Chaos

Disentangling intrinsic motion from neighborhood effects in heterogeneous collective motion

Cite as: Chaos **32**, 063119 (2022); https://doi.org/10.1063/5.0093682
Submitted: 30 March 2022 • Accepted: 17 May 2022 • Published Online: 10 June 2022

Arshed Nabeel and Danny Raj Masila

Disentangling intrinsic motion from neighborhood effects in heterogeneous collective motion

Cite as: Chaos **32**, 063119 (2022); doi: 10.1063/5.0093682
Submitted: 30 March 2022 · Accepted: 17 May 2022 ·
Published Online: 10 June 2022

Arshed Nabeel[1,2] and Danny Raj Masila[2,a)]

AFFILIATIONS

[1]Center for Ecological Sciences, Indian Institute of Science, Bengaluru, India
[2]Department of Chemical Engineering, Indian Institute of Science, Bengaluru, India

[a)]**Author to whom correspondence should be addressed:** dannym@iisc.ac.in

ABSTRACT

Most real-world collectives, including animal groups, pedestrian crowds, active particles, and living cells, are heterogeneous. The differences among individuals in their intrinsic properties have emergent effects at the group level. It is often of interest to infer how the intrinsic properties differ among the individuals based on their observed movement patterns. However, the true individual properties may be masked by the nonlinear interactions in the collective. We investigate the inference problem in the context of a bidisperse collective with two types of agents, where the goal is to observe the motion of the collective and classify the agents according to their types. Since collective effects, such as jamming and clustering, affect individual motion, the information in an agent's own movement is insufficient for accurate classification. A simple observer algorithm, based only on individual velocities, cannot accurately estimate the level of heterogeneity of the system and often misclassifies agents. We propose a novel approach to the classification problem, where collective effects on an agent's motion are explicitly accounted for. We use insights about the phenomenology of collective motion to quantify the effect of the neighborhood on an agent's motion using a *neighborhood parameter*. Such an approach can distinguish between agents of two types, even when their observed motion is identical. This approach estimates the level of heterogeneity much more accurately and achieves significant improvements in classification. Our results demonstrate that explicitly accounting for neighborhood effects is often necessary to correctly infer intrinsic properties of individuals.

In collective systems, such as animal groups, human crowds, biological cells, or active particles, simple rules of interaction between the individuals often give rise to complex dynamics at the level of the collective. Usually, the individuals that make up the collective are heterogeneous, with different movement characteristics. For example, an animal group may consist of animals of multiple species, migrating cells may have a hierarchy of leader–follower relationships, or a pedestrian crowd may contain individuals with different speeds. It is often important to identify these intrinsic characteristics of the individuals that make up the collective: for example, in a pedestrian crowd, timely identification of slow-moving or trapped individuals is crucial for stampede prevention. We tackle this problem in the context of a model collective system, consisting of two groups of agents with different heading directions, and attempt to identify the desired directions of motion of the individuals. We demonstrate that the observed movement of an individual does not always contain enough information to identify its intrinsic characteristics: this information is usually lost due to complex non-linear interactions between the individual and its neighbors. We devise an approach that explicitly accounts for interactions with the neighborhood and quantifies how much the neighborhood of an agent aids or hinders its movement. This allows us to accurately infer the intrinsic directions of agents. The approach we present is based on a phenomenological understanding of collective movement, making this a *data-agnostic* approach, i.e., one that does not rely on large amounts of training-data. Approaches like this are of particular relevance to real-world problems where clean, labeled data are hard to obtain.

I. INTRODUCTION

With the advent of sophisticated imaging techniques and machine learning algorithms, many experimental studies in the field of complex systems and complex flows turn to computer vision to investigate the underlying dynamics of the individuals (agents). Examples include flocking behavior of social organisms,[1–4] dynamics of traffic flows[5] and crowds,[6–8] cell migration,[9–12] motion of synthetic active particles,[13,14] and droplet[15–17] and granular flows.[18,19] The agents that make up the collective are tracked, and their velocities are computed from the measurements. The motion of these agents is a result of both self-propulsion (or external driving) and interactions between the agents. Generally, these interactions are highly nonlinear and give rise to complex emergent dynamics. One of the primary goals in collective motion research is to understand the relationship between the observed dynamics, the intrinsic motion, and the interaction effects of the agents in the collective.

Most studies of collective phenomena assume that the individual agents in a collective are identical. However, in real-world systems, this is seldom the case. For example, in animal groups, heterogeneity can arise due to the differences in age, sex, or behavioral tendencies of the individuals,[20–22] all of which can reflect in the movement patterns of the individuals. Animals can also form multi-species flocks where each species may have different movement properties.[23,24] In pedestrian dynamics, people in the crowd with different mobility or with different destinations could significantly alter the dynamics of the crowd.[25,26] In cell migration, one finds heterogeneity to arise dynamically in the form of leader–follower states that dictate the migration dynamics of these cells, especially during processes, such as wound closure.[27] In material-collectives, such as droplets driven through a microchannel, differences in sizes could modify the level of confinement, which alters the dynamics significantly.[17] In the case of Janus particles, the inhomogeneity in the gold coating on these particles could produce gravitational torques that affect how they cluster.[13] In all these systems, it is usually of interest to identify the individual differences among agents based on their movement information and understand the impact on collective motion.

However, the question of how well one can make such inferences, and how the collective dynamics affect the inferences, is relatively unexplored, as the collective dynamics of the system depend on the heterogeneity in interesting ways. Schumacher *et al.*,[28] in the context of cell migration, demonstrate that it is hard to quantify the actual heterogeneity in a system: higher inter-agent interactions could exaggerate the true level of heterogeneity, while confinement can do the opposite.

The problem of inferring properties of a heterogeneous collective can be posed at three different levels:

1. *Estimating the level of heterogeneity:* At the coarsest level, one may need to estimate some measure of the level of heterogeneity in the collective and characterize the system on a scale ranging from fully homogeneous to highly heterogeneous.
2. *Characterizing individual agents:* A harder problem would be to characterize individual agents according to their properties. For example, it might be possible to classify individuals into different types according to some intrinsic characteristics.
3. *Recovering intrinsic properties and governing dynamics:* At the highest level, one would wish to recover the full intrinsic properties of the individual agents or reconstruct the governing equations of motion for the individuals. This is a significantly harder problem.

To investigate these questions, we turn to a simple heterogeneous collective: a *bidisperse collective*, which consists of two distinct groups of agents. The agent properties are set to be identical within a group but are distinct between groups. Although simple in principle, such bidisperse collectives come up in a variety of fields, e.g., migrating cells,[29,30] oppositely driven systems, such as charged colloids or dusty plasmas,[31–33] pedestrian crowds,[34–36] and mixed-species animal groups.[23,24]

We approach the problem of inferring a heterogeneous collective, by setting up an observer that classifies agents into their groups, based on their movement information. Depending on the context, fast and accurate classification can be very crucial. For example, while working with densely packed agents, it may be very hard to retain the tracking for long time windows, and in problems, such as predicting the onset of a stampede, timely classification is critical in employing prevention strategies. The classification problem becomes particularly challenging when clean, labeled data are not readily available—as is often the case in real-world scenarios. In these situations, we cannot rely on the traditional supervised-learning techniques[37] for classification. A physics-inspired approach, based on phenomenological understanding of the collective dynamics, is appealing in this context. By incorporating insights about the dynamics, such an approach can reduce the dependence on labeled data.

We begin with a description of a model for a bidisperse collective and show how, despite being simple, it gives rise to a wide range of collective phenomena, which are interesting from the context of inferring the heterogeneity of the collective. We first examine the performance of a simple observer that takes into account only the dynamics of a focal agent, but it is reasonable to expect that the information that aids the classification of a particular agent is encoded in the motion of that agent itself. However, since the dynamics of an agent is also driven by collective effects arising from non-linear interactions with other agents, this approach often produces misclassifications. We address this problem by explicitly accounting for the collective effects on the movement of an individual. Based on the phenomenology of collective motion, we derive a *neighborhood parameter* that quantifies the influence of neighbors on the movement of the focal agent and classify agents based on how the agent's velocity compares to the neighborhood parameter. This enables us to distinguish between agents of different groups, even when their observed movement is identical. Our analysis presents a physics-inspired approach to the inference problem, instead of a supervised-learning approach, which is reliant on the availability of labeled training-data.

II. MODEL AND DYNAMICS

A. A simple model of a bidisperse collective

We use a simple model of bidisperse collectives (i.e., consisting of two distinct groups of agents), similar to the models previously

used in the literature.[38,39] Circular agents with radius R are arranged in a 2D periodic domain, with a packing density ρ. Each agent has a desired direction of movement, which is the same for all agents in the group but different between the groups (which we call Group 1 and Group 2). Group 1 agents have a desired velocity along the positive x-direction and Group 2 agents in the negative x-direction. An agent is driven by two forces: a *restitution force*, denoting the intrinsic effort by the agent to move in the desired direction, and an *inter-agent force*, denoting the interactions between agents. The forces are designed such that the agents cannot overlap.

The following equations govern the dynamics of the agents:

$$m\frac{d\mathbf{v}_i}{dt} = \frac{m}{\tau}(\mathbf{v}_{0,i} - \mathbf{v}_i) + \sum_{\forall j \neq i} \mathbf{F}_{ij}, \tag{1}$$

$$\mathbf{v}_{0,i} = \begin{cases} +s_0\mathbf{e}_x & i \in \text{Group 1}, \\ -s_0\mathbf{e}_x & i \in \text{Group 2}, \end{cases} \tag{2}$$

$$\mathbf{F}_{ij} = \begin{cases} -\gamma\,(d_{ij} - 2R)^{-3}\hat{\mathbf{d}}_{ij} & d_{ij} < l_{cr}, \\ 0 & \text{otherwise}. \end{cases} \tag{3}$$

The intrinsic velocity $\mathbf{v}_{0,i}$ has a magnitude s_0 and is directed along the positive or negative x-direction depending on the group of the individual [see Eq. (2)]. This is the source of heterogeneity in the model. The inter-individual force $\mathbf{F}_{ij}$ is a repulsive force and decays as a power-law with distance. To prevent collisions, $\mathbf{F}_{ij}$ is chosen such that its magnitude blows up to infinity when the agent boundaries touch ($d_{ij} = 2R$), where d_{ij} is the distance between the centers of agents i and j. To avoid spurious interactions between far-off agents, $\mathbf{F}_{ij}$ is set to 0 beyond a cut-off radius, $l_{cr} = 3R$ [see Eq. (3)]. m is the mass of an individual agent (set to 1), τ is the inertial timescale of the system (set to 0.2), and γ determines the strength of the inter-agent interactions (set to 0.2). $\mathbf{e}_x$ denotes the unit vector along the x-direction. We define the packing density ρ as the number of agents per unit area.

Key parameters that determine the collective dynamics include the packing density ρ, which we vary between $[0.22, 0.58]$, the intrinsic speed s_0 between $[0.1, 3]$ and the number ratio, the fraction of agents in the minority, and N_r between $[1/42, 1/2]$, which quantifies the degree of heterogeneity in the collective. We perform simulations for each combination of parameters $\{\rho, s_0, N_r\}$. Since phenomena observed in bidisperse collectives, such as laning, clustering, jamming, etc., depend upon the initial conditions of the simulations, we perform 100 simulations for each set of parameter values where agent positions and their group-identities are assigned randomly.

B. Model dynamics

In the absence of any obstacles or collisions, an agent approaches its steady-state velocity $\mathbf{v}_{0,i}$ with a timescale τ. However, when there are other agents present, the inter-agent interactions affect the movement in interesting ways. An agent can be blocked or pushed around by other agents in its path, which can result in diverse dynamics depending on the model parameters.

When the packing density ρ is not too high (agents can freely move past each other), lateral migration due to interactions with other agents causes agents of the same group to find each other passively and form clusters [Fig. 1(b), top panel (Multimedia view)]. The cluster of agents moves together as a unit, and collisions with the opposite group appear only at its boundary. For these reasons,

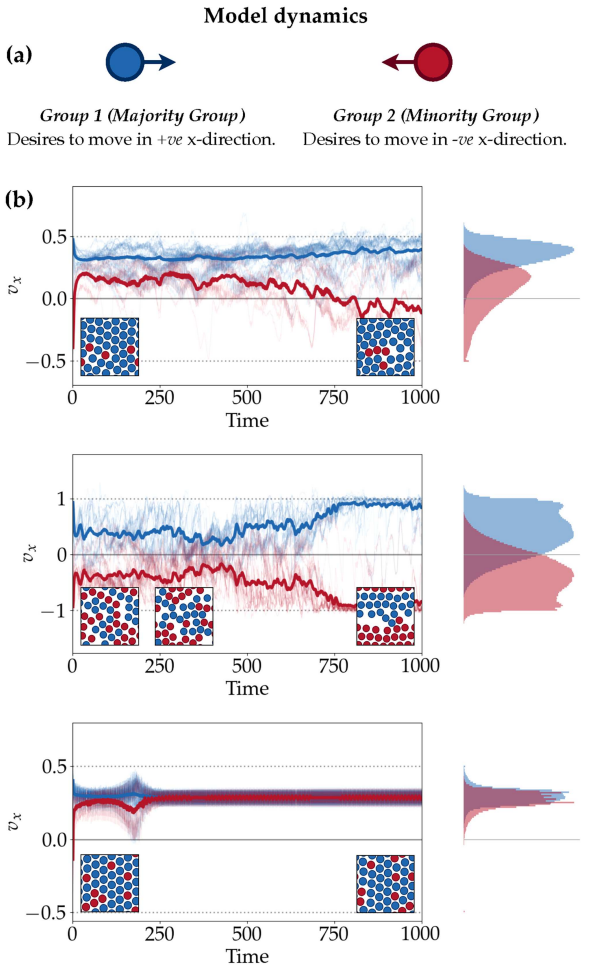

FIG. 1. The bidisperse collective system exhibits a wide range of dynamics. (a) The system consists of two different groups of agents, each with its own desired direction of motion. (b) Examples of dynamics exhibited by the model at different parameter values. Plots show the horizontal velocities ($\mathbf{v}_{i,j}\cdot\mathbf{e}_x$) for the two groups of agents from a single realization. The faint blue and red traces are the individual agent velocities for Group 1 and Group 2 agents, respectively. The thick blue and red traces are the average velocities (averaged over all agents in the group) for Group 1 and Group 2 agents. The dotted gray lines show $\pm s_0$, the desired x-velocities for the two groups. Insets show snapshots of the model dynamics at different time-points. The histograms show the distribution of agent velocities for the two groups, across all time-points, and across 100 realizations. Notice that the histograms have a significant overlap, and histograms of the minority group (red in the top and bottom plots) are significantly shifted from their respective s_0. (Also, see Movie S1) Top: Example of clustering ($N_r = 3/14$, $\rho = 0.46$, $s_0 = 0.5$. Middle: Example of lane formation ($N_r = 1/2$, $\rho = 0.46$, $s_0 = 1$). Bottom: Example of jamming ($N_r = 3/14$, $\rho = 0.58$, $s_0 = 0.5$). Multimedia view: https://doi.org/10.1063/5.0093682.1

clustered state is an absorbing state: i.e., once a cluster is formed, it does not break easily, but new agents can join the cluster. Clustering improves mobility of the agents, as the cluster as a whole is able to better force its way through opposing agents. Here, the word mobility is used to qualitatively describe the net movement of the agents in the collective: higher mobility implies a larger movement of agents in their respective desired directions.

When N_r is close to $\frac{1}{2}$ (symmetric or nearly symmetric regime), clustering can eventually lead to formation of system-spanning lanes [Fig. 1(b), middle panel (Multimedia view)]. Since each lane consists only of one group of agents, mobility is maximum (agent speeds are close to $\pm s_0$) in the laned state.

When ρ is high and the intrinsic velocity s_0 is relatively small, the collective can enter a jammed state, where agents meet head-on and do not have enough space to move past each other [Fig. 1(b), bottom panel (Multimedia view)]. For the symmetric case ($N_r = \frac{1}{2}$), this causes the entire assembly to freeze. For asymmetric cases ($N_r < \frac{1}{2}$), due to an imbalance in the net force, the jammed assembly of agents slowly drifts in the direction dictated by the majority group.

Shown alongside the velocity plots are the distributions of the individual velocities for the two groups. For asymmetric cases, the minority group agents are often pushed in the opposite direction by the majority group. As a result, the velocity distribution for the minority group is shifted in the positive direction (see top and bottom panels). Besides, due to collective effects, such as local jamming, there is a significant overlap between the velocity distributions of the two groups: for example, when mobility is very low, the two distributions are nearly indistinguishable (bottom panel). In other words, there can be agents of Group 1 and Group 2 moving with identical velocities.

III. OBSERVER DESIGN AND CLASSIFICATION

In the context of our model collective, the three levels of the inference problem become the following:

- *Group heterogeneity level*: The number ratio N_r is a direct measure of the level of heterogeneity in the group: heterogeneity is maximum at $N_r = 1/2$ and low when N_r is low. Hence, at this level, our goal would be to estimate N_r.
- *Classification level*: To classify agents into Group 1 or Group 2, we only need to ascertain the direction of $\mathbf{v}_{0,i}$, and the magnitude is irrelevant.
- *Discovering intrinsic drives*: This involves precisely identifying both the magnitude and direction of $\mathbf{v}_{0,i}$.

In this paper, we mainly focus on the first and second inference problems and provide a broad discussion on how one could tackle the third, harder problem.

An observer collects movement information about the agents, i.e., their positions and velocities, but has no other information about the details of the model. The observer then attempts to classify agents as Group 1 or Group 2 [Fig. 2(a)].

A. Observing agents with a simple observer model

A straightforward approach for classification is to classify agents based on their observed direction of motion. For a better estimate that eliminates transient fluctuations, one may use the average velocity computed over a time window. Therefore, our simple observer algorithm proceeds as follows. First, the observer computes the average velocity v_i^w of the agents, averaged over a time window of length w,

$$v_i^w = \langle \mathbf{v}_i \cdot \mathbf{e}_x \rangle_w. \tag{4}$$

The observer classifies an agent based on the net direction of motion during this window, i.e., the sign of v_i^w,

$$\begin{aligned} v_i^w \geq 0: &\quad \text{Group 1,} \\ v_i^w < 0: &\quad \text{Group 2.} \end{aligned} \tag{5}$$

Figure 2(b) is a visual representation of the algorithm.

For a given window, the number of misclassifications is identified by comparing the labels of the agents predicted by the observer to the ground-truth labels (the actual desired directions for the agents, which are unknown to the observer). For each set of values for the parameters N_r and s_0, we compute the probability of misclassifications (p_m), computed over multiple time windows and multiple independent realizations of the simulation. To estimate the level of heterogeneity in the system, we use the estimated value of N_r, i.e., the fraction of agents classified as Group 2 by the observer.

This simple observer algorithm is the most straightforward way to classify agents. It assumes that the information necessary to classify an agent is fully contained in its own motion and does not explicitly account for its interactions with the surroundings. The simple observer will serve as a baseline to study the effect of emergent collective dynamics on classification before we develop improved classification models.

In a real-world scenario, an observer has to tackle the problem of sensor noise that may corrupt the observations. However, we make the simplifying assumption that the observer can observe the system perfectly without noise; the only unknown is the individual group identities of the agents, which the observer desires to infer.

Notice that the window-length w is an observer timescale, independent of any intrinsic timescales of the system. Here, w corresponds to the duration of time for which the observer collects information before it has to make a prediction. In our analyses, we used a window-length $w = 50$, which varies from 0.5 to 15 times the system timescale R/s_0. A shorter window corresponds to decision-making with fewer data points, while a larger window size may result in improved classification performance. The exact window length does not qualitatively affect the results, and we verified this by repeating our analysis with several window sizes between 25 and 100.

1. The simple observer mis-identifies minority group agents

The simple, velocity-based classification approach is most effective when the mobility is high (s_0 is high and ρ is low), and agents are free to move in their own desired directions [Fig. 2(c), top right panel]. In general, however, agents will be pushed around by their neighbors and may not move in their desired directions at all times, which will lead to misclassifications. This effect is worse for the minority group—hence, the misclassification probability for the minority group is higher for smaller group sizes [Fig. 2(c), main panel]. This effect is even more pronounced in the case of very low

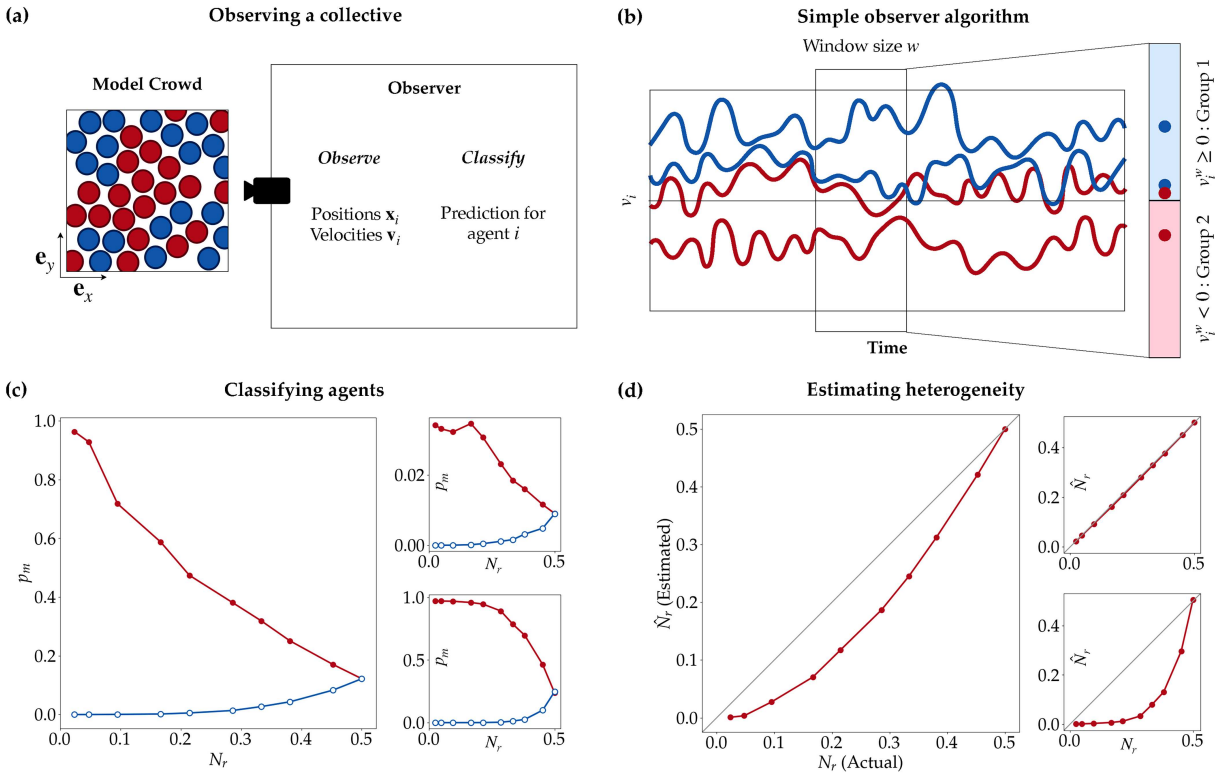

FIG. 2. Classifying agents with a simple observer: the simple observer algorithm systematically misclassifies minority group agents and underestimates the level of heterogeneity. (a) Illustration of the observer framework: the observer observes the positions and velocities of agents in the collective and employs a classifier to classify agents as Group 1/Group 2. (b) Classification algorithm for the simple observer: For each agent, the observer finds the average x-velocity of the agent within a short time window and classifies the agent based on whether this velocity is positive or negative. (c) The probability of misclassifying agents p_m for the minority (red, closed circles) and majority (blue, open circles) groups, as a function of the number ratio N_r, for different levels of agent mobility (main: intermediate mobility, $\rho = 0.46$, $s_0 = 1$, top right: high mobility, $\rho = 0.31$, $s_0 = 2$, bottom right: low mobility, $\rho = 0.58$, $s_0 = 0.75$). p_m is higher for lower N_r, denoting poorer classification performance for smaller groups. (d) The estimated value of N_r, which is a measure of estimated heterogeneity, as a function of true N_r, for different levels of agent mobility. Except when mobility is high, simple observer systematically underestimates $\hat{N}_r$.

mobility (low s_0 and high ρ) and low N_r when almost all minority agents will be pushed in the wrong direction. Until N_r reaches a critical threshold so that the agents can start to form clusters, all minority agents will be misclassified.

This effect is also visible in the estimated heterogeneity, $\hat{N}_r$ [Fig. 2(d)]. In general, the simple observer consistently underestimates the proportion of minority group agents except when the mobility is very high. When mobility is low, the observer estimates zero heterogeneity for small values of N_r; i.e., the observer cannot even detect the presence of minority agents.

B. Correcting misclassifications with a neighborhood-based observer

The simple observer algorithm classifies agents based on their net velocity measured over a time window w. For instance, when the observed velocity v_x^w of a Group 2 agent is positive, it will be classified as Group 1 incorrectly by the observer. Thus, correcting a misclassification would require the observer to classify the agent as belonging to Group 2 even if it exhibits a net movement in the positive direction. From the perspective of an observer, this is a counter-intuitive step: the observer has to differentiate agents in Group 2 from those in Group 1 that happen to move in the same x-direction. We need to ascertain the conditions under which an observer should swap the identified group-identity of the agent to correct a possible misclassification event.

1. An agent's neighborhood encodes information about its intrinsic direction

In the absence of interactions, an agent moves in its desired direction of motion with velocity s_0, and classification is a trivial task. However, an agent's movement is influenced by both its own intrinsic motion as well as interaction effects from its neighbors, which can aid or hinder the motion. For improved classification performance, an observer needs to decouple the component of an

agent's motion due to its intrinsic drive and the component of motion caused by these interaction effects.

In the current formulation of the classifier, the observer uses v_i as a proxy for the agent's desired velocity, $\mathbf{v}_{0,i}$. However, in reality, the observed $\mathbf{v}_i$ is a combination of the agent's effort to move in $\mathbf{v}_{0,i}$ and the support or resistance offered by the neighborhood $\boldsymbol{\phi}_i$,

$$v_i = \mathbf{v}_{0,i} \cdot \mathbf{e}_x + \boldsymbol{\phi}_i \cdot \mathbf{e}_x. \tag{6}$$

See Fig. 3(a) for an illustration. With this definition [Eq. (6)], the new classification criterion becomes

$$\begin{aligned} \mathbf{v}_{0,i} \cdot \mathbf{e}_x = v_i - \phi_i \geq 0: &\quad \text{Group 1}, \\ \mathbf{v}_{0,i} \cdot \mathbf{e}_x = v_i - \phi_i < 0: &\quad \text{Group 2}, \end{aligned} \tag{7}$$

where $\phi_i = \boldsymbol{\phi}_i \cdot \mathbf{e}_x$. In other words,

$$\begin{aligned} v_i \geq \phi_i: &\quad \text{Group 1}, \\ v_i < \phi_i: &\quad \text{Group 2}. \end{aligned} \tag{8}$$

That is, an agent i is classified as Group 1 or Group 2 by comparing v_i to ϕ_i. This means that two agents i and j can be classified differently even when $v_i = v_j$ depending on ϕ_i and ϕ_j.

We now discuss an approach to estimate $\boldsymbol{\phi}_i$. As mentioned before, we use information from the local neighborhood to compute a *neighborhood parameter* $\widehat{\boldsymbol{\phi}}_i$, which will serve as a surrogate for $\boldsymbol{\phi}_i$.

We pause to recap our notations: $\boldsymbol{\phi}_i$ is the (unknown) effect of the neighborhood on agent motion, and ϕ_i is the x-component of $\boldsymbol{\phi}_i$. $\widehat{\boldsymbol{\phi}}_i$ is our estimate for $\boldsymbol{\phi}_i$, and $\hat{\phi}_i$ is the x-component of $\widehat{\boldsymbol{\phi}}_i$.

To define $\widehat{\boldsymbol{\phi}}_i$, we make the following observations and assumptions:

(i) an agent's motion is influenced predominantly by its immediate neighbors,
(ii) a neighbor can either aid or oppose the intrinsic movement of the agent, and
(iii) the influence of a neighbor on an agent is dependent on the relative velocity of the neighbor, i.e., how fast the neighbor is approaching the position of the agent.

Based on this, we define $\widehat{\boldsymbol{\phi}}_i$ as follows:

$$\widehat{\boldsymbol{\phi}}_i = \mu \left\langle (\mathbf{v}_j \cdot \mathbf{e}_{ji})\, \mathbf{e}_{ji} \right\rangle_{j \in \mathcal{N}_i}. \tag{9}$$

The parameter $\widehat{\boldsymbol{\phi}}_i$ is a vector capturing the net effect of neighbors (in the Voronoi neighborhood $\mathcal{N}_i$) on the focal agent i. Agents interact as they approach each other; therefore, the neighbor velocities are projected along the line joining the centers of i and j. The projected velocity vectors are then averaged to get the net neighborhood effect. μ is a scaling factor that ensures that $\widehat{\boldsymbol{\phi}}_i$ is of the same scale as $\mathbf{v}_i$. See Fig. 3(a) for an illustration.

One can estimate μ using a very simple scaling argument. Consider the case with the focal agent surrounded by six neighbors, in a regular hexagonal arrangement. In the extreme case, all of these neighbors are moving in the same direction, with the same velocity $\mathbf{v}$. In this case, $\left\langle (\mathbf{v}_j \cdot \mathbf{e}_{ji})\, \mathbf{e}_{ji} \right\rangle_{j \in \mathcal{N}_i}$ will evaluate to $\frac{1}{6}(2 + 4\cos\frac{\pi}{3})\mathbf{v} = \frac{2}{3}\mathbf{v}$, giving $\mu = \frac{3}{2}$. Although a crude approximation, this value of μ works well in practice. See Sec. IV for a comparison with an alternative, data-driven approach to estimating μ.

It is important to note that the neighborhood parameter is independent of the details of the simulation model. We use relative velocities of the neighbors as a proxy for inter-agent interactions; in other words, the neighborhood parameter only depends on the overall phenomenology of the collective and not on specific modeling assumptions.

2. Classifying agents with the neighborhood observer

In addition to computing the movement characteristics of just the agent v_i, the neighborhood observer calculates ϕ_i, which is a characteristic of its neighborhood. As before, we use the time-window-averaged versions of these quantities.

Figure 3(b) (Multimedia view) illustrates how the neighborhood observer works. The plot shows the time-series of the (window-averaged) agent velocity v_i^w for a Group 2 agent. When it encounters Group 1 agents in its path, it may get pushed, causing v_i^w to rise above 0. The agent-only observer would then classify the agent—incorrectly—as Group 1: see blue shaded regions in the plot. However, the classification threshold for the neighborhood observer is ϕ_i^w (blue dashed line in the plot), incorporates information about the neighborhood of the agent, and can vary according to how much the agent is being pushed. Hence, the neighborhood observer is able to correctly classify the agent as Group 2 even when $v_i^w > 0$; i.e., the agent is moving in the positive x-direction.

3. The neighborhood observer improves performance on minority agents

Accounting for neighborhood effects significantly improves classification performance, especially for the minority group [Fig. 3(c), main panel]. Even when only a single Group 2 agent is present, the neighborhood observer can correctly classify the Group 2 agent 60% of the time. On the other hand, since a lone Group 2 agent is prone to getting pushed by Group 1 agents most of the time, simple observer almost always misclassifies it. There is a slight increase in the probability of misclassifying the majority group agents but is small relative to the improvement achieved for the minority group: the *overall* probability of misclassification is lower for the neighborhood observer (see Appendix A and Fig. 6).

Even in the high-mobility scenario, where the performance of the simple observer is already high, the neighborhood observer achieves significant improvements. In the low mobility scenario, where the agents are mostly jammed and there is very little relative movement, the improvement is relatively minor—see Sec. III C for a discussion about why this happens.

The neighborhood observer can accurately estimate the level of heterogeneity across a wide range of parameters, even in cases where the simple observer underestimates it [Fig. 3(d), main panel]. The heterogeneity estimate suffers when the mobility is very low, but the estimates are overall better than that of the simple observer. In particular, the neighborhood observer is able to detect the presence of Group 2 agents at much smaller values of N_r than the simple observer.

Here, we considered the probability of misclassification p_m averaged across the duration of the simulation across all realizations. However, across the duration of the simulation, the agents can

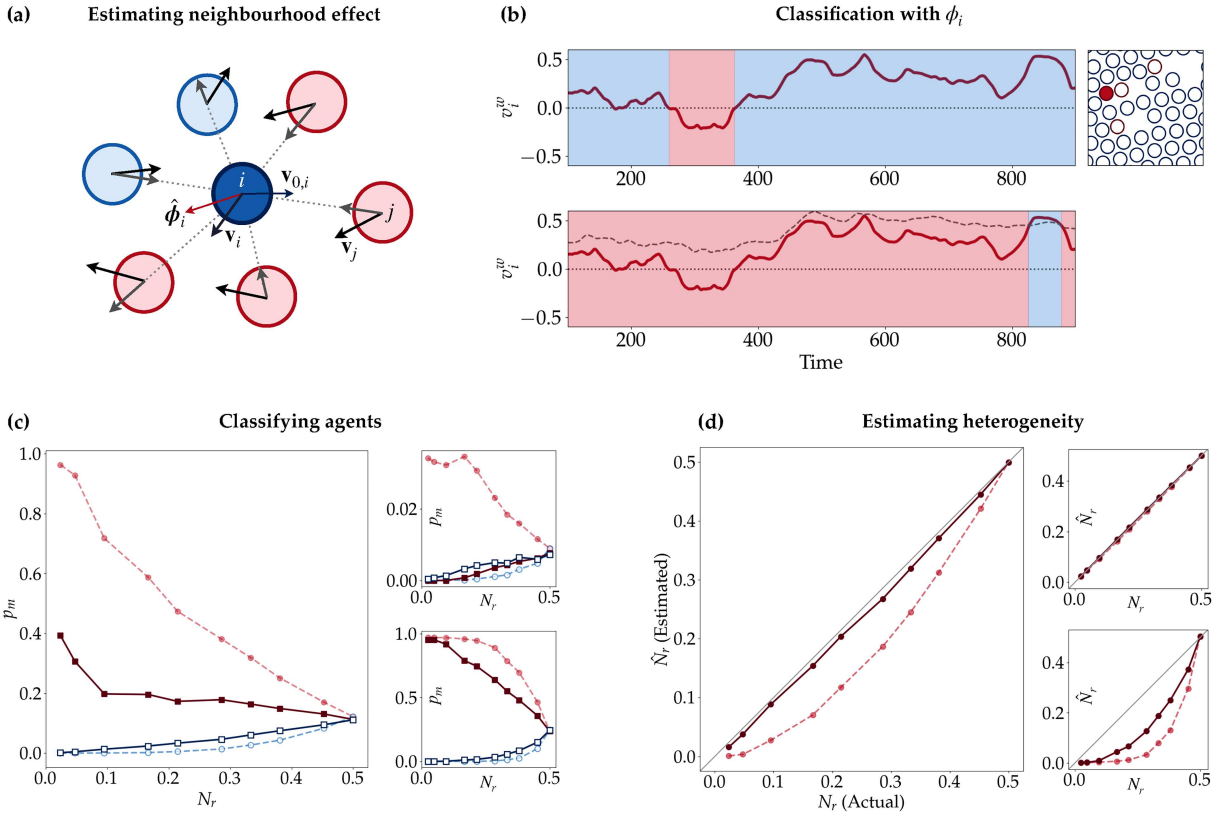

FIG. 3. Classifying agents with a neighborhood observer: an observer algorithm that accounts for neighborhood effects significantly improves classification performance and estimates heterogeneity more accurately. (a) The velocity $\mathbf{v}_i$ of an agent can be decomposed into two components: $\mathbf{v}_{0,i}$, the intrinsic velocity of the agent, and ϕ_i, the influence of neighbors. The neighborhood parameter $\widehat{\phi}_i$ is an estimate of ϕ_i using velocities of neighbors. $\widehat{\phi}_i$ for an agent i (deep blue) is determined by the velocities of its neighbors (pale blue and red). The contribution of neighbor j is $\tilde{\mathbf{v}}_j$, the component of the velocity vector $\mathbf{v}_j$ toward i. $\widehat{\phi}_i$ is the mean of $\tilde{\mathbf{v}}_j$'s, scaled by an appropriate scaling factor. (b) Visualizing the working of the neighborhood classifier: The top panel illustrates the working of the simple observer. The red solid line is the observed velocity v_i^w of a Group 2 agent from a single realization (averaged over window w). The shading corresponds to the predictions of the simple observer (blue: Group 1, red: Group 2). The simple observer misclassifies the agent as Group 1 whenever $v_i^w > 0$, which is often the case. The box of the right shows a snapshot from the simulation, showing the Group 2 focal agent being pushed by Group 1 agents. The bottom panel illustrates the working of the neighborhood observer for the same agent. The red solid line is the observed velocity v_i^w, and the dotted line is the neighborhood parameter ϕ_i^w. The shading corresponds to the predictions of the neighborhood observer. The neighborhood observer classifies the agent correctly as Group 2 most of the time. (c) The probability of misclassifying agents p_m for the minority (red, filled squares) and majority (blue, open squares) groups, as a function of the number ratio N_r, for different levels of agent mobility (main: intermediate mobility, $\rho = 0.46$, $s_0 = 1$; top right: high mobility, $\rho = 0.31$, $s_0 = 2$; bottom right: low mobility, $\rho = 0.58$, $s_0 = 0.75$). The performance curves of the simple observer are shown as dotted lines. The neighborhood observer drastically reduces misclassifications on the minority group. (d) The estimated value of N_r, which is a measure of estimated heterogeneity, as a function of true N_r, for different levels of agent mobility. The performance curves of the simple observer are shown as dotted lines. The neighborhood observer estimates heterogeneity better than the simple observer and can detect the presence of heterogeneity at a lower N_r. Multimedia view: https://doi.org/10.1063/5.0093682.2

self-organize and form clusters or lanes. To understand this behavior, one may look at how p_m evolves with time. Although a detailed study of this aspect is outside the scope of this article, we briefly explore this in Appendix B.

C. Analysis of the observer algorithms

Figure 4(a) illustrates why the simple, velocity-based classification approach often fails. The top panel shows the histograms of the observed horizontal velocities ($v_i = \mathbf{v}_i \cdot \mathbf{e}_x$) of the two groups for a typical scenario (asymmetric, intermediate mobility). The minority group (Group 2) agents end up being pushed in the opposite direction frequently. As a result, the velocity distribution shifts to the right, with the bulk of the distribution lying on the positive side. As a result, the simple observer, which classifies agents based on whether the observed velocity is positive or negative, misclassifies most of the minority group agents.

The bottom panel of Fig. 4(a) illustrates how the neighborhood observer gets around this problem. The distribution of $v_i - \phi_i$ for the

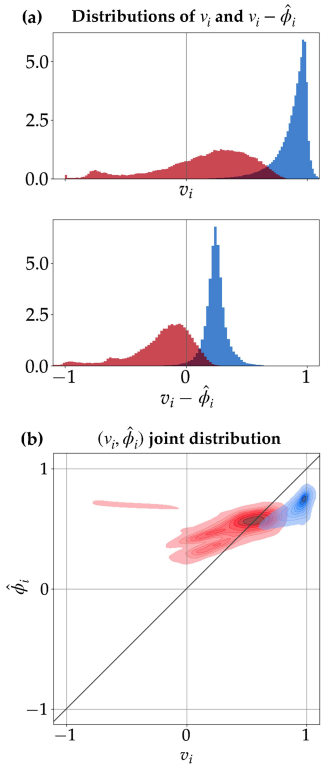

FIG. 4. Illustration of how the neighborhood classifier works. (a, top) Example distribution of individual horizontal velocities v_i for Group 1 (blue) and Group 2 (red) agents. A large portion of the distribution for Group 2 (the minority group) lies on the positive side, leading to misclassifications. (a, bottom) Example distribution of $v_i - \phi_i$, the classification parameter used by the neighborhood observer. The distributions are shifted so that the majority of the Group 2 distribution lies on the negative side. (b) The joint distribution of v_i and ϕ_i for the two groups of agents. The classification boundary for the simple observer is the $v_i = 0$ vertical line, while the classification boundary for the neighborhood observer is the $v_i = \phi_i$ diagonal line.

minority group is more predominantly on the negative side compared to the v_i distribution. Therefore, the neighborhood observer (which classifies agents based on $v_i \lessgtr \phi_i$, i.e., $v_i - \phi \lessgtr 0$) is able to classify the minority group agents much more accurately. Notice that the velocity distribution of the majority group stays predominantly on the positive side, even though it moves closer to the origin.

The observers can alternatively be viewed as linear classifiers in the (ϕ_i, v_i) plane, as illustrated by Fig. 4(b). The figure shows the joint distributions of ϕ_i and v_i for the two different groups. The simple observer classifies agents based on v_i and ignores ϕ_i altogether; therefore, its decision boundary is the $v_i = 0$ line, i.e., a vertical line through the origin. Since the bulk of the minority group distribution lies to the right of this line, misclassifications will be high. On the other hand, the decision boundary for the neighborhood observer is the $v_i = \phi_i$ line, i.e., a diagonal line with slope 1. The bulk of the minority group distribution is above this line and will be classified correctly. Notice that all agents with a given observed velocity v_i will be classified similarly by the simple observer. On the other hand, the neighborhood observer can distinguish between agents with the same observed velocity depending on where they lie along the ϕ_i-axis.

1. Effect of mobility and heterogeneity

The (v_i, ϕ_i) plane presents an elegant way to observe how mobility and heterogeneity affect the classification performance of the two observers, as illustrated in Fig. 5.

The effect of mobility is quite clear: as mobility increases (s_0 increases and/or ρ decreases), obstructions and jamming decrease and the agents move with higher velocities more often. This reflects as a separation of Group 1 and Group 2 distributions on the v_i, ϕ_i plane [Figs. 5(a)–5(c)]. At low mobility, there is a large amount of overlap between the two distributions, and both classifiers perform relatively poorly. On the other hand, at high mobility, the distributions are well separated and even the simple observer performs well.

The effect of heterogeneity on the distributions is perhaps more interesting. In the symmetric case ($N_r = 1/2$), there is no bias in the movement freedom for either group. This means that the distributions are arranged symmetrically on either sides of the $v_i = 0$ line, with their overlap determined by the level of mobility. As N_r decreases and the collective becomes more asymmetric, the minority agents gets pushed more and the mixture velocity (i.e., the average velocity computed over all the agents) increases in the positive direction, causing both distributions to shift right. This degrades the performance of the simple observer.

However, when an agent is being pushed with some velocity, ϕ, which is a proxy for the level of "push" on an agent by its neighbors, also increases proportionately: this means that in the (v_i, ϕ_i) plane, the shift happens diagonally along the classification boundary of the neighborhood observer: the regions of the distributions that are above and below the classification boundary stay roughly the same. This is the reason why asymmetry does not degrade the performance of the neighborhood observer too much.

IV. DISCUSSION

A. Summary

In this article, we set out to understand how well can we infer individual properties of agents in a heterogeneous collective by observing their movement. We used a bidisperse collective as a context and studied how the classification process of an observer is connected to the phenomenology of collective motion. We began with a simple observation/classification technique based on the measured velocity of the individual agents, independent of their neighborhood. We found that the simple observer underestimates the level of heterogeneity of the collective and systematically misclassifies the minority group agents more often than the majority group.

To improve classification, we developed an observer that classifies an agent based on information not just about itself, but also

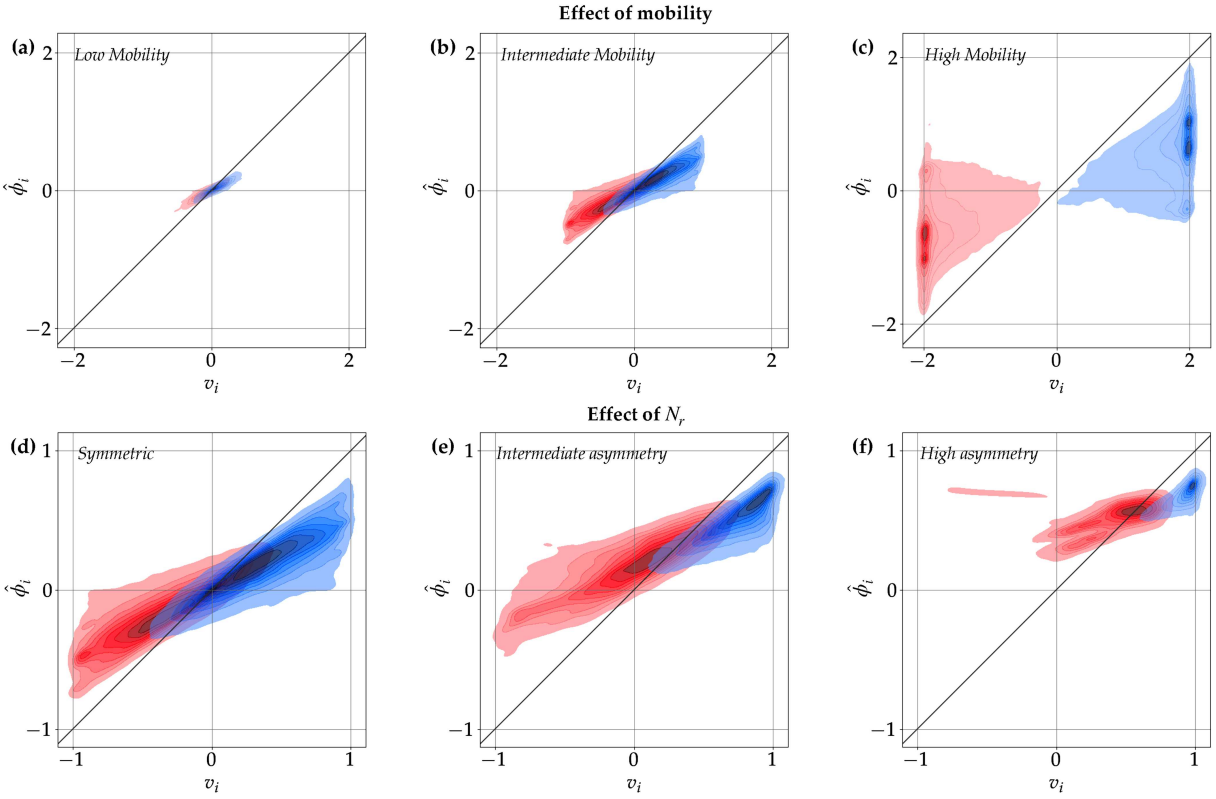

FIG. 5. Effect of mobility and heterogeneity on the distributions and classification performance. (a)–(c) As mobility increases, the agents move more freely, causing the distributions to become well separated, improving classification performance of both observers. (d)–(f) As asymmetry between the group sizes increases (i.e., N_r decreases), the distributions shift toward the right, resulting in higher misclassifications for the smaller group by the simple observer. However, the shift is along the diagonal, which ensures that the performance of the neighborhood observer does not drop significantly.

about its neighbors. Although simple in principle, this neighborhood observer does something quite non-trivial: it distinguishes between a Group 1 and Group 2 agent even when they are moving in the same direction with identical velocities. The fact that the neighborhood observer takes into account the influence of the neighboring agents for classification helps it to "read the mind" of the focal agent by distinguishing when it is moving on its own volition vs when it is being pushed. This resulted in better estimates of the level of heterogeneity and improved classification for the minority group.

B. Data-driven approaches to the classification problem

The (v_i, ϕ_i) classifier that the neighborhood observer employs is not data-driven: the classification boundary (which is decided by μ) is not obtained by a data fit but is instead derived from scale considerations. This makes our approach applicable in scenarios where labeled data are not easy to obtain. This also makes our classifier readily interpretable, as it is inspired by the physics of the problem. The assumptions the classification algorithm makes about the underlying governing dynamics are minimal. Specifically, the neighborhood parameter is computed based on neighbor velocities only and is independent of specific details of how the agents interact.

If labeled data (i.e., data where the group memberships of agents are known) are readily available, we can learn μ in a data-driven manner by fitting a linear classifier. When a classifier is fit in this manner, it will not be susceptible to unfulfilled assumptions in computing μ. However, such a data-driven classifier offered no improvements over the simple linear classifier (see Appendix C), underscoring the accuracy of the scaling argument.

An orthogonal approach is to build a purely data-driven classifier, for example, based on neural networks. One could build a neural-net classifier that takes as input the positions and velocities of the focal agent and its neighbors and makes predictions. Given enough labeled data, such classifiers can learn high-quality, low-dimensional feature representations that can make effective predictions. To be effective, such neural network classifiers should be informed and constrained by the physics and inherent symmetries of the problem. In our case, it means that the neural

network should respect the rotational and translational invariances of the system and should be invariant to any permutation of the agent ordering. Constraints such as these should be built into the neural network design. There is an emerging body of research in graph neural networks[40,41] and physics-informed deep learning,[42,43] which can be explored in this regard. Building neural network classifiers for collective dynamics and studying their feature representations is an exciting research direction.

C. Toward a general inference framework for heterogeneous collectives

There are several ways in which the classification algorithm can be improved further. First, notice that the ultimate goal would be to recover $\mathbf{v}_{0,i}$, for example, by finding ϕ_i such that $\mathbf{v}_i - \phi_i = \mathbf{v}_{0,i}$. Our estimated neighborhood parameter does not quite achieve this: however, it gets the sign correctly in general; i.e., $\mathrm{sgn}(v_i - \phi_i) = \mathrm{sgn}(v_{0,i})$, which is sufficient for classification.

Recall that the neighborhood parameter for an agent i was computed based on the mean of the velocities of the Voronoi neighbors of i. Potential ways to improve this would be to use a different neighborhood, use a weighted average weighted by the distance to the neighbor, or use more complicated functions of the velocity.

One could also use information about the spatial arrangement of the neighbors of i. For example, it is conceivable that the fore-aft asymmetry pattern in the neighborhood of i will be different when i is being pushed vs when i is moving freely. Quantifying these spatial patterns could be an effective approach to recover intrinsic motion.

Finally, note that both the observers treat the time windows as independent: when making predictions for a given time window, the observers do not use information from previously observed windows. One could conceive an observer that uses historical information and updates beliefs with time, perhaps in a Bayesian fashion. We do not explore this approach because our goal is to explore techniques that work in a nearly *real-time* manner with minimal observations and to understand how instantaneous and short-term dynamics in the system affects our ability to classify agents. Another way to incorporate temporal information would be to look for patterns in the higher derivatives (e.g., acceleration) of motion of both the focal agent and the neighborhood.

D. Conclusion

It is worth reiterating that the goal of this study is to understand the ways in which the microscopic properties of a heterogeneous collective affect the classification performance of an observer that is only privy to observables, such as agent positions and velocities. We study this in the context of a model system, which is an idealized version of a heterogeneous collective in the real world. For this reason, hunting for the best data-driven classifier to solve this specific problem may not be very useful; such a classifier will not be transferable to a real-world collective as real-world collectives can be poly-disperse and can involve many complex interactions not included in our model. Hence, our main goal is to understand how the nonlinear collective dynamics can affect classification. A clear understanding of this relationship is essential to solve similar problems in real-world collectives. Our phenomenological approach to the classification problem is a first step toward hybrid techniques, where a data-driven approach is combined with domain-specific understanding of the collective dynamics to build better observer models.

ACKNOWLEDGMENTS

The authors thank Jason Ryan Picardo, Ganapathy Ayappa, Vishwesha Guttal, Vivek Jadhav, Basil Thurakkal, and Harishankar Muppirala for critical readings of the manuscript. The authors thank the DST INSPIRE faculty award for the funding (Grant No. DST/INSPIRE/04/2017/002985).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The simulation and analysis code used in this work is openly available in GitHub at https://github.com/arshednabeel/ObservingACrowd, Ref. 44.

APPENDIX A: TOTAL MISCLASSIFICATIONS FOR THE NEIGHBORHOOD CLASSIFIER

Figure 3 shows the overall misclassification probabilities separately for the two groups of agents. The neighborhood observer

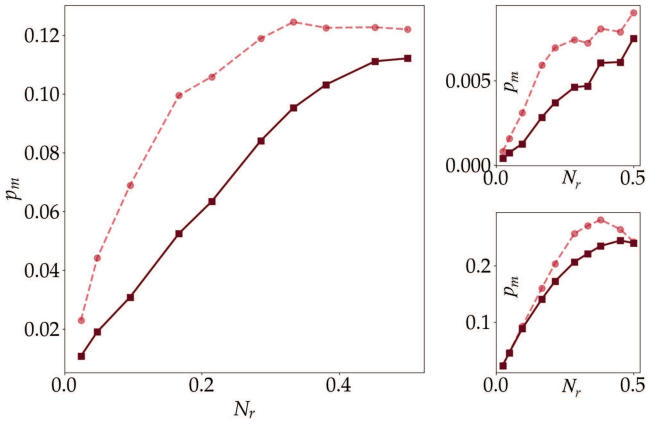

FIG. 6. Comparing the overall misclassification probability between the two observers. The overall probability of misclassifying agents, p_m as a function of the number ratio N_r, for different levels of agent mobility (main: intermediate mobility, $\rho = 0.46, s_0 = 1$, top right: high mobility, $\rho = 0.31, s_0 = 2$, bottom right: low mobility, $\rho = 0.58, s_0 = 0.75$). The performance curves of the simple observer are shown as dotted lines. The neighborhood observer reduces overall misclassifications across a range of N_r and mobility-regimes.

improves classification performance for the minority group while slightly degrading the performance for the majority group. Therefore, it is possible that the neighborhood observer actually increases the *overall* misclassification probability, especially for smaller values of N_r. Figure 6 shows that this is not the case. The overall probability of misclassification is lower for the neighborhood observer for all values of N_r and across mobility-regimes—the accuracy loss for the majority group is compensated by the much greater accuracy improvements for the minority group.

APPENDIX B: EVOLUTION OF MISCLASSIFICATION PROBABILITY WITH TIME

In the main article, we studied the probability of misclassification p_m averaged across time and across all realizations. However, p_m may depend on the state of the collective, which varies with time. For instance, a well-laned configuration is expected to have low values of p_m for both groups, while a jammed configuration will have large p_m especially for the minority group. We know from the dynamics of a bidisperse crowd that the state of the collective changes with time as agents interact and self-organize as they move in the crowd (see Sec. II B and Fig. 1, Multimedia view). Hence, it is interesting to study how p_m changes with time.

In Fig. 7, we show how p_m evolves in time for asymmetric and symmetric crowds with high and intermediate mobilities; p_m is averaged across all the agents of a given group and over many realizations. When mobility is high, the agents can form clusters or lanes, which facilitate easy movement. Hence, p_m quickly decreases to 0 in this case for both observers in both the asymmetric and symmetric crowds [Figs. 7(a) and 7(b)]. On the other hand, when mobility is lower, organizing into stable clusters and lanes is more difficult. Hence, the variations in p_m are dominated by local collisions and interactions and tend to fluctuate more [Figs. 7(c) and 7(d)].

As observed before [see Sec. III B and Fig. 3(c)], the neighborhood observer achieves a lower p_m for the minority group, throughout time, irrespective of the collective state of the system [top panels of Figs. 7(a)–7(d)].

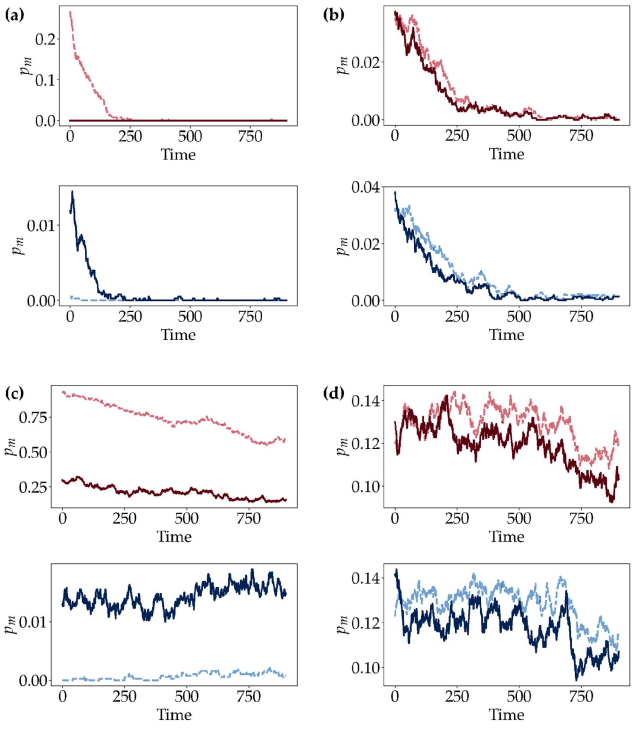

FIG. 7. Time-evolution of misclassification probability p_m. (a) and (b) show p_m as a function of time for the high-mobility regime ($\rho = 0.31, s_0 = 2$), averaged over all 100 realizations, for the asymmetric ($N_r = 2/21$) and symmetric ($N_r = 0.5$) cases, respectively. The top plot in each panel shows p_m for the minority group (dashed line: simple observer, solid line: neighborhood observer), and the bottom graph shows the majority group. Agents quickly attain laned configurations, and the p_m rapidly decreases to 0. (c) and (d) show p_m as a function of time for the intermediate mobility regime ($\rho = 0.46, s_0 = 1$). Formation of stable clusters and lanes is more difficult in this regime; hence, there is no quick drop in p_m.

APPENDIX C: CLASSIFYING AGENTS WITH A DATA-DRIVEN CLASSIFIER

In Sec. III B, we used a scaling argument to analytically estimate μ. Alternatively, if labeled data are available, μ can be estimated in a data-driven manner by fitting a linear classifier.

Section III B defined the neighborhood classifier with a classification criterion $v_i \lessgtr \phi_i$. Let $\phi_i = \mu\varphi_i$, where φ_i corresponds to the unscaled mean from Eq. (9). In this formulation, the classification boundary (see Sec. III C and Fig. 4) corresponds to a line of slope $1/\mu$ in the (v_i, φ_i) plane. With labeled data, one could fit a linear classifier (e.g., a support vector machine) to the data to obtain the "optimal" μ for the dataset.

We trained an SVM to distinguish between Group 1 and Group 2 agents in this way. Since we are only interested in learning a scaling factor μ, we enforce that the intercept of the classifier be zero. A separate model was fit for each set of parameters (N_r, s_0, and ρ), with data pooled over all realizations and time-points. Since our goal is to find the theoretically optimal μ, we did not use separate training and test sets, and performance reported is from the same dataset used for training: this is an upper bound for reported accuracy numbers with separate training and test sets.

Figure 8 compares the performance of the analytically derived (as in Sec. III B) and data-driven classifiers. The data-driven approach does not show any significant improvement in classification performance—in terms of classifying agents as well as estimating heterogeneity, the SVM-based classifier performs nearly identically as our analytically derived classifier. This suggests that our estimate of μ is close to optimal.

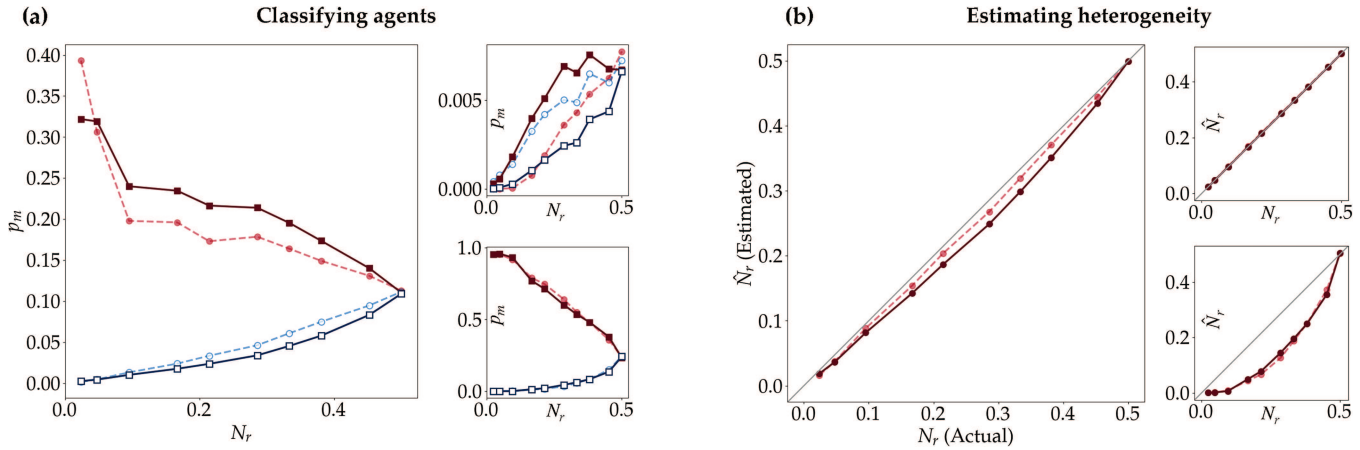

FIG. 8. Comparing the (physics-informed) neighborhood classifier to a data-driven classifier: the physics-informed neighborhood observer performs as good as a data-driven classifier fine-tuned to the dataset. (a) The probability of misclassifying agents p_m for the minority (red, filled squares) and majority (blue, open squares) groups, as a function of the number ratio N_r, for different levels of agent mobility (main: intermediate mobility, $\rho = 0.46, s_0 = 1$; top right: high mobility, $\rho = 0.31, s_0 = 2$; bottom right: low mobility, $\rho = 0.58, s_0 = 0.75$). The performance curves of the physics-informed neighborhood observer are shown as dotted lines. There is no significant improvement with the data-driven classifier. (b) The estimated value of N_r, which is a measure of estimated asymmetry, as a function of true N_r, for different levels of agent mobility. The performance curves of the (analytically derived) neighbor observer are shown as dotted lines.

REFERENCES

[1]M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic, Proc. Natl. Acad. Sci. U.S.A. **105**, 1232–1237 (2008).
[2]Y. Katz, K. Tunstrom, C. C. Ioannou, C. Huepe, and I. D. Couzin, Proc. Natl. Acad. Sci. U.S.A. **108**, 18720–18725 (2011).
[3]D. S. Calovi, A. Litchinko, V. Lecheval, U. Lopez, A. Pérez Escudero, H. Chaté, C. Sire, and G. Theraulaz, PLoS Comput. Biol. **14**, e1005933 (2018).
[4]J. Jhawar, R. G. Morris, U. Amith-Kumar, M. D. Raj, T. Rogers, H. Rajendran, and V. Guttal, Nat. Phys. **16**, 488–493 (2020).
[5]S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (IEEE, 1999), Vol. 1, pp. 703–708.
[6]M. Moussaïd, E. G. Guillot, M. Moreau, J. Fehrenbach, O. Chabiron, S. Lemercier, J. Pettré, C. Appert-Rolland, P. Degond, and G. Theraulaz, PLoS Comput. Biol. **8**, e1002442 (2012).
[7]A. Nicolas, M. Kuperman, S. Ibañez, S. Bouzat, and C. Appert-Rolland, Sci. Rep. **9**, 105 (2019).
[8]V. Murino, M. Cristani, S. Shah, and S. Savarese, *Group and Crowd Behavior for Computer Vision* (Academic Press, 2017), pp. 1–424.
[9]A. Tremel, A. Cai, N. Tirtaatmadja, B. D. Hughes, G. W. Stevens, K. A. Landman, and A. J. O'Connor, Chem. Eng. Sci. **64**, 247–253 (2009).
[10]P. Ariano, C. Distasi, A. Gilardino, P. Zamburlin, and M. Ferraro, J. Neurosci. Methods **141**, 271–276 (2005).
[11]R. McLennan, L. Dyson, K. W. Prather, J. A. Morrison, R. E. Baker, P. K. Maini, and P. M. Kulesa, Development **139**, 2935–2944 (2012).
[12]Y. Sharma, D. A. Vargas, A. F. Pegoraro, D. Lepzelter, D. A. Weitz, and M. H. Zaman, Integr. Biol. **7**, 1526–1533 (2015).
[13]S. Masoumeh Mousavi, I. Kasianiuk, D. Kasyanyuk, S. K. Velu, A. Callegari, L. Biancofiore, and G. Volpe, Soft Matter **15**, 5748–5759 (2019).
[14]B. R. Si, P. Patel, and R. Mangal, Langmuir **36**, 11888–11898 (2020).
[15]B. Shen, M. Leman, M. Reyssat, and P. Tabeling, Exp. Fluids **55**, 1728 (2014).
[16]B. M. Jose and T. Cubaud, Microfluid. Nanofluidics **12**, 687–696 (2012).
[17]N. Bremond, H. Doméjean, and J. Bibette, Phys. Rev. Lett. **106**, 214502 (2011).
[18]R. Candelier and O. Dauchot, Phys. Rev. E **81**, 011304 (2010).
[19]R. Harich, T. Darnige, E. Kolb, and E. Clément, Europhys. Lett. **96**, 54003 (2011).
[20]M. del Mar Delgado, M. Miranda, S. J. Alvarez, E. Gurarie, W. F. Fagan, V. Penteriani, A. di Virgilio, and J. M. Morales, Philos. Trans. R. Soc. B: Biol. Sci. **373**, 20170008 (2018).
[21]J. R. Dyer, D. P. Croft, L. J. Morrell, and J. Krause, Behav. Chem. Ecol. **20**, 165–171 (2008).
[22]J. W. Jolles, A. J. King, and S. S. Killen, Trends Ecol. Evol. **35**, 278–291 (2020).
[23]A. J. W. Ward, T. M. Schaerf, A. L. J. Burns, J. T. Lizier, E. Crosato, M. Prokopenko, and M. M. Webster, R. Soc. Open Sci. **5**, 181132 (2018).
[24]H. Sridhar and V. Guttal, Philos. Trans. R. Soc. B: Biol. Sci. **373**, 20170014 (2018).
[25]H. M. Zhang and T. Kim, Transport. Res. B: Methodological **39**(5), 385–399 (2005).
[26]P. Geoerg, J. Schumann, S. Holl, M. Boltes, and A. Hofmann, Fire Mater. **45**, 529–542 (2021).
[27]L. Qin, D. Yang, W. Yi, H. Cao, and G. Xiao, Mol. Biol. Cell **32**, 1267–1272 (2021).
[28]L. J. Schumacher, P. K. Maini, and R. E. Baker, Cell Syst. **5**, 119–127 (2017).
[29]T. Kwon, O.-S. Kwon, H.-J. Cha, and B. J. Sung, Sci. Rep. **9**, 1–13 (2019).
[30]G. B. Blanchard, A. G. Fletcher, and L. J. Schumacher, Semin. Cell Dev. Biol. **93**, 46–54 (2019).
[31]T. Vissers, A. Wysocki, M. Rex, H. Löwen, C. P. Royall, A. Imhof, and A. Van Blaaderen, Soft Matter **7**, 2352–2356 (2011).
[32]M. E. Leunissen, C. G. Christova, A. P. Hynninen, C. P. Royall, A. I. Campbell, A. Imhof, M. Dijkstra, R. Van Roij, and A. Van Blaaderen, Nature **437**, 235–240 (2005).
[33]K. R. Sütterlin, A. Wysocki, A. V. Ivlev, C. Räth, H. M. Thomas, M. Rubin-Zuzic, W. J. Goedheer, V. E. Fortov, A. M. Lipaev, V. I. Molotkov, O. F. Petrov, G. E. Morfill, and H. Löwen, Phys. Rev. Lett. **102**, 149901 (2009).
[34]D. Helbing, L. Buzna, A. Johansson, and T. Werner, Trans. Sci. **39**, 1–24 (2005).
[35]C. Feliciani and K. Nishinari, Phys. Rev. E **94**, 032304 (2016).
[36]D. Zhang, H. Zhu, S. Hostikka, and S. Qiu, Physica A **525**, 72–84 (2019).
[37]T. Hastie, R. Tibshirani, and J. Friedman, in *The Elements of Statistical Learning* (Springer, 2009), pp. 9–41.

[38]D. Helbing, I. J. Farkas, and T. Vicsek, Phys. Rev. Lett. **84**, 1240–1243 (2000).
[39]C. Reichhardt and C. J. Reichhardt, Soft Matter **14**, 490–498 (2018).
[40]J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, AI Open **1**, 57–81 (2020).
[41]M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," arXiv:2104.13478 (2021).
[42]G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nature Rev. Phys. **3**(6), 422–440 (2021).
[43]M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686–707 (2019).
[44]A. Nabeel and M. D. Raj, "Observing and inferring a collective," (Github repository), see https://github.com/arshednabeel/ObservingACrowd.