.9993	2.7110	1.6136	2.4577



FIG. 3. Plots of prediction vs observed value for our models: a)NN, b)SVR, c)KRR, and d)GPR.

We also included Fig. 3, which illustrates the predicted values plotted against the observed values for each ML method we used, it includes the data for all three datasets – training, testing, and external validation – and also a guideline to show the perfect predictions (benchmark). Each plot shows how closely predictions from each of the ML models match up to the actual target values across the datasets.

Overall, based on the evaluation metrics and also plots



FIG. 4. Feature importance using SHAP analysis. The xaxis represents the SHAP values and the y-axis represents the feature names. Feature values change from cyan to blue as they increase from low to high. The most important feature is placed at the top. Here, a positive (negative) SHAP value for a feature means that the prediction increases (decreases) with an increase (decreases) in the given feature value.

of predicted vs observed values, the predictive power of the ML methods from the highest performing to the lowest one – with a minimal variation – is: ANN, SVR, GPR, and SVR respectively.

Model Interpretation

All the models we used, as well as many ML methods, present challenges to rationalizing and interpreting the results of predictions, this is due to their intrinsic black-box nature. To overcome this shortcoming we will use the SHapley Additive exPlanations (SHAP) method, utilizing SHAP we can measure the contribution of each feature on the model's predictions [153].

In Figure 4, x coordinates show the SHAP values and the y coordinate depicts the feature names, there is also color coding for the values of each feature from low to high when going from cyan to blue. This plot also shows the importance of features, with the most important one placed at the top. We also note that a positive (negative) SHAP value for one feature means the prediction, namely the Pèclet value, increases (decreases) with the given feature value. Feature importance from the figure is: $r_s > \omega > l_{pore} > d_{mono} > k_B T$. Further, from the color coding in the figure, for radius (r_s) , increasing the feature values increases Pècelt values. This is in line with the apparatus design and configuration of our systems, higher radius (r_s) values would mean a higher linear velocity for the monomers of the apparatus while all other parameters including angular velocity (ω) are kept constant, this translates into higher Pèclet values needed for active particles to escape outside of the apparatus. This is the same for the angular velocity of the apparatus, an increase in this feature would lead to an increase in the velocity needed for particles to pass through the pore of the apparatus, as it is shown from color coding for ω . Higher monomer diameter d_{mono} also shows increased



FIG. 5. Last snapshots of the simulations, radial probability distributions, and also number of particles going outside of their respective confinements. a, c, e for system with $\omega=1.0$, $k_BT=1.0$, $l_{pore}=2.0$, $d_{mono}=1.0$, and Pe_s : 27.28, 66.11, 105.65, 145.47. b, d, f for systems with $\omega=3.0$, $k_BT=1.0$, $l_{pore}=4.0$, $d_{mono}=2.0$, and Pe_s : 70.57, 162.37, 250.83, 338.16. Colors indicate particle types with specific Péclet values that are predicted with ANN. In the legend for e and f, the first number indicates the number of particles that escaped their confinement, and the second value is their ratio (same color).

SHAP values, considering that when all parameters are kept constant, a higher d_{mono} would mean smaller pore length and bigger contact area for the swimmers to pass through the pore. For pore length l_{pore} this means that a higher value of the feature would decrease the Pèclet needed for active particles to go outside the apparatus, as evident from the figure. Finally, as it can be seen from the figure, increasing k_BT decreases the SHAP values, it is intuitive that increasing k_BT values would mean a higher activity of the swimmers, and the Pèclet values needed to escape get smaller.

Demixing using ML

Here, we showcase how our ML models define our systems capable of demixing. We consider our bestperforming model, namely ANN. Figures 5a and 5b show the last snapshots of simulations of two separate systems in which we modeled the system using our ANN model. This figure also includes the radial probability distributions of the particles for their respective systems, Figures 5c and 5d. Further, we show the number of particles that escape their respective confinements and the ratio of escaped particles to the total particles of the respective type in Figures 5e and 5f. Figure 5a is the last snapshot of the system with $\omega=1.0$, $k_BT=1.0$, $l_{pore}=2.0$, $d_{mono}=1.0$. Pe_s values predicted for this system using our GPR model are Pe_s : 27.28, 66.11, 105.65, 145.47. We used these parameters and Pe_s values to simulate the system, and from the figure, we see the onset of demixing. Figure 5c shows the Radial Probability Distribution for this system, from this, we see that most of the particles remain in between their respective confinements. From Figure 5e, 95% of particles of type A remain below the radius 10 and only 5% of them go outside. For other types of particles, namely B, C, and D, less than 5% of each goes above their respective confinement radii, namely 20, 30, and 40 respectively.

Other system with $\omega=3.0$, $k_BT=1.0$, $l_{pore}=4.0$, $d_{mono}=2.0$ and predicted values of Pe_s : 70.57, 162.37, 250.83 and 338.16 showcase a similar pattern (see Figures 5b,5d,5f), showing effectiveness of our ANN to predict Péclet values for demixing.

DISCUSSION

The trapping phenomena of active particles, as observed in various experimental and numerical studies, have been instrumental in advancing our understanding of how geometrically designed boundaries can be leveraged for effective substance purification and microorganism control. These studies have demonstrated the potential of active [146, 147, 154] and chiral active [49, 50, 148, 155–157] particles to navigate through geometrically designed boundaries, offering promising applications in the purification of substances by selectively eliminating unwanted microorganisms. A notable aspect of these studies is the exploration of narrow escape times of active particles from a circular domain, which was achieved using numerical simulation [150]. In addition, trapping particles using an array of funnel-shaped barriers with wide and narrow openings was developed to concentrate bacteria (E. coli) suspensions with motility on the side with narrow openings [145]. Another study investigated the impact of spatially periodic potentials on trapping and sorting motile active particles [147]. Furthermore, a study proposed a method that incorporates an acoustofluidic setup for selecting particles based on their motility, where particles with high motility would escape from the acoustic trap [149]. This method was demonstrated using both simulations and experiments with Janus particles and human sperm, proposing that this method could be used to select highly motile sperm for medically assisted reproduction. All of the previous research focused on entrapment and sorting active particles, however, one key challenge that remained unexplored was sorting microorganisms with various propulsions with a high selectivity.

In the present study, we have developed a novel mechanism to sort active particles based on their motility values, specifically the Pèclet number, utilizing a combination of Brownian dynamics simulations and machine learning methods. Our proposed mechanism is based on a circular spinning apparatus with pores allowing active particles with certain Pèclet numbers to pass through it. Initially, we devised a system with a configuration that would demix and separate active particles with four different motilities. To better understand the system configurations required to separate motile particles, considering their complex and non-equilibrium dynamics, the utilization of machine learning methods was necessary.

Using Brownian dynamics we conducted an extensive number of simulations to generate a dataset to be used by the machine learning methods to capture the complex dynamics of active particles. After gathering the data, four machine learning methods, namely: artificial neural network, support vector regression, kernel ridge regression, and Gaussian process regression were developed, with the best performing one being the artificial neural network. Further, we showcased the onset of demixing and sorting based on the Pèclet numbers predicted by the neural network, simulation snapshots, plots of radial probability distribution, and also plots of the escape number of particles from the apparatus were included as visual and numerical supplements to our findings. We also employed the SHAP method to consider the importance of system configurations and to interpret the model's predictions, this revealed that the radius (r_s) and angular velocity (ω) of the apparatus, the monomer diameter (d_{mono}) , and the pore length (l_{pore}) significantly influenced the Pèclet numbers required for active particles to escape the apparatus, while $k_B T$ values, indicating the noise part of the

motion, resulted in lower required Pèclet numbers and had the lowest impact.

Although our system was confined to two dimensions. such a mechanism could easily be implemented in 3D experimental setups. However, further research is needed to explore the effectiveness of this mechanism with different motility parameters and in real-world systems. Further, with significant advancements in microfluidic device fabrication techniques [158, 159], and the introduction of 3D printing technologies, rapid and single-step production of intricate microfluidic devices is not only more accessible but has also expanded the design possibilities [160, 161], [162]. Such devices show great potential as advanced systems for manipulating and immobilizing small organisms. Therefore, we believe that our proposed mechanism and apparatus can be designed and fabricated with an autonomous mechanism, opening up possibilities for applications in various fields such as physics, biology, and drug delivery.

This work was supported by the Iran National Science Foundation (INSF) via Grant No. 4003065. Computational resources were provided by the Center for High-Performance Computing (SARMAD) at Shahid Beheshti University, Tehran. We acknowledge useful discussions with R. K. Bowles.

* yashar.torrik@gmail.com

- [†] m_zarif@sbu.ac.ir; Corresponding author: m_zarif@sbu.ac.ir
- M. Das, C. F. Schmidt, and M. Murrell, Introduction to active matter, Soft Matter 16, 7185 (2020).
- [2] G. Volpe, C. Bechinger, F. Cichos, R. Golestanian, H. Löwen, M. Sperl, and G. Volpe, Active matter in space, npj Microgravity 8, 54 (2022).
- [3] C. N. Likos, Effective interactions in soft condensed matter physics, Physics Reports 348, 267 (2001).
- [4] P. Romanczuk, M. Bär, W. Ebeling, B. Lindner, and L. Schimansky-Geier, Active brownian particles-from individual to collective stochastic dynamics p., The European Physical Journal Special Topics **202** (2012).
- [5] C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, and G. Volpe, Active particles in complex and crowded environments, Reviews of Modern Physics 88, 045006 (2016).
- [6] M. F. Hagan and A. Baskaran, Emergent selforganization in active materials, Current opinion in cell biology 38, 74 (2016).
- [7] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, *et al.*, Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study, Proceedings of the national academy of sciences **105**, 1232 (2008).
- [8] M. Moussaid, S. Garnier, G. Theraulaz, and D. Helbing, Collective information processing and pattern formation in swarms, flocks, and crowds, Topics in Cognitive Sci-

ence 1, 469 (2009).

- [9] J. Buhl, D. J. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, From disorder to order in marching locusts, Science **312**, 1402 (2006).
- [10] M. Tennenbaum, Z. Liu, D. Hu, and A. Fernandez-Nieves, Mechanics of fire ant aggregations, Nature materials 15, 54 (2016).
- [11] T. Vicsek and A. Zafeiris, Collective motion, Physics reports 517, 71 (2012).
- [12] D. B. Kearns, A field guide to bacterial swarming motility, Nature Reviews Microbiology 8, 634 (2010).
- [13] H. C. Berg, *E. coli in Motion* (Springer, New York, 2003).
- [14] E. A. Gaffney, H. Gadêlha, D. J. Smith, J. R. Blake, and J. C. Kirkman-Brown, Mammalian sperm motility: observation and theory, Annual Review of Fluid Mechanics 43, 501 (2011).
- [15] D. Woolley, Motility of spermatozoa at surfaces, Reproduction 126, 259 (2003).
- [16] W.-J. Rappel, A. Nicol, A. Sarkissian, H. Levine, and W. F. Loomis, Self-organized vortex state in twodimensional dictyostelium dynamics, Physical review letters 83, 1247 (1999).
- [17] D. A. Kessler and H. Levine, Pattern formation in dictyostelium via the dynamics of cooperative biological entities, Physical Review E 48, 4801 (1993).
- [18] S. Nagano, Diffusion-assisted aggregation and synchronization in dictyostelium discoideum, Physical review letters 80, 4826 (1998).
- [19] K. Drescher, J. Dunkel, C. D. Nadell, S. Van Teeffelen, I. Grnja, N. S. Wingreen, H. A. Stone, and B. L. Bassler, Architectural transitions in vibrio cholerae biofilms at single-cell resolution, Proceedings of the National Academy of Sciences 113, E2066 (2016).
- [20] R. E. Goldstein, Green algae as model organisms for biological fluid dynamics, Annu. Rev. Fluid Mech 47, 343 (2015).
- [21] G. Jékely, J. Colombelli, H. Hausen, K. Guy, E. Stelzer, F. Nédélec, and D. Arendt, Mechanism of phototaxis in marine zooplankton, Nature 456, 395 (2008).
- [22] W. F. Paxton, K. C. Kistler, C. C. Olmeda, A. Sen, S. K. St. Angelo, Y. Cao, T. E. Mallouk, P. E. Lammert, and V. H. Crespi, Catalytic nanomotors: autonomous movement of striped nanorods, Journal of the American Chemical Society **126**, 13424 (2004).
- [23] R. Kumar, M. Kiristi, F. Soto, J. Li, V. V. Singh, and J. Wang, Self-propelled screen-printable catalytic swimmers, RSC advances 5, 78986 (2015).
- [24] S. C. Takatori, R. De Dier, J. Vermant, and J. F. Brady, Acoustic trapping of active matter, Nature communications 7, 10694 (2016).
- [25] V. Narayan, S. Ramaswamy, and N. Menon, Long-lived giant number fluctuations in a swarming granular nematic, Science **317**, 105 (2007).
- [26] G. Kokot, G. V. Kolmakov, I. S. Aranson, and A. Snezhko, Dynamic self-assembly and self-organized transport of magnetic micro-swimmers, Scientific reports 7, 14726 (2017).
- [27] L. Zhang, J. J. Abbott, L. Dong, B. E. Kratochvil, D. Bell, and B. J. Nelson, Artificial bacterial flagella: Fabrication and magnetic control, Applied Physics Letters 94, 10.1063/1.3079655 (2009).
- [28] S. Sánchez, L. Soler, and J. Katuri, Chemically powered micro-and nanomotors, Angewandte Chemie Inter-

national Edition 54, 1414 (2015).

- [29] K. J. Rao, F. Li, L. Meng, H. Zheng, F. Cai, and W. Wang, A force to be reckoned with: A review of synthetic microswimmers powered by ultrasound, Small 11, 2836 (2015).
- [30] S. Mohanty, I. S. Khalil, and S. Misra, Contactless acoustic micro/nano manipulation: a paradigm for next generation applications in life sciences, Proceedings of the Royal Society A 476, 20200621 (2020).
- [31] H. Zhou, C. C. Mayorga-Martinez, S. Pané, L. Zhang, and M. Pumera, Magnetically driven micro and nanorobots, Chemical Reviews 121, 4999 (2021).
- [32] M. Sitti, H. Ceylan, W. Hu, J. Giltinan, M. Turan, S. Yim, and E. Diller, Biomedical applications of untethered mobile milli/microrobots, Proceedings of the IEEE 103, 205 (2015).
- [33] A. Walther and A. H. Muller, Janus particles: synthesis, self-assembly, physical properties, and applications, Chemical reviews 113, 5194 (2013).
- [34] S. Jiang, Q. Chen, M. Tripathy, E. Luijten, K. S. Schweizer, and S. Granick, Janus particle synthesis and assembly, Advanced materials 22, 1060 (2010).
- [35] J. Zhang, B. A. Grzybowski, and S. Granick, Janus particle synthesis, assembly, and application, Langmuir 33, 6964 (2017).
- [36] E. Poggi and J.-F. Gohy, Janus particles: from synthesis to application, Colloid and Polymer Science 295, 2083 (2017).
- [37] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, Self-organized flocking in mobile robot swarms, Swarm Intelligence 2, 97 (2008).
- [38] M. B. Wan, C. J. Olson Reichhardt, Z. Nussinov, and C. Reichhardt, Rectification of swimming bacteria and self-driven particle systems by arrays of asymmetric barriers, Phys. Rev. Lett. 101, 018102 (2008).
- [39] J. Tailleur and M. Cates, Sedimentation, trapping, and rectification of dilute bacteria, Europhysics Letters 86, 60002 (2009).
- [40] C. Maggi, A. Lepore, J. Solari, A. Rizzo, and R. Di Leonardo, Motility fractionation of bacteria by centrifugation, Soft Matter 9, 10885 (2013).
- [41] W. Yang, V. Misko, K. Nelissen, M. Kong, and F. Peeters, Using self-driven microswimmers for particle separation, Soft Matter 8, 5175 (2012).
- [42] S. R. McCandlish, A. Baskaran, and M. F. Hagan, Spontaneous segregation of self-propelled particles with different motilities, Soft Matter 8, 2527 (2012).
- [43] X. Yang, M. L. Manning, and M. C. Marchetti, Aggregation and segregation of confined active particles, Soft matter 10, 6477 (2014).
- [44] Y. Fily, A. Baskaran, and M. F. Hagan, Dynamics of self-propelled particles under strong confinement, Soft matter 10, 5609 (2014).
- [45] I. Berdakin, Y. Jeyaram, V. Moshchalkov, L. Venken, S. Dierckx, S. Vanderleyden, A. Silhanek, C. Condat, and V. I. Marconi, Influence of swimming strategy on microorganism separation by asymmetric obstacles, Physical Review E 87, 052702 (2013).
- [46] C. Reichhardt and C. J. O. Reichhardt, Dynamics and separation of circularly moving particles in asymmetrically patterned arrays, Phys. Rev. E 88, 042306 (2013).
- [47] G. Volpe, S. Gigan, and G. Volpe, Simulation of the active brownian motion of a microswimmer, American Journal of Physics 82, 659 (2014).

- [48] A. Torrik, A. Naji, and M. Zarif, Dimeric colloidal inclusion in a chiral active bath: Effective interactions and chirality-induced torque, Phys. Rev. E 104, 064610 (2021).
- [49] B.-q. Ai, Y.-f. He, and W.-r. Zhong, Chirality separation of mixed chiral microswimmers in a periodic channel, Soft Matter 11, 3852 (2015).
- [50] Q. Chen and B.-q. Ai, Sorting of chiral active particles driven by rotary obstacles, The Journal of Chemical Physics 143, 104113 (2015).
- [51] M. Mijalkov and G. Volpe, Sorting of chiral microswimmers, Soft Matter 9, 6376 (2013).
- [52] A. Costanzo, J. Elgeti, T. Auth, G. Gompper, and M. Ripoll, Motility-sorting of self-propelled particles in microchannels, Europhysics Letters 107, 36003 (2014).
- [53] D. S. Guzick, J. W. Overstreet, P. Factor-Litvak, C. K. Brazil, S. T. Nakajima, C. Coutifaris, S. A. Carson, P. Cisneros, M. P. Steinkampf, J. A. Hill, *et al.*, Sperm morphology, motility, and concentration in fertile and infertile men, New England Journal of Medicine **345**, 1388 (2001).
- [54] T. Chinnasamy, J. L. Kingsley, F. Inci, P. J. Turek, M. P. Rosen, B. Behr, E. Tüzel, and U. Demirci, Guidance and self-sorting of active swimmers: 3d periodic arrays increase persistence length of human sperm selecting for the fittest, Advanced Science 5, 1700531 (2018).
- [55] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A highbias, low-variance introduction to machine learning for physicists, Physics reports 810, 1 (2019).
- [56] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, Unsupervised learning methods for molecular simulation data, Chemical Reviews **121**, 9722 (2021).
- [57] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science **349**, 255 (2015).
- [58] C. M. Bishop, Pattern Recognition and Machine Learning, 1st ed., Information Science and Statistics (Springer, New York, NY, 2006) hardcover edition published on 17 August 2006.
- [59] J. E. Van Engelen and H. H. Hoos, A survey on semisupervised learning, Machine learning 109, 373 (2020).
- [60] S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, and M. J. Ahsan, Machine learning in drug discovery: a review, Artificial Intelligence Review 55, 1947 (2022).
- [61] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, *et al.*, Applications of machine learning in drug discovery and development, Nature reviews Drug discovery 18, 463 (2019).
- [62] P. Carracedo-Reboredo, J. Liñares-Blanco, N. Rodríguez-Fernández, F. Cedrón, F. J. Novoa, A. Carballal, V. Maojo, A. Pazos, and C. Fernandez-Lozano, A review on machine learning approaches and trends in drug discovery, Computational and structural biotechnology journal **19**, 4538 (2021).
- [63] L. Patel, T. Shukla, X. Huang, D. W. Ussery, and S. Wang, Machine learning methods in drug discovery, Molecules 25, 5277 (2020).
- [64] J. Jiménez-Luna, F. Grisoni, and G. Schneider, Drug discovery with explainable artificial intelligence, Nature Machine Intelligence 2, 573 (2020).
- [65] S. Kolluri, J. Lin, R. Liu, Y. Zhang, and W. Zhang,

Machine learning and artificial intelligence in pharmaceutical research and development: a review, The AAPS Journal **24**, 1 (2022).

- [66] K. P. Kording, A. Benjamin, R. Farhoodi, and J. I. Glaser, The roles of machine learning in biomedical science, in *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2017 Symposium* (National Academies Press, 2018).
- [67] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol, Ai in health and medicine, Nature medicine 28, 31 (2022).
- [68] J. Goecks, V. Jalili, L. M. Heiser, and J. W. Gray, How machine learning will transform biomedicine, Cell 181, 92 (2020).
- [69] A. Z. Woldaregay, E. Årsand, T. Botsis, D. Albers, L. Mamykina, and G. Hartvigsen, Data-driven blood glucose pattern classification and anomalies detection: machine-learning applications in type 1 diabetes, Journal of medical Internet research 21, e11030 (2019).
- [70] N. S. Artzi, S. Shilo, E. Hadar, H. Rossman, S. Barbash-Hazan, A. Ben-Haroush, R. D. Balicer, B. Feldman, A. Wiznitzer, and E. Segal, Prediction of gestational diabetes based on nationwide electronic health records, Nature medicine 26, 71 (2020).
- [71] D.-M. Koh, N. Papanikolaou, U. Bick, R. Illing, C. E. Kahn Jr, J. Kalpathi-Cramer, C. Matos, L. Martí-Bonmatí, A. Miles, S. K. Mun, *et al.*, Artificial intelligence and machine learning in cancer imaging, Communications Medicine 2, 133 (2022).
- [72] C. Curtis, S. P. Shah, S.-F. Chin, G. Turashvili, O. M. Rueda, M. J. Dunning, D. Speed, A. G. Lynch, S. Samarajiwa, Y. Yuan, *et al.*, The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups, Nature **486**, 346 (2012).
- [73] F. Gao, W. Wang, M. Tan, L. Zhu, Y. Zhang, E. Fessler, L. Vermeulen, and X. Wang, Deepcc: a novel deep learning-based framework for cancer molecular subtype classification, Oncogenesis 8, 44 (2019).
- [74] Y.-C. Chiu, H.-I. H. Chen, T. Zhang, S. Zhang, A. Gorthi, L.-J. Wang, Y. Huang, and Y. Chen, Predicting drug response of tumors from integrated genomic profiles by deep neural networks, BMC medical genomics 12, 143 (2019).
- [75] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, *et al.*, International evaluation of an ai system for breast cancer screening, Nature **577**, 89 (2020).
- [76] X. Liu, L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern, *et al.*, A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis, The lancet digital health 1, e271 (2019).
- [77] M. K. Siddiqui, R. Morales-Menendez, X. Huang, and N. Hussain, A review of epileptic seizure detection using machine learning classifiers, Brain informatics 7, 1 (2020).
- [78] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, nature 542, 115 (2017).
- [79] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, Automated detection of covid-19 cases using deep neural networks with x-ray

images, Computers in biology and medicine **121**, 103792 (2020).

- [80] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, Deep learning-enabled medical computer vision. npj digital medicine, 4 (1), 5 (2021).
- [81] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Q. Ni, Y. Chen, J. Su, *et al.*, A deep learning system to screen novel coronavirus disease 2019 pneumonia, Engineering 6, 1122 (2020).
- [82] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for covid-19, IEEE reviews in biomedical engineering 14, 4 (2020).
- [83] A. Narin, C. Kaya, and Z. Pamuk, Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks, Pattern Analysis and Applications 24, 1207 (2021).
- [84] L. Wang, Z. Q. Lin, and A. Wong, Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images, Scientific reports 10, 19549 (2020).
- [85] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, Recent advances and applications of machine learning in solid-state materials science, npj Computational Materials 5, 83 (2019).
- [86] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. Billinge, *et al.*, Recent advances and applications of deep learning methods in materials science, npj Computational Materials 8, 59 (2022).
- [87] R. Batra, L. Song, and R. Ramprasad, Emerging materials intelligence ecosystems propelled by machine learning, Nature Reviews Materials 6, 655 (2021).
- [88] H. Tao, T. Wu, M. Aldeghi, T. C. Wu, A. Aspuru-Guzik, and E. Kumacheva, Nanoparticle synthesis assisted by machine learning, Nature reviews materials 6, 701 (2021).
- [89] G. L. Hart, T. Mueller, C. Toher, and S. Curtarolo, Machine learning for alloys, Nature Reviews Materials 6, 730 (2021).
- [90] M. Aykol, P. Herring, and A. Anapolsky, Machine learning for continuous innovation in battery technologies, Nature Reviews Materials 5, 725 (2020).
- [91] T. DebRoy, T. Mukherjee, H. Wei, J. Elmer, and J. Milewski, Metallurgy, mechanistic models and machine learning in metal printing, Nature Reviews Materials 6, 48 (2021).
- [92] Z. Zhu, D. W. H. Ng, H. S. Park, and M. C. McAlpine, 3d-printed multifunctional materials enabled by artificial-intelligence-assisted fabrication technologies, Nature Reviews Materials 6, 27 (2021).
- [93] G. Prashun, S. Vladan, and S. Eric, Computationally guided discovery of thermoelectric materials, Nat. Rev. Mater 2, 17053 (2017).
- [94] P. G. Boyd, Y. Lee, and B. Smit, Computational development of the nanoporous materials genome, Nature Reviews Materials 2, 1 (2017).
- [95] W. Sha, Y. Li, S. Tang, J. Tian, Y. Zhao, Y. Guo, W. Zhang, X. Zhang, S. Lu, Y.-C. Cao, *et al.*, Machine learning in polymer informatics, InfoMat **3**, 353 (2021).
- [96] M. Reis, F. Gusev, N. G. Taylor, S. H. Chung, M. D. Verber, Y. Z. Lee, O. Isayev, and F. A. Leibfarth,

Machine-learning-guided discovery of 19f mri agents enabled by automated copolymer synthesis, Journal of the American Chemical Society **143**, 17677 (2021).

- [97] T. K. Patra, Data-driven methods for accelerating polymer design, ACS Polymers Au 2, 8 (2021).
- [98] L. Tao, V. Varshney, and Y. Li, Benchmarking machine learning models for polymer informatics: an example of glass transition temperature, Journal of Chemical Information and Modeling **61**, 5395 (2021).
- [99] M. M. Cencer, J. S. Moore, and R. S. Assary, Machine learning for polymeric materials: an introduction, Polymer International 71, 537 (2022).
- [100] A. J. Gormley and M. A. Webb, Machine learning in combinatorial polymer chemistry, Nature Reviews Materials 6, 642 (2021).
- [101] C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, Polymer design using genetic algorithm and machine learning, Computational Materials Science 186, 110067 (2021).
- [102] P. M. Maffettone, L. Banko, P. Cui, Y. Lysogorskiy, M. A. Little, D. Olds, A. Ludwig, and A. I. Cooper, Crystallography companion agent for high-throughput materials discovery, Nature Computational Science 1, 290 (2021).
- [103] L. Banko, P. M. Maffettone, D. Naujoks, D. Olds, and A. Ludwig, Deep learning for visualization and novelty detection in large x-ray diffraction datasets, Npj Computational Materials 7, 104 (2021).
- [104] B. D. Lee, J.-W. Lee, W. B. Park, J. Park, M.-Y. Cho, S. Pal Singh, M. Pyo, and K.-S. Sohn, Powder x-ray diffraction pattern is all you need for machine-learningbased symmetry identification and property prediction, Advanced Intelligent Systems 4, 2200042 (2022).
- [105] J. Zimmermann, F. Beguet, D. Guthruf, B. Langbehn, and D. Rupp, Finding the semantic similarity in singleparticle diffraction images using self-supervised contrastive projection learning, npj Computational Materials 9, 24 (2023).
- [106] R. Vinuesa and S. L. Brunton, Enhancing computational fluid dynamics with machine learning, Nature Computational Science 2, 358 (2022).
- [107] R. Vinuesa, S. L. Brunton, and B. J. McKeon, The transformative potential of machine learning for experiments in fluid mechanics, Nature Reviews Physics 5, 536 (2023).
- [108] M. Z. Yousif, L. Yu, S. Hoyas, R. Vinuesa, and H. Lim, A deep-learning approach for reconstructing 3d turbulent flows from 2d observation data, Scientific Reports 13, 2529 (2023).
- [109] L. Guastoni, J. Rabault, P. Schlatter, H. Azizpour, and R. Vinuesa, Deep reinforcement learning for turbulent drag reduction in channel flows, The European Physical Journal E 46, 27 (2023).
- [110] C. Vignon, J. Rabault, and R. Vinuesa, Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions, Physics of Fluids 35 (2023).
- [111] L. Yu, M. Z. Yousif, M. Zhang, S. Hoyas, R. Vinuesa, and H.-C. Lim, Three-dimensional esrgan for superresolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning, Physics of Fluids 34 (2022).
- [112] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, Machine learning–accelerated

computational fluid dynamics, Proceedings of the National Academy of Sciences **118**, e2101784118 (2021).

- [113] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, Annual review of fluid mechanics 52, 477 (2020).
- [114] J. N. Kutz, Deep learning in fluid dynamics, Journal of Fluid Mechanics 814, 1 (2017).
- [115] S. L. Brunton, Applying machine learning to study fluid mechanics, Acta Mechanica Sinica 37, 1718 (2021).
- [116] K. Fukami, K. Fukagata, and K. Taira, Assessment of supervised machine learning methods for fluid flows, Theoretical and Computational Fluid Dynamics 34, 497 (2020).
- [117] E. Bedolla, L. C. Padierna, and R. Castaneda-Priego, Machine learning for condensed matter physics, Journal of Physics: Condensed Matter 33, 053001 (2020).
- [118] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nature Physics 13, 431 (2017).
- [119] A. L. Ferguson, Machine learning and data science in soft materials engineering, Journal of Physics: Condensed Matter **30**, 043002 (2017).
- [120] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, Machine learning for active matter, Nature Machine Intelligence 2, 94 (2020).
- [121] F. Cichos, S. M. n. Landin, and R. Pradip, Artificial intelligence (ai) enhanced nanomotors and active matter, in *Intelligent Nanotechnology* (Elsevier, 2023) pp. 113–144.
- [122] I. Essafri, B. Ghosh, C. Desgranges, and J. Delhommelle, Designing, synthesizing, and modeling active fluids, Physics of Fluids 34 (2022).
- [123] R. Jadrich, B. Lindquist, and T. Truskett, Unsupervised machine learning for detection of phase transitions in off-lattice systems. i. foundations, The Journal of chemical physics 149 (2018).
- [124] R. Jadrich, B. Lindquist, W. Piñeros, D. Banerjee, and T. Truskett, Unsupervised machine learning for detection of phase transitions in off-lattice systems. ii. applications, The Journal of chemical physics 149 (2018).
- [125] M. Fränzl and F. Cichos, Active particle feedback control with a single-shot detection convolutional neural network, Scientific reports 10, 12571 (2020).
- [126] M. J. Falk, V. Alizadehyazdi, H. Jaeger, and A. Murugan, Learning to control active matter, Physical Review Research 3, 033291 (2021).
- [127] A. Argun, G. Volpe, and S. Bo, Classification, inference and segmentation of anomalous diffusion with recurrent neural networks, Journal of Physics A: Mathematical and Theoretical 54, 294003 (2021).
- [128] J. Colen, M. Han, R. Zhang, S. A. Redford, L. M. Lemma, L. Morgan, P. V. Ruijgrok, R. Adkins, Z. Bryant, Z. Dogic, *et al.*, Machine learning activenematic hydrodynamics, Proceedings of the National Academy of Sciences **118**, e2016708118 (2021).
- [129] Z. Zhou, C. Joshi, R. Liu, M. M. Norton, L. Lemma, Z. Dogic, M. F. Hagan, S. Fraden, and P. Hong, Machine learning forecasting of active nematics, Soft matter 17, 738 (2021).
- [130] A. Frishman and K. Keren, Learning active nematics one step at a time, Proceedings of the National Academy of Sciences 118, e2102169118 (2021).
- [131] S. Bag and R. Mandal, Interaction from structure using machine learning: in and out of equilibrium, Soft Matter 17, 8322 (2021).

- [132] M. Nasiri, H. Löwen, and B. Liebchen, Optimal active particle navigation meets machine learning (a), Europhysics Letters 142, 17001 (2023).
- [133] Z. Zou, Y. Liu, Y.-N. Young, O. S. Pak, and A. C. Tsang, Gait switching and targeted navigation of microswimmers via deep reinforcement learning, Communications Physics 5, 158 (2022).
- [134] Y. Liu, Z. Zou, O. S. Pak, and A. C. Tsang, Learning to cooperate for low-reynolds-number swimming: a model problem for gait coordination, Scientific Reports 13, 9397 (2023).
- [135] B. Hartl, M. Hübl, G. Kahl, and A. Zöttl, Microswimmers learning chemotaxis with genetic algorithms, Proceedings of the National Academy of Sciences 118, e2019683118 (2021).
- [136] Z. El Khiyati, R. Chesneaux, L. Giraldi, and J. Bec, Steering undulatory micro-swimmers in a fluid flow through reinforcement learning, The European Physical Journal E 46, 43 (2023).
- [137] M. Caraglio, H. Kaur, L. J. Fiderer, A. López-Incera, H. J. Briegel, T. Franosch, and G. Muñoz Gil, Learning how to find targets in the micro-world: The case of intermittent active brownian particles, arXiv preprint arXiv:2311.16692 (2023).
- [138] F. Borra, L. Biferale, M. Cencini, and A. Celani, Reinforcement learning for pursuit and evasion of microswimmers at low reynolds number, Physical Review Fluids 7, 023103 (2022).
- [139] L. Yang, J. Jiang, X. Gao, Q. Wang, Q. Dou, and L. Zhang, Autonomous environment-adaptive microrobot swarm navigation enabled by deep learning-based real-time distribution planning, Nature Machine Intelligence 4, 480 (2022).
- [140] M. A. Larchenko, P. Osinenko, G. Yaremenko, and V. V. Palyulin, A study of first-passage time minimization via q-learning in heated gridworlds, IEEE Access 9, 159349 (2021).
- [141] D. McDermott, C. Reichhardt, and C. Reichhardt, Characterizing different motility induced regimes in active matter with machine learning and noise, arXiv preprint arXiv:2308.07296 (2023).
- [142] A. R. Dulaney and J. F. Brady, Machine learning for phase behavior in active matter systems, Soft Matter 17, 6808 (2021).
- [143] M. R. Bailey, F. Grillo, and L. Isa, Tracking janus microswimmers in 3d with machine learning, Soft Matter 18, 7291 (2022).
- [144] H. Bachimanchi, M. I. Pinder, C. Robert, P. De Wit, J. Havenhand, A. Kinnby, D. Midtvedt, E. Selander, and G. Volpe, Deep-learning-powered data analysis in plankton ecology, arXiv preprint arXiv:2309.08500 (2023).
- [145] P. Galajda, J. Keymer, P. Chaikin, and R. Austin, A wall of funnels concentrates swimming bacteria, Journal of bacteriology 189, 8704 (2007).
- [146] N. Kumar, R. K. Gupta, H. Soni, S. Ramaswamy, and A. K. Sood, Trapping and sorting active particles: Motility-induced condensation and smectic defects, Phys. Rev. E 99, 032605 (2019).
- [147] H. E. Ribeiro, W. P. Ferreira, and F. Q. Potiguar, Trapping and sorting of active matter in a periodic background potential, Phys. Rev. E 101, 032126 (2020).
- [148] B.-q. Ai, Z.-g. Shao, and W.-r. Zhong, Mixing and demixing of binary mixtures of polar chiral active par-

ticles, Soft Matter 14, 4388 (2018).

- [149] V. R. Misko, L. Baraban, D. Makarov, T. Huang, P. Gelin, I. Mateizel, K. Wouters, N. De Munck, F. Nori, and W. De Malsche, Selecting active matter according to motility in an acoustofluidic setup: self-propelled particles and sperm cells, Soft Matter 19, 8635 (2023).
- [150] M. Paoluzzi, L. Angelani, and A. Puglisi, Narrow-escape time and sorting of active particles in circular domains, Phys. Rev. E 102, 042617 (2020).
- [151] J. Happel and H. Brenner, Low Reynolds Number Hydrodynamics: with special applications to particulate media (Springer Netherlands & Martinus Nijhoff Publishers, The Hague, The Netherlands, 1983).
- [152] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, Scikit-learn: Machine learning in python, the Journal of machine Learning research **12**, 2825 (2011).
- [153] S. M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17 (Curran Associates Inc., Red Hook, NY, USA, 2017) p. 4768–4777.
- [154] A. Kaiser, H. H. Wensink, and H. Löwen, How to capture active particles, Phys. Rev. Lett. 108, 268307 (2012).
- [155] B.-Q. Ai, S. Quan, and F. guo Li, Spontaneous demixing of chiral active mixtures in motility-induced phase

separation, New Journal of Physics 25, 063025 (2023).

- [156] D. Levis, I. Pagonabarraga, and B. Liebchen, Activity induced synchronization: Mutual flocking and chiral self-sorting, Phys. Rev. Res. 1, 023026 (2019).
- [157] T. Barois, J.-F. m. c. Boudet, J. S. Lintuvuori, and H. Kellay, Sorting and extraction of self-propelled chiral particles by polarized wall currents, Phys. Rev. Lett. 125, 238003 (2020).
- [158] A.-G. Niculescu, C. Chircov, A. C. Bîrcă, and A. M. Grumezescu, Fabrication and applications of microfluidic devices: A review, International Journal of Molecular Sciences 22, 2011 (2021).
- [159] S. M. Scott and Z. Ali, Fabrication methods for microfluidic devices: An overview, Micromachines 12, 319 (2021).
- [160] L. H. Duong and P.-C. Chen, Simple and low-cost production of hybrid 3D-printed microfluidic devices, Biomicrofluidics 13, 024108 (2019).
- [161] S. Waheed, J. M. Cabot, N. P. Macdonald, T. Lewis, R. M. Guijt, B. Paull, and M. C. Breadmore, 3d printed microfluidic devices: enablers and barriers, Lab on a Chip 16, 1993 (2016).
- [162] R. Amin, S. Knowlton, A. Hart, B. Yenilmez, F. Ghaderinezhad, S. Katebifar, M. Messina, A. Khademhosseini, and S. Tasoglu, 3d-printed microfluidic devices, Biofabrication 8, 022001 (2016).