Supplementary material for "Dead or alive: Distinguishing active from passive particles using supervised learning"

Giulia Janzen[1,2] [(a)], Xander L.J.A. Smeets[1] [(a)], Vincent E. Debets[1,2], Chengjie Luo[1,2], Cornelis Storm[1,2], Liesbeth M.C. Janssen[1,2] [(b)] and Simone Ciarella[1,3] [(c)]

[1] *Department of Applied Physics, Eindhoven University of Technology - P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands*
[2] *Institute for Complex Molecular Systems, Eindhoven University of Technology - P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands*
[3] *Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris - F-75005 Paris, France*

Here we provide additional results to support our findings in the main paper.

Simulation model. – Our simulation model is a Kob-Andersen mixture [1–3] in either two or three dimensions, modified to include a fraction ϕ_a of active Brownian particles. This gives the following equation of motion for each particle i [4–6]:

$$\dot{\mathbf{r}}_i = \zeta^{-1}\left(\mathbf{F}_i + \mathbf{f}_i\right) + \boldsymbol{\xi}_i. \tag{S1}$$

Here, $\mathbf{r}_i$ denotes the position of particle i, ζ the friction coefficient, and $\mathbf{F}_i$ and $\mathbf{f}_i$ the interaction and self-propulsion force acting on particle i, respectively. Moreover, $\boldsymbol{\xi}_i$ represents a Gaussian noise with zero mean and variance $\langle\boldsymbol{\xi}_i(t)\boldsymbol{\xi}_j(t')\rangle_{\text{noise}} = 2k_BT\zeta^{-1}\mathbf{I}\delta_{ij}\delta(t-t')$, with k_BT the thermal energy, T the temperature, k_B the Boltzmann constant (we set $k_B = 1$), t the time, and $\mathbf{I}$ the 3×3 unit matrix. The total interaction force on particle i (of type $\alpha = \mathrm{A}, \mathrm{B}$) due to all other particles j (of type $\beta = \mathrm{A}, \mathrm{B}$) is $\mathbf{F}_i = -\sum_{j\neq i}\nabla_i V_{\alpha\beta}(r_{ij})$, where $r_{ij} = |\boldsymbol{r}_{ij}| = |\boldsymbol{r}_j - \boldsymbol{r}_i|$ is the radial distance between particles i and j. We use a standard Lennard-Jones interaction potential

$$V_{\alpha\beta}(r) = \begin{cases} 4\epsilon_{\alpha\beta}\left[\left(\frac{\sigma_{\alpha\beta}}{r}\right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r}\right)^{6} + C_{\alpha\beta}\right] , & r \leq r^c_{\alpha\beta} , \\ 0 , & r > r^c_{\alpha\beta} , \end{cases} \tag{S2}$$

with parameters $\epsilon_{\mathrm{AA}} = 1$, $\epsilon_{\mathrm{AB}} = 1.5$, $\epsilon_{\mathrm{BB}} = 0.5$, $\sigma_{\mathrm{AA}} = 1$, $\sigma_{\mathrm{AB}} = 0.8$, and $\sigma_{\mathrm{BB}} = 0.88$. The constant $C_{\alpha\beta}$ fixes the potential to zero at the cutoff radius $r^c_{\alpha\beta} = 2.5\sigma_{\alpha\beta}$. Setting the temperature to $T = 0.6$, the bulk density to $\rho = 1.2$, and the friction coefficient to $\zeta = 1$, we can achieve high-density conditions without any crystallization effects [1–3].

For the self-propulsion force, we use the active Brownian particle (ABP) model [7–10] and let the absolute value of the force F_{a} remain constant in time so that $\mathbf{f}_i = F_{\mathrm{a}}\mathbf{e}_i$. For the passive particles $F_{\mathrm{a}} = 0$. The orientation of the force $\mathbf{e}_i$ undergoes rotational diffusion with a diffusion coefficient D_{r}. This yields [4, 6]

$$\dot{\mathbf{e}}_i = \boldsymbol{\chi}_i \times \mathbf{e}_i, \tag{S3}$$

with $\boldsymbol{\chi}_i$ denoting a Gaussian noise process with zero mean and variance $\langle\boldsymbol{\chi}_i(t)\boldsymbol{\chi}_j(t')\rangle_{\text{noise}} = 2D_{\mathrm{r}}\mathbf{I}\delta_{ij}\delta(t-t')$.

Simulations are performed by solving the Langevin equation [Eq. (S1)] via a forward Euler scheme using LAMMPS [11]. We include periodic boundary conditions in a cubic box and run the simulation sufficiently long (on the order of 200 time units) to ensure that the system has reached steady-state conditions. Note that all results are presented in reduced units where σ_{AA}, ϵ_{AA}, $\epsilon_{\mathrm{AA}}/k_{\mathrm{B}}$, and $\zeta\sigma^2_{\mathrm{AA}}/\epsilon_{\mathrm{AA}}$ represent the units of length, energy, temperature, and time, respectively [3]. Afterward, we save the configuration of the particles every 20 time units. This time interval is an order of magnitude larger than the relaxation time of the fully passive reference system and therefore allows each configuration to be considered statistically independent of the previously saved one [12]. In total, we retrieve 2000 different independent configurations for each studied setting.

Gradient Boosting model versus Multilayer perceptron. – In this section, we compare the performance of the Gradient Boosting model (GB), as discussed in the

[(a)] These authors contributed equally to this work.
[(b)] E-mail: l.m.c.janssen@tue.nl
[(c)] E-mail: simoneciarella@gmail.com

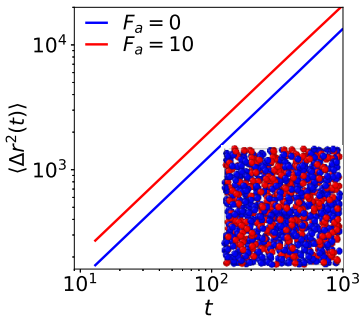

FIG. S1: Mean-squared displacements as a function of time t for passive particles ($F_a = 0$) and active particles with $F_a = 10$. The inset shows a snapshot of the system, where active particles are represented in red and passive particles in blue.

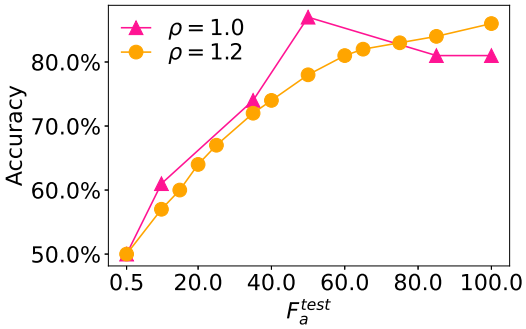

FIG. S2: Model accuracy as a function of the active force $F_a^{\text{test}} = F_a^{\text{train}}$, with $\phi_a = 0.008$. The pink triangles and the orange circles represent scores obtained for a system with density $\rho = 1.0$ and $\rho = 1.2$, respectively.

main text, with a Multilayer Perceptron (MLP). In fact our static models (both Voronoi and shell-based), and even the pseudo-static model, can be implemented with either a MLP or a GB to perform the classification task. Here we want to motivate the choice of GB that we make in the main manuscript.

In general, a MLP consists of multiple layers of interconnected neurons. Specifically, we consider an MLP with two hidden layers, consisting of 50 and 30 nodes, respectively. Even though this particular architecture has been selected after performing a coarse grid search, it is still possible that a deeper architecture may achieve better performance, at the cost of an increased computational cost. In terms of accuracy, we have verified that the MLP is comparable to the Gradient Boosting model. However, it is important to notice that the computational time required for the MLP is significantly longer compared to the LightGBM model, without even including the cost associated to the grid search in designing the architecture of the MLP. In detail, the ratio of computational times between the LightGBM model and the MLP is approximately 0.052, so using a predefined GB is at least 20 times faster than an optimized MLP while achieving the same accuracy. Furthermore, it should be highlighted that using a different MLP can lead to remarkably worse performance. For these reasons, the results presented in the main text correspond to the LightGBM model. Moreover, since in Ref. [13] a linear regression model has been successfully used to predict the propensity in purely passive systems here we attempted to train a similar model using both the Voronoi and the shell-based approach as input. However, this model did not accurately predict whether a particle is active or passive, achieving an accuracy of only 0.5. Therefore, we can conclude that for this classification problem, a more advanced machine learning model such as LightGBM is required.

Mean-squared displacement. – Here we compute the mean squared-displacement

$$\langle \Delta r^2(t) \rangle = \langle \frac{1}{N} \sum_{i=1}^{N} |\mathbf{r}_i(t) - \mathbf{r}_i(t_0)|^2 \rangle \qquad \text{(S4)}$$

where $\mathbf{r}_i = (x_i, y_i, z_i)$ represents the particle's spatial coordinates, N is the number of particles, t is time, t_0 represents the time origin and $\langle \dots \rangle$ denotes the average over different time origins. Figure S1 shows the mean-squared displacements for passive particles ($F_a = 0$) and active particles with $F_a = 10$, which confirms that active particles have faster dynamics compared to passive particles [14, 15]. The inset of Fig. S1 shows a snapshot of the system, which is the input for the static approach that has been explained in the main text.

Voronoi-based model for lower density. – Figure S2 illustrates a comparison of the model accuracy using a Voronoi-based approach for two different system densities: $\rho = 1.0$ and $\rho = 1.2$, both with a fraction of active particles $\phi_a = 0.008$. The results reveal that the system with lower density exhibits higher performance compared to the denser system, particularly for $F_a \leq 50$. This indicates that at lower activities, the spatial inhomogeneity caused by active particles is more pronounced at a lower density. However, at higher activities, the model still demonstrates good performance in distinguishing between active and passive particles, albeit slightly worse than in the denser system.

Voronoi-based model in 3D. – In this section, we present the results obtained by applying the Voronoi model to a three-dimensional system. Figure S3 illustrates the accuracy achieved by the Voronoi active/passive classifiers for three values of the active particle fraction ϕ_a, while keeping F_a fixed at 100. Consistent with the findings in the two-dimensional case discussed in the main text, when $\phi_a < 0.1$, the static approach exhibits an excellent predictive capability when the model is trained and tested at a specific ϕ_a value (indicated by orange circles). Similarly, the radial approach demonstrates high accuracy in scenarios with high activity and a low fraction of active particles.

Model explanation: SHAP analysis. In addition to the features discussed in the main text for the two-dimensional system, the three-dimensional mixtures requires the addition of the following descriptors: Volume, minimum and

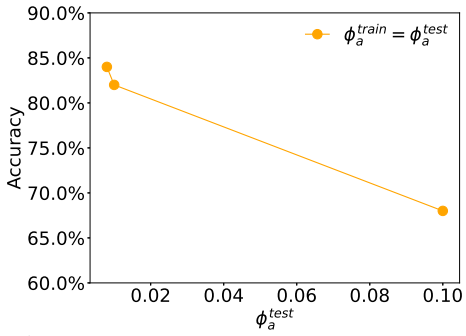

FIG. S3: Accuracy of the Voronoi model in 3D as a function of the fraction of active particles $\phi_a = \phi_a^{\text{test}}$, with $F_a = 100$. The orange circles represent scores obtained from separate models, where each one was trained using $\phi_a^{\text{train}} = \phi_a^{\text{test}}$.

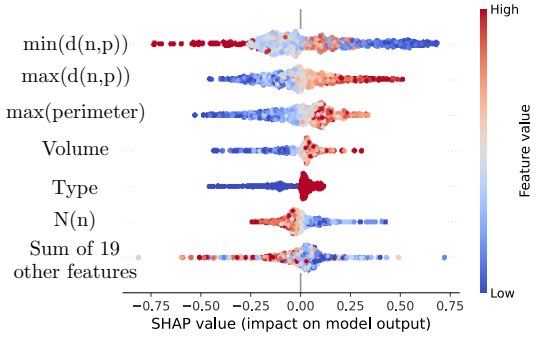

FIG. S4: Feature importance for the 3D Voronoi model, represented as a SHAP beeswarm plot at $F_a = 100$ and $\phi_a = 0.008$. The position of the dots is determined by the SHAP values of the features, and the color is used to display the value of the features. The top six most important features include the minimum distance between the particle and its neighbors min(d(n,p)), the maximum distance between the particle and its neighbors max(d(n,p)), face with the maximum perimeter max(perimeter), volume, particle type (A or B), and the number of neighbors N(n).

maximum perimeter, and the number of faces with a vertex count ranging from 3 to 10. Figure S4 shows the SHAP beeswarm plot, which indicates the six most important features and how the values of these features influence the model's predictions. The first six most important features are: the minimum distance between the particle and its neighbors min(d(n,p)), the maximum distance between the particle and its neighbors max(d(n,p)), face with the maximum perimeter (max(perimeter)), volume, particle type (A or B), and the number of neighbors N(n). In Fig. S4, the colors representing the two most important features, min(d(n,p)) and max(d(n,p)), indicate that the model interprets low and high values of these features, respectively, as indicative of active particles, and high and low values, respectively, as indicative of passive particles. As passive particles tend to accumulate in front of active particles while creating voids behind them, we expect that active particles will be closer to the passive particles in the front and farther away from the passive particles in the back. The SHAP analysis confirms this expectation, as it demonstrates that the distance between the particle and its neighbors effectively captures this anisotropic phenomenon.

Pseudo-static approach. –

Model definition. Here we introduce a modified ML approach that can correctly classify the active and passive particles in the regime where the purely static approach fails. This so-called pseudo-static method uses multiple *unordered* snapshots of the system as input (Fig. S5). For this we employ a multilayer perceptron neural network consisting of one hidden layer of 200 neurons. The configuration of our neural network was optimized by performing a grid search; i.e. a scan over the hyperparameter space in order to select the ones providing the best performance. The optimal hyperparameters that we used in the final model are reported in Table S1. In the end, after calculating all the relevant structural input features, the training of our model takes only several minutes on a standard laptop.

As input features for the pseudo-static approach, we include measures for the *fluctuations* of particle-resolved local structural features. In principle we could consider fluctuations of e.g. our Voronoi features or the shell-based descriptors, but here we choose effectively a combination of them, namely Voronoi weighted order parameters. These order parameters have previously also been shown to describe local crystal structures, e.g., cubic ($q_{4,6}$), BCC (q_8), and FCC (q_{12}) [19], and they are faster to calculate and easier to interpret than those of the shell-based approach.Explicitly, we use the l-th Voronoi weighted order parameters [19]

$$Q_l(i) = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} \left| \sum_{f \in \mathcal{F}(i)} \frac{A(f)}{A(i)} Y_l^m(\theta_f, \phi_f) \right|^2 \right)^{\frac{1}{2}}, \quad \text{(S5)}$$

where the inner sum is taken over all shared boundaries (facets) f of the Voronoi cell containing particle i, $A(f)$ is the area of the facet, $A(i)$ is the total area of the Voronoi cell, and θ_f and ϕ_f are the spherical angles of the outer normal vector of the facet f. In addition, we use the lth averaged Voronoi weighted order parameters

$$\bar{Q}_l(i) = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} \left| \frac{1}{\tilde{N}(i)} \sum_{k=0}^{\tilde{N}(i)} \sum_{f \in \mathcal{F}(k)} \frac{A(f)}{A} Y_{lm}(\theta_f, \phi_f) \right|^2 \right)^{\frac{1}{2}} \quad \text{(S6)}$$

where the middle sum is taken over all the $\tilde{N}$ neighbors of the particle i, including particle i itself. The difference between the parameters is that those in Eq. (S5) only describe the first shell of particles around particle i, while the parameters in Eq. (S6) also describe the second shell of particles around particle i.

Importantly, since these are stochastic quantities, we perform one additional processing step and calculate, for each particle i, the mean, median, minimum, maximum,

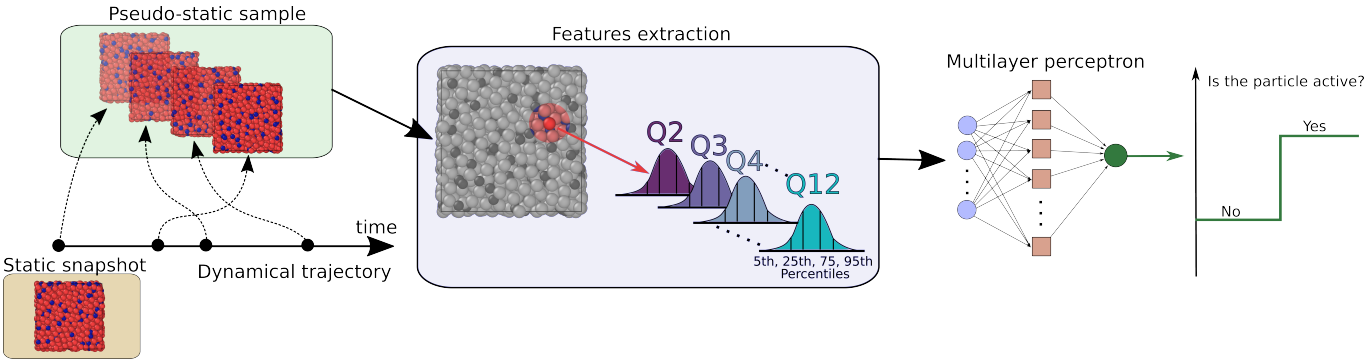

FIG. S5: Sketch of our pseudo-static approach for the identification of active particles in an active/passive mixture.

Parameter	Value
solver	'lbfgs'
activation	'tanh'
alpha	0.05
max_iter	10 ** 9
max_fun	10**5
validation_fraction	0.1
early_stopping	True
hidden_layer_sizes	(200,)
n_iter_no_change	10
tol	0.0001

Table S1: The hyperparameters used for the multilayer perceptron. The parameter names refer to the inputs for the multilayer perceptron implementation provided by the scikit-learn [16] library.

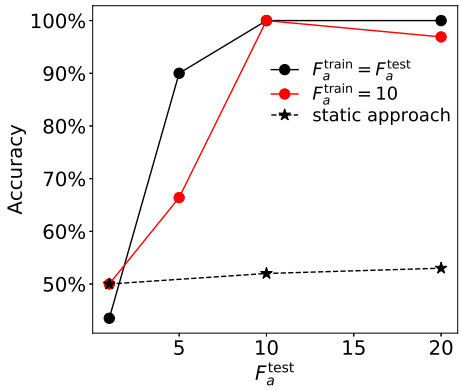

FIG. S6: Model accuracy for our pseudo-static approach as a function of the active force $F_a = F_a^{\text{test}}$, with $\phi_a = 0.5$. The black points represent scores obtained from separate models, where each one was trained using $F_a^{\text{train}} = F_a^{\text{test}}$. The black stars correspond to the score obtained from separate models trained with the features computed with the static shell-based approach. The red points represent scores obtained from a single global model trained with data from $F_a = 10$.

and standard deviation corresponding to the distribution of $Q_l(i)$ averaged over the number of configurations in the training set. To capture more static information, we also calculate the 5th, 25th, 75th, and 95th percentile of these distributions. It will be these statistics, which characterize the distribution of $Q_l(i)$ with $l = 2, \ldots, 12$, as well as the particle type (A or B), that are used as input for the pseudo-static ML model. In total, this model considers 199 input quantities; note that this feature space is significantly smaller than that of the shell-based approach, but larger than that of the static Voronoi approach.

To quantify the network's performance, we use two indicators: (i) accuracy and (ii) f1-score. Accuracy is the most basic metric for classification, defined as the number of correct predictions divided by the total number of predictions. This metric, however, is not optimal when the classes are unbalanced, which is the case here when the fraction of active particles $\phi_a \neq 0.5$. (Note that in the main text we have always enforced a balanced data set by discarding particles if needed, but for the pseudo-static approach we cannot afford this in view of the required data quality.) Hence, when we have a different number of active and passive particles we evaluate the model with the f1-score

$$\text{f1-score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

where the precision is the sum of true positives across all classes divided by the sum of both true and false positives over all classes, and the recall is the sum of true positives across all classes divided by the sum of true and false negatives across all classes. The f1-score reaches its largest value of 1 when the model has perfect precision and recall, and its lowest value of 0 if either the precision or the recall is equal to zero.

We implement this method in our three-dimensional simulation system at a temperature of $T = 0.6$. The configurations of the particles are saved every 13th time unit. This time interval is chosen to be significantly larger than the relaxation time of the fully passive reference system. As a result, each saved configuration is considered to be statistically independent of the previously saved one. The statistics are computed over 10 000 independent configu-

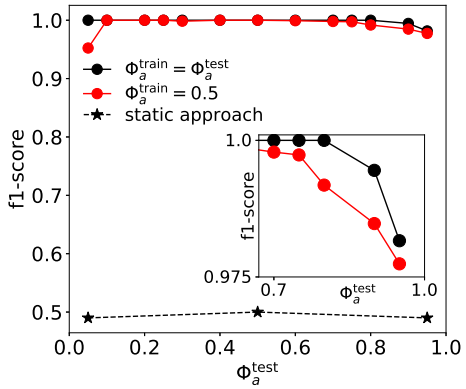

FIG. S7: The f1-score as a function of the fraction of active particles $\phi_a = \phi_a^{\text{test}}$, with $F_a = 10$. The black points represent scores obtained from separate models, where each model was trained using $\phi_a^{\text{train}} = \phi_a^{\text{test}}$. The black stars correspond to the score obtained from separate models trained with the features computed with the static shell-based approach. The red points represent scores obtained from a single model trained with data from $\phi_a = 0.5$. In the inset, we highlight the small score drop that happens for $\phi_a^{\text{test}} \to 1$.

rations.

Results for the pseudo-static approach. Figure S6 reports the accuracy that our pseudo-static ML model achieves in the classification of a 50/50 active/passive mixture, for different values of the active force F_a. We compare the accuracy that we get when the model is trained and tested at a single specific value of F_a (black circles), with the accuracy of a model trained when fixing $F_a = 10$ (red circles). When the active force F_a is very small, it is difficult to distinguish between motion related to active forces and passive Brownian motion, so the pseudo-static approach fails as much as the static approach, once again confirming that the statics-dynamics connection is extremely subtle for active systems in this regime.

For $F_a \geq 5$ the pseudo-static approach achieves $> 90\%$ accuracy, clearly outperforming any purely static method that takes only instantaneous structural information as input. As expected, we also see that the accuracy of the pseudo-static approach increases when the activity is strong, because the difference between passive and active particles becomes more significant. Furthermore, a single model trained at an intermediate value of $F_a = 10$ is able to produce good predictions for $F_a > F_a^{\text{train}}$, thus showing reasonable generalizability to unseen parameter regimes, although the accuracy gets lower for $F_a < F_a^{\text{train}}$. In Fig. S6 the dashed line shows results corresponding to the shell-based approach (similar results can be obtained with the Voronoi model). Consistent with the findings discussed in the main text, the static approach fails to achieve even 60% accuracy for high fractions of active particles and low activity. This suggests that activity does not leave a clear, simple signature in the instantaneous local structure.

In Figure S7 we evaluate the performance of the active/passive classifiers as a function of the percentage of active particles ϕ_a at fixed $F_a = 10$. Here the pseudo-static approach achieves very good predictive power, quantified by the very large f1-score of ~ 1, even when the model is trained only at $\phi_a = 0.5$ (red circles). Thus, the model also generalizes well to other active/passive stoichiometries. Once again, a static approach (dashed line) is not effective for any value of ϕ_a. When the fraction of active particles approaches 1, we highlight in the inset that the model becomes slightly less accurate, though the f1-score still remains above 0.97. Our interpretation of this small score drop is that it is easier to identify a single particle that is moving due to activity (small ϕ_a, black circles) rather than identifying a single passive particle with many active neighbors (large ϕ_a), since the activity of the neighbors usually disrupts the local environment.

Model explanation. Having thus established an effective pseudo-static ML model that can accurately distinguish active from passive particles, let us now seek to gain more insight into the decisions made by the model. That is, rather than using it as a black box [20,21] we calculate the model *explanations*. For this we use LIME [17,18]. In brief, these explanations describe the correlation between a given input parameter and the probability of predicting a given particle to be active and can be considered local approximations to the model. The correlation values given by the explanations are used to assess the relevance of certain parameters to the overall predictions made by the model; the stronger the deviation from 0, the more relevant a feature is for determining whether a particle is active or not.

Figure S8 presents an aggregation of the explanations over all particles in an active/passive mixture with $\phi_a = 0.5$ and $F_a = 10$. The boxplots in Fig. S8(a) show the distribution of the correlation between a given parameter and the probability that a given particle is predicted to be active by the model. For clarity, we show only the 23 most important features, and we note that the ordering may vary somewhat depending on the initialization settings of the LIME algorithm. While there is no single dominant feature, overall we see that the features related to Q_5 (e.g. its mean and its 50th, 75th, and 95th percentiles) are relatively strongly correlated to the model prediction. In Fig. S8(b) we average all the statistical features related to the same physical Q_i observable, also confirming that Q_5 is a relatively important structural property. This finding is consistent with recent work on densely disordered passive Lennard-Jones particles, which found that Q_5 produces the largest contribution in a principal component analysis [22]. Thus, local 5-fold symmetries constitute relevant descriptors of passive particles, and if we incorporate their fluctuations using our pseudo-static technique, they also provide informative structural signatures of active particles in a disordered mixture. However, our results indicate that active/passive mixtures generally have a broad spec-

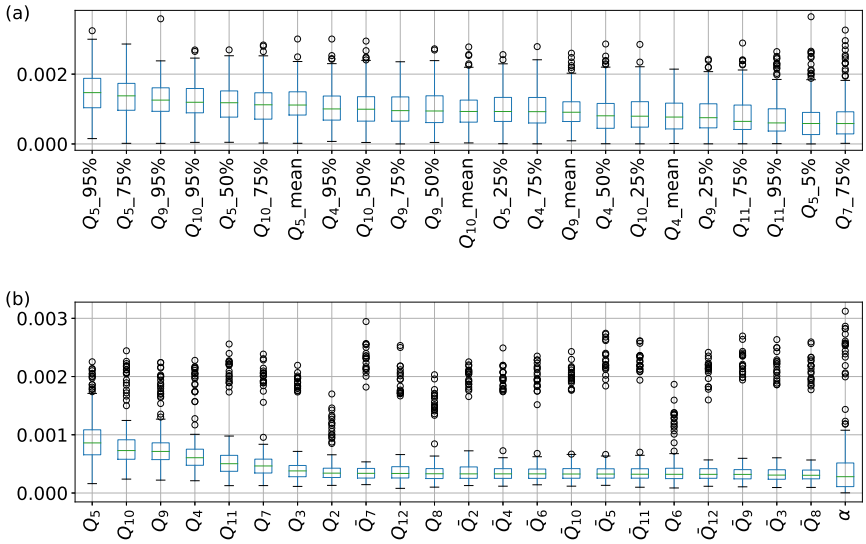

FIG. S8: Importance of the structural features in the identification of active and passive particles, as determined by LIME [17, 18]. The parameters are ordered according to their mean importance. In (a) we report the 23 most important features, while in (b) we average the features relative to the same physical observable [Eq. (S5) or (S6)]. In (a) the notation `Qn_m%` refers to the mth percentile of the distribution of Q_n, with `Qn_50%` the median. The data was produced from the pseudo-static ML model trained for a mixture with $\Phi_a = 0.5$ and $F_a = 10$.

trum of structural features with no single dominant signal, and hence we conclude that active particles do not assume well-defined local structures, even at relatively small activity. This lack of a single, unique feature underlying the structure-dynamics relation is also similar to the case of fully passive disordered systems [22, 23].

Lastly, we notice the importance of the particle type α, i.e. the species label A or B in the Kob-Andersen mixture [the last feature listed in Fig. S8(b)]. While on average the particle type is weakly correlated to the identification of active particles, there are some significant outliers (black circles) with a strong correlation. Our interpretation is that consistently with passive Kob-Andersen mixtures [24], type-B particles are smaller, which increases their mobility. However, type-B particles only constitute 20% of our mixture [25]. We hypothesize that the model implicitly knows that the particle identity is not important for 80% of the cases (the majority of large particles), while it is significant to differentiate the 20% of small particles.

References

[1] Kob W. and Andersen H. C., *Phys. Rev. Lett.*, **73** (1994) 1376.

[2] Michele C. D., Sciortino F. and Coniglio A., *Journal of Physics: Condensed Matter*, **16** (2004) L489.

[3] Flenner E. and Szamel G., *Phys. Rev. E*, **72** (2005) 1.

[4] Farage T. F. F., Krinninger P. and Brader J. M., *Phys. Rev. E*, **91** (2015) 042310.

[5] Ni R., Stuart M. A. C. and Dijkstra M., *Nature Communications*, **4** (2013) 2704.

[6] Feng M. and Hou Z., *Soft Matter*, **13** (2017) 4464.

[7] Romanczuk P., Bär M., Ebeling W., Lindner B. and Schimansky-Geier L., *The European Physical Journal Special Topics*, **202** (2012) 1.

[8] Ramaswamy S., *Journal of Statistical Mechanics: Theory and Experiment*, **2017** (2017) 054002.

[9] Löwen H., *The Journal of Chemical Physics*, **152** (2020) 040901.

[10] ten Hagen B., van Teeffelen S. and LÃ¶wen H., *Journal of Physics: Condensed Matter*, **23** (2011) 194119.

[11] Plimpton S., *Journal of Computational Physics*, **117** (1995) 1.

[12] Flenner E. and Szamel G., *Nature Communications*, **6** (2015) 7392.

[13] Boattini E., Smallenburg F. and Filion L., *Physical Review Letters*, **127** (2021) 88007.

[14] Liluashvili A., Ónody J. and Voigtmann T., *Phys. Rev. E*, **96** (2017) 062608.

[15] Bechinger C., Di Leonardo R., Löwen H., Reichhardt C., Volpe G. and Volpe G., *Rev. Mod. Phys.*, **88** (2016) 045006.

[16] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. and Duchesnay E., *Journal of Machine Learning Research*, **12** (2011) 2825.

[17] Ribeiro M. T., Singh S. and Guestrin C., in proc. of *Proceedings of the 22nd ACM SIGKDD* 2016 pp. 1135–1144.

[18] Ribeiro M. T., Singh S. and Guestrin C., *arXiv preprint arXiv:1602.04938*, (2016) .

[19] Mickel W., Kapfer S. C., SchrÃ¶der-Turk G. E. and Mecke K., *The Journal of Chemical Physics*, **138** (2013) 044501.

[20] Rudin C., *Nature Machine Intelligence*, **1** (2019) 206.
[21] Murdoch W. J., Singh C., Kumbier K., Abbasi-Asl R. and Yu B., *Proceedings of the National Academy of Sciences*, **116** (2019) 22071.
[22] Coslovich D., Jack R. L. and Paret J., *The Journal of Chemical Physics*, **157** (2022) 204503.
[23] Boattini E., Marín-Aguilar S., Mitra S., Foffi G., Smallenburg F. and Filion L., *Nature Communications*, **11** (2020) 5479.
[24] Hocky G. M., Coslovich D., Ikeda A. and Reichman D. R., *Phys. Rev. Lett.*, **113** (2014) 157801.
[25] Kob W., , **6** (1997) 1.