Spatial distribution order parameter prediction of collective system using graph network

Huimin Zhao(赵慧敏)[1], Rui Wang(王瑞)[1], Cai Zhao(赵偲)[2], and Wen Zheng(郑文)[1,3,†]

[1] *Institute of Public-Safety and Big Data, College of Data Science, Taiyuan University of Technology, Taiyuan* 030024, *China*
[2] *Center of Information Management and Development, Taiyuan University of Technology, Taiyuan* 030024, *China*
[3] *Center for Healthy Big Data, Changzhi Medical College, Changzhi* 046000, *China*

(Received 6 October 2022; revised manuscript received 5 December 2022; accepted manuscript online 8 February 2023)

In the past few decades, the study of collective motion phase transition process has made great progress. It is also important for the description of the spatial distribution of particles. In this work, we propose a new order parameter φ to quantify the degree of order in the spatial distribution of particles. The results show that the spatial distribution order parameter can effectively describe the transition from a disorderly moving phase to a phase with a coherent motion of the particle distribution and the same conclusion could be obtained for systems with different sizes. Furthermore, we develop a powerful molecular dynamic graph network (MDGNet) model to realize the long-term prediction of the self-propelled collective system solely from the initial particle positions and movement angles. Employing this model, we successfully predict the order parameters of the specified time step. And the model can also be applied to analyze other types of complex systems with local interactions.

Keywords: order parameter, graph network, collective system, active matter

PACS: 64.75.–g, 07.05.Mh, 05.65.+b, 87.64.Aa **DOI:** 10.1088/1674-1056/acb9fa

1. Introduction

Research on collective systems has become a hot topic, and attracts the attention of many scholars in recent few decades. The phenomenon of collective motion exists widely formed by self-propelled particles in nature.[1–5] Collective behavior is ubiquitous in nature, biological systems, and human society. The typical cluster movement mainly includes the proliferation of bacteria and the cluster behavior of a large number of living bodies such as moving fish, birds and formation control of robot swarm.[6–9] It is of great practical significance to study cluster movement in the control, coordination and formation control of multi-agent systems.[10] Then how to quantitatively describe the cluster phenomenon becomes a crucial issue.

With the continuous development of information technology, it has been widely used to simulate the actual collective system by computer technology.[11–13] The Vicsek model was proposed and it was a simple and effective model for simulating collective motion. The model defined an order parameter v_a to describe the synchronization of the system.[14] Based on the Vicsek model, many variations were proposed, such as the limited view angle model.[15] Both models quantify the synchronization of particle motion by the order parameter v_a. However, the spatial distribution of particles has a great difference with the same degree of consistency in the direction of motion, and v_a cannot describe the difference.

Using classical methods, a step-by-step physical evolution process is required. These methods are highly dependent on the motion of particles at each time step. Therefore, it takes a lot of time to obtain the order parameters of systems with large scales and long time steps. To overcome these limitations, many researchers turn to other methods. Recently, there has a surge of interest in using deep learning algorithms to assist physical systems. It has widely used in condensed matter physics,[16,17] fluid mechanics,[18] and statistical physics.[19] New models and applications of deep learning have constantly proposed.[20–31] However, molecular systems involve reference invariance, degeneracy and symmetry, smoothness and other inductive biases[32] that are completely different from images or texts. So it can not straightforward to apply traditional artificial neural networks architectures to deal with them directly. Since the original work on interaction, the graph neural network is applied to predict dynamics.[33–38] The graph network[39] framework fully considers the relational inductive biases with mathematical methods. And it has relational reasoning ability and supports combinatorial generalization. Since then, the research direction and application fields of graph neural networks have been further expanded. A graph neural network is used to predict the long-time evolution of a glassy system.[40] The GNS is trained to simulate complex physics.[41] Therefore, the graph neural network can be effectively used in the prediction and simulation of molecular systems.

In this work, we leverage the developments in deep learning to help describe the collective behavior of self-propelled particles. And the molecular dynamic graph network model

†Corresponding author. E-mail: zhengwen@tyut.edu.cn
© 2023 Chinese Physical Society and IOP Publishing Ltd

is applied to cluster dynamics system. First, we simulate the cluster movement according to the rules of the Vicsek model with restricted vision. And it is found that the view angle has an effect on the direction of particle movement. When the particle motion direction in the system has the same degree of direction consensus, its spatial distribution is different. The article aims to investigate the phenomenon of consistent synchronization but different spatial distribution. We propose a new order parameter φ to quantify the degree of order in the spatial distribution of particles. The results show that the order parameter can effectively describe the transition of spatial distribution from disorder to order. Then, we apply the data obtained from the Vicsek model with restricted vision to the molecular dynamic graph network model, which achieves the long-term prediction of v_{a} and φ at different field of vision. And the results indicate that the model has strong predictive ability. Compared with classical dynamics simulation, our model can get the order parameters of large collective system faster.

2. Models and method

2.1. Datasets

We use the rules of the limited view angle model[15] to produce datasets for each point in a square shaped plane of linear size L with the periodic boundary conditions by a (x_i, θ_i) pair.

The limited view angle model is the Vicsek model[14] with restricted vision. In the Vicsek model, N individuals regarded as mass points move on the $L \times L$ two-dimensional plane, and the plane has periodic boundary conditions. We use the time unit $\Delta t = 1$ as the time interval between two updates of the directions and positions. At the initial moment, each individual is randomly distributed in the plane with random position and random direction of movement. Each individual has the same absolute velocity v. At each moment in the movement process, the direction of movement of each individual is determined by the average direction of individuals within a circle of field of view radius R surrounding the given individual. According to the position and direction at the previous time, the position and the direction of movement at the next time can be obtained. The difference between the limited view angle model and the Vicsek model is that in the Vicsek model, each individual's field of view is omnidirectional views, while the field of view in the limited view angle model is not global. We define the view angle of each individual as ω, $\omega \in (0, 2\pi)$, and the viewing range of each individual is composed of two viewing radii with an angle of ω and the corresponding arc. Only individuals within the field of vision will have an impact on the direction of movement of the individual.

During the movement of the individual on the plane, the position of the i-th individual at time step $t+1$ is

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i \Delta t, \tag{1}$$

here the velocity $\boldsymbol{v}_i$ is a vector with an absolute value v and the direction. And the direction of motion is obtained from following expression:

$$\theta_i(t) = \langle \theta_i(t-1) \rangle_{R\omega} + \Delta\theta, \tag{2}$$

where $\Delta\theta$ represents the random number in the interval $\left[-\frac{\eta}{2}, \frac{\eta}{2}\right]$, η refers to the random disturbance in the environment. $\langle \theta_i(t-1) \rangle_{R\omega}$ is the average velocity direction of all individuals in the neighborhood where individual i is the center, the field of view radius is R, and the field of view angle is ω at time step $t-1$.

According to the rules of the limited view angle model, we used Python to design an iterative program to simulate the self-propelled cluster motion of 1000 particles in a box with periodic boundary conditions $L = 5$, where the initial velocity was set to 0.03 and the field of view radius was 1, the time step was 1000. In order to eliminate the influence of noise on the synchronization of particle motion, the noise was set to 0. For each time step under each angle of view, we generated 100 training sets to train the network, and 100 test sets to test the model. Each data set had a different initial position and direction. After iteration, the position and movement directions of each time step were obtained.

2.2. The MDGNet model

The graph network[39] framework defines a class of functions for relational reasoning on the representation of graph structure, which combines neural networks with mathematical methods. And it has relational reasoning ability and supports combinatorial generalization. We develop a powerful molecular dynamic graph network (MDGNet) based on the graph network model. Here, we will outline the implementation of the MDGNet model.

The MDGNet framework is a graph-to-graph function. The main calculation unit of the MDGNet framework is the GN block, which takes the graph as input and performs calculations on the graph structure, and the returned result is also a graph. Before the GN block, an encoder is added to preprocess the input, which compresses the input into a latent space representation, and extracts important features of the input information. During the whole modeling process, the particle information in the self-driven collective dynamics is converted into a graphic structure. The graph is represented by a triple $G = (u, V, E)$. Here, u represents a global attribute, a feature of the entire graph. $V = \{v_i\}_{i=1:N^v}$ is a set of v_i, where N^v represents the total number of nodes. $E = \{(e_k, r_k, s_k)\}_{k=1:N^{\mathrm{e}}}$ is the set of edges, where e_k is the feature of the k-th edge, r_k and s_k are the indexes of the receiving node and the sending node respectively, and N^{e} is the total number of edges.

When constructing the dataset, we calculate the order parameters v_a and φ of the system at each time step, and train the model to predict. For each training example, we input the coordinates and movement angle of all N particles to our MDGNet model. The graph is constructed according to the distance between particles: the particles at a distance less than 1 are connected by edges in the graph structure. Since the field of view radius is set to 1 in the limited view angles model. The feature of node is the movement angle of the particle, the edge feature is the relative distance between the particles, and the v_a and φ of each time step system are the global feature. Then, we apply the MDGNet model to the input graph to predict order parameters. When the graph G is provided as input to the GN block, the update starts. A GN block contains six functions, namely three update functions and three aggregate functions. The choice of update functions directly determines the performance of the model. The network first uses three multilayer perceptrons to independently encode the input graph, one for each node, one for each edge, and one for the update global. The updating function is used to update the attributes of edges, nodes and global based on their connected edges and nodes. The aggregate function is a simple sum. Formally, this process is formulated as follows:

$$e_k{}' = \Phi^e(e_k, v_{rk}, v_{sk}, u), \tag{3}$$

$$\overline{e}_i{}' = \rho^{e\to v}(E_i'), \tag{4}$$

$$V_i' = \Phi^v(\overline{e}_i{}', v_i, u), \tag{5}$$

$$\overline{V}' = \rho^{v\to u}(V'), \tag{6}$$

$$\overline{e}' = \rho^{e\to u}(E'), \tag{7}$$

$$u' = \Phi^u(\overline{e}', \overline{v}', u), \tag{8}$$

where $e_k{}'$, V_i', and u' are the updated versions of V, E, and u, respectively. The process involves five steps.

(i) Update the edge according to its own characteristics, the index of the sending node and the receiving node, and the global state. Φ^e (Eq. (3)) is the update function of the edge.

(ii) Aggregate the edge features around each node as shown in Eq. (4). Here, $\rho^{e\to v}$ is applied to E_i, where $E_i{}' = \{(e_k{}', r_k, s_k)\}_{r_k=i,\ k=1:N^e}$. And $\overline{e}_i{}'$ is used to update the nodes in the next step.

(iii) Φ^v (Eq. (5)) is the update function for nodes, which updates the state of each node according to the edge features after aggregation, the features of the original node and the global graph state.

(iv) Aggregate the attributes of all edges and nodes by Eqs. (6) and (7). $\rho^{e\to u}$ and $\rho^{v\to u}$ are the aggregation function of all edges and the aggregation function of all nodes.

(v) According to the features of the aggregated nodes, the features after the aggregation, and the global features, the global update is realized, as shown in Eq. (8). Φ^u is the global update function. This completes the entire update process.

In general, it uses the current node information, edge information and global information to update the edge. With the new edge information, the node is updated. And the new node information and edge information are used to complete the global update. The update process ends. A flowchart of our learning process is outlined in Fig. 1. To make almost all particles in the box contribute to predicting the order parameter under any given condition, we set the GN block to loop 5 times.

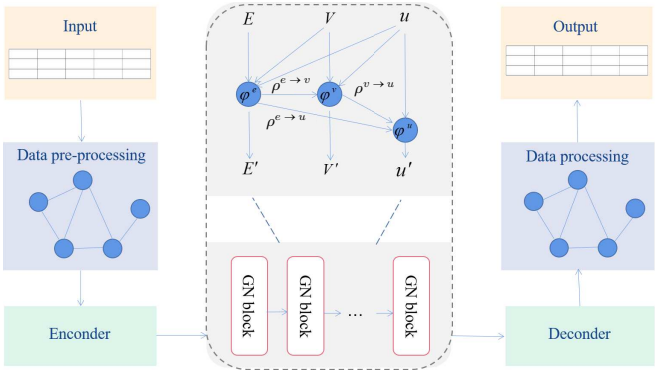


Fig. 1. The learning architecture is used in this work to predict system order parameters. Each GN block contains edge block, node block and global block, which can update edge, node and global, respectively.

In the prediction stage, the MDGNet model performance is measured through the relative error of individual predictions. This error is difined as

$$\Delta x = \left|\frac{x - x'}{x}\right|, \tag{9}$$

where x is the truth value of order parameter, x' is the corresponding prediction of the MDGNet model.

3. Results

3.1. Spatial distribution order parameter

According to the updated equation of the limited view angle model, it can be seen that the field of view angle is an important factor affecting the cluster evolution. Therefore, in order to study the effect of field of view angle on cluster evolution, we simulate the conditions of different field of view angles and obtain some meaningful results.

In Figs. 2(b) and 2(c), excluding the influence of noise density, we simulate the motion of particles under different viewing angles. When the noise is 0, after a long time of evolution, all particles can eventually reach a state where the direction of motion is consistently ordered.

The limited view angle model still uses the order parameter defined by Vicsek *et al.* to analyze the synchronization of the particle motion direction, which is represented by v_a, and its calculation formula is

$$v_a = \frac{1}{Nv}\left|\sum_{i=1}^{N} \boldsymbol{v}_i\right|, \tag{10}$$

where N is the total number of particles, v is the absolute velocity, and $\boldsymbol{v}_i$ is the velocity of the i-th particle, which is a vector. The normalized average speed is used to describe the degree of synchronization of the system. Obviously, $0 \leq v_{\mathrm{a}} \leq 1$. The more the average speed v of the whole system tends to 0, the more messy the whole system is. v_{a} tends to 1, indicating that the synchronization degree of the system is higher, and the movement direction of the particles tends to be the same. That is to say, the larger the value of v_{a}, the higher the synchronization degree of the system.

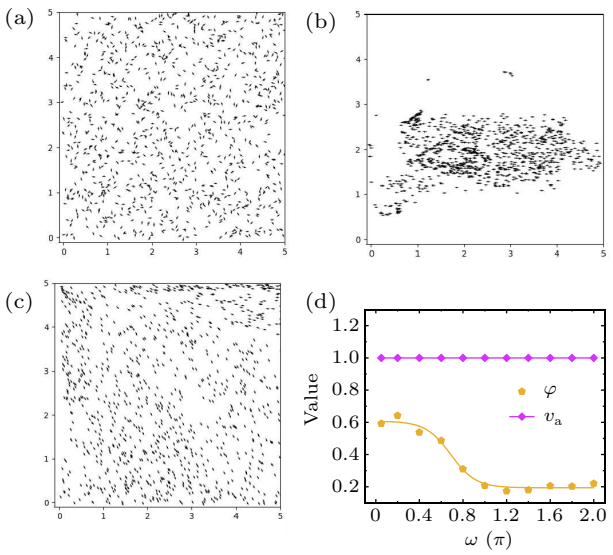


Fig. 2. (a) The initial configuration of the particles. (b) The configuration diagram after the system stabilizes when the viewing angle ω is 0.4π. (c) The configuration diagram after the system stabilizes when the viewing angle ω is 0.8π. (d) The spatial distribution order parameter at different field angles after the same degree of synchronization of particles in the system.

However, in the process of simulation, we find that the spatial distribution of particles is different when their motion is ordered in the same direction, as shown in Figs. 2(b) and 2(c). Although v_{a} can effectively describe the synchronization degree of particle motion, it cannot explain the spatial distribution of particles. But it is important for the study of spatial distribution. Deng *et al.* discovered the critical point of interference transition of the two-dimensional monodisperse by calculating the distribution of the local density.[42] Zhang *et al.* investigated the low-cost communication mechanism that breaks the state of uniformly distributed particles, with particles aggregating into increasingly tight clusters.[43] Therefore, in addition to analyzing the relationship between the synchronization index v_{a} and the viewing angle, it is also necessary to study the deviation of the particle distribution in the specified area caused by the viewing angle. How to describe the spatial distribution of particles in self-driven cluster motion is an important research question.

The side of the two-dimensional plane is L, so we divide the plane into $L \times L$ grids and count the number of particles in each grid. The number of particles in the i-th grid is represented by n_i, and $\overline{n}$ represents the average number of particles in each grid. The degree of non-uniformity of particles in the whole region is denoted by φ', where $\varphi' = \sum_{i=1}^{L^2} |n_i - \overline{n}|$. When all particles are concentrated in one lattice, φ' takes the maximum value of $2\overline{n}(L^2-1)$. When the particles are uniformly distributed, $\varphi' = 0$. We normalize φ' as

$$\varphi = \frac{\varphi'}{2\overline{n}(L^2-1)}, \tag{11}$$

obviously, $0 \leq \varphi \leq 1$. After the whole system is stable, all particles are concentrated in a grid, then $\varphi = 1$. And when all particles are uniformly distributed, then $\varphi = 0$. Because the particles are constantly moving, then n_i in Eq. (11) is changing, we average 50 sets of different initial configurations, and select the average of 800 to 1000 time steps as the stable stage. The value of φ is obtained as shown in Fig. 2(d). When the v_{a} of the particle's motion is the same, the value of φ is different. That is to say, the particles move in the same orderly direction and their spatial distribution is different. Then φ can effectively describe this difference.

It can be seen from Fig. 2(d) that the relation between the spatial distribution order parameter φ and view angle is not linear. It is not that the larger the view angle is, the larger φ will be. When the view angle reaches the threshold, the φ changes within a small range. The order parameter can effectively describe the transition of spatial distribution from disorder to order. When v_{a} is the same, the spatial distribution order parameter φ can effectively describe the variability of the spatial distribution of self-driven particle motion caused by the field of view angle. Under other conditions, the spatial distribution of the system has a similar situation.

To illustrate the effectiveness of the spatial distribution order parameter at multiple system sizes, we calculate φ of different system sizes. We divide the plane into 5×5 grids. For systems of different sizes, we gradually increase the amount of the view angle ω for a fixed noise and observe a transition from uniform distribution to non-uniform phase.

In Fig. 3, we show that the spatial distribution order parameter φ^* gradually decreases and reaches a stable value as the viewing angle increases. Although the distribution of points is different for different system sizes, the trend is the same. We make a normalization φ^* of the four different system sizes as

$$\varphi^* = \frac{\varphi - \varphi_{\mathrm{c}}(N)}{\varphi_{\max}(N) - \varphi_{\mathrm{c}}(N)}, \tag{12}$$

where $\varphi_{\mathrm{c}}(N)$ is the critical spatial distribution order parameter ($\omega \to 2\pi$) and $\varphi_{\max}(N)$ is the maximum value of φ in Eq. (12). For finite sizes, $\varphi_{\mathrm{c}}(N)$ and $\varphi_{\max}(N)$ are N dependent.

As can be seen in Fig. 3, the line has a significant downward when the viewing angle is about 0.7π. And it can be seen

that the view angle of the first uniform state of the system decreases as the number of particles increases. The simulations are carried out in a square shaped cell of linear size L with periodic boundary conditions. We vary the density of the system by changing the number of particles. In a system with relatively large density, particles can obtain sufficient information in a relatively small range, and the view angle of particles in the system to reach a uniform state is also reduced.

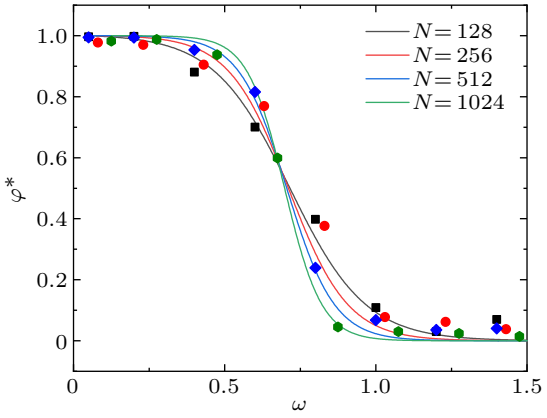


Fig. 3. The normalized value of the spatial distribution order parameter φ, for systems with number of particles N. Each dot shows the real value of φ, and the line shows the trend.

3.2. Collective motion prediction of MDGNet model

It is difficult to obtain the order parameters of the system using classical methods for large systems at each time-step. However, the MDGNet model we built can realize the long-term prediction of v_a and φ for large systems. Here we present the results from our deep learning model. Our model was trained on different vision.

In order to evaluate the accuracy of the prediction, we compare the predicted value with the true value. The results are summarised in Figs. 4 and 5 for v_a and φ predictions, respectively.

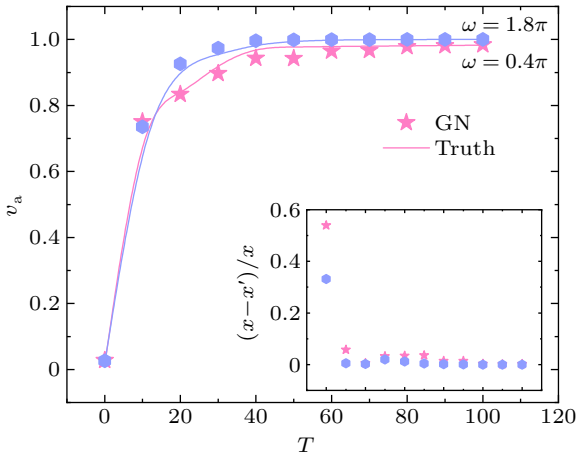


Fig. 4. The truth and prediction of v_a over time at different field angles, where other parameter values are $L = 5$, $R = 1$, $N = 1000$, $\eta = 0$. The symbols are the graph network predictions and the lines are the truth values of v_a. The inset contains the relative error between MDGNet model predictions and the truth values, and the symbols represent the same ω.

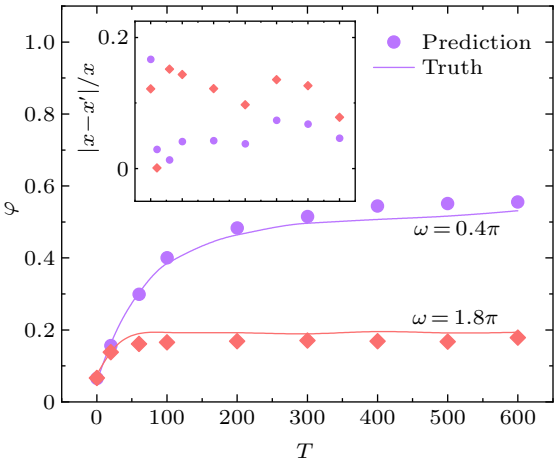


Fig. 5. The truth and prediction of φ over time at different field angles, where other parameter values are $L = 5$, $R = 1$, $N = 1000$, $\eta = 0$. The symbols are the graph network predictions and the lines are the truth values of φ. The inset contains the relative error between MDGNet model predictions and the truth values, and the symbols represent the same ω.

First, We fix other parameter conditions and use the MDGNet model to predict the variation of order parameter v_a with time step at different field angles. It can be seen from Fig. 4, although the size of visual field angle is different, v_a can reach 1 after a period of time under such condition. As we can see in Fig. 4, the time steps for the system to reach an orderly state are different under different view angles. Whether in a larger or smaller viewing angle, our graph network has a very good prediction effect to predict v_a. Then, in Fig. 5 we can see that after a period of movement, particles can eventually reach a relatively stable spatial distribution state. When the view angle is small, φ changes from 0 to 0.6, indicating that the particles in the whole system are relatively clustered in spatial distribution. When the view angle is larger, φ eventually stabilizes at about 0.2, and the spatial distribution is relatively uniform compared with that at a smaller field angle. Under different view angles, our model has good prediction effect.

Furthermore, in the inset of Figs. 4 and 5, the determination of relative error between the truth and predictions is shown. From Fig. 4, we can observe that the deviation of prediction results of other time steps lies within 5% except $T = 0$. Because the particles are randomly distributed at the initial moment, and they are uncertain. In Fig. 5, the predicted results are slightly different from the actual results, but within a reasonable range. Because the particles are moving, the number of particles in the grid is always changing at each time step. In summary, it is clear that the MDGNet model we built can predict order parameters. As can be observed in the figures, the agreement between the truth and prediction is quite good.

In addition, we predict the order parameters v_a and φ of the system synchronization phase under different noise η with fixed densities and field angles. Figure 6 demonstrates that the error of MDGNet prediction results increases relatively under

larger noise, but the prediction results are good from an overall perspective. Compared with other methods, the relative error of our method (MDGNet) is smaller than other machine learning (convolutional network (CNN), (SVM)). With only input types and relative positions, and without any handcrafted features, the MDGNet outperforms all the others methods at all noise.

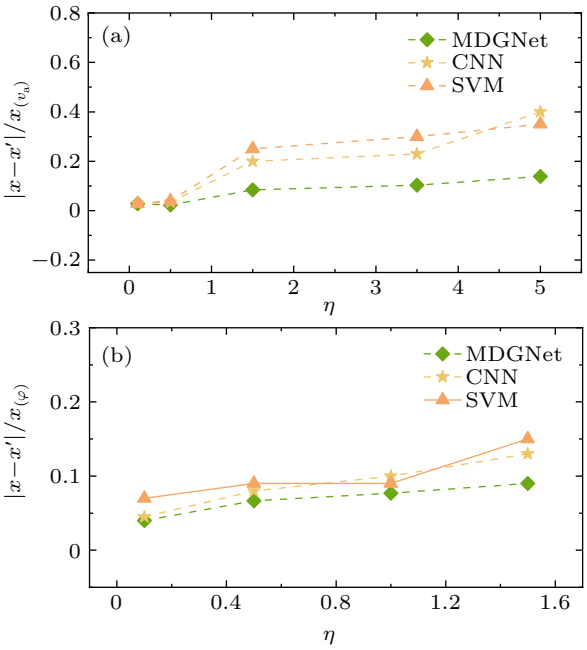


Fig. 6. (a) Relative errors of different methods for predicting order parameter v_a under different noises. (b) Relative errors of different methods for predicting order parameter φ under different noises.

4. Conclusions

We have numerically studied the dynamic behavior of self-propelled particles, and the view angle of each particle is limited. We have demonstrated that the view angle plays an important role in the phase behavior of self-driven particle systems. Interestingly, we have found that the spatial distribution of particles is different when the direction of motion is uniform and orderly at different view angles. We quantify the degree of order of spatial distribution by introducing a new order parameter φ. Our results indicate that the order parameter can effectively describe the transition of spatial distribution from disorder to order. In addition, the same conclusion could be obtained for systems with different sizes.

Furthermore, we have built a deep learning model based on the graph network to predict the order parameters of the self-propelled particle system. We show that the MDGNet model is an effective tool to predict the dynamic behavior of self-propelled particles. Because it can construct a graph structure for the position and direction information of the movement of the input particle and predict it at the graph level. The direct use of particle coordinates as input into the network underlies the robustness and simplicity of our approach. The deep learning model can accurately predict v_a and φ of the system.

Moving forward, we believe that the MDGNet model could be applied to more difficult prediction tasks that elude standard techniques. Specifically, we are interested in applying the MDGNet model to analyze other types of complex system. A natural next step involves using graph network to predict three-dimensional self-propelled particle systems, which could provide new methods for ordering these systems. This may require modification to the neural architecture and also the generation of new datasets with particles in three-dimensional. In conclusion, there is still great potential for applying deep learning methods to the prediction and classification of complex systems.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 11702289), Key core technology and generic technology research and development project of Shanxi Province of China (Grant No. 2020XXX013), and the National Key Research and Development Project of China.

References

[1] Köpf M H and Pismen L M 2013 *Soft Matter* **9** 3727
[2] Soma R, Nakayama B, Kuwahara M, Yamamoto E and Saiki T 2020 *Appl. Phys. Lett.* **117** 221601
[3] Keta Y E, Fodor é, van Wijland F, Cates M E and Jack R L 2021 *Phys. Rev. E* **103** 022603
[4] Reichhardt C and Reichhardt C J O 2021 *Phys. Rev. E* **103** 022602
[5] Cichos F, Gustavsson K, Mehlig B and Volpe G 2020 *Nat. Mach. Intell.* **2** 94
[6] Zhang J, Zhao Y, Tian B, Peng L, Zhang H T, Wang B H and Zhou T 2009 *Physica A* **388** 1237
[7] Zhang B and Shao Z G 2021 *Physica A* **563** 125382
[8] Das M, Schmidt C F and Murrell M 2020 *Soft Matter* **16** 7185
[9] Kumar S, Singh J P, Giri D and Mishra S 2021 *Phys. Rev. E* **104** 024601
[10] Jadbabaie A, Lin J and Morse A S 2002 *Proceedings of the* 41*st IEEE Conference on Decision and Control* **3** 2953
[11] Papadopoulou M, Hildenbrandt H, Sankey D W, Portugal S J and Hemelrijk C K 2022 *PLoS Comput. Biol.* **18** e1009772
[12] Hall B, Roggero A, Baroni A and Carlson J 2021 *Phys. Rev. D* **104** 063009
[13] Sarma S D, Deng D L and Duan L 2019 *Physics Today* **72** 48
[14] Vicsek T, Czirók A, Ben-Jacob E, Cohen I and Shochet O 1995 *Phys. Rev. Lett.* **75** 1226
[15] Tian B M, Yang H X, Li W, Wang W X, Wang B H and Zhou T 2009 *Phys. Rev. E* **79** 052102
[16] Mehta P, Bukov M, Wang C H, Day A G, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1
[17] Schütt K T, Chmiela S, von Lilienfeld O A, Tkatchenko A, Tsuda K and Müller K R 2020 *Lect. Notes Phys.* **968** 48
[18] Brunton S L, Noack B R and Koumoutsakos P 2020 *Ann. Rev. Fluid Mech.* **52** 477
[19] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[20] Bourilkov D 2019 *International Journal of Modern Physics A* **34** 1930019
[21] Zitnik M, Agrawal M and Leskovec J 2018 *Bioinformatics* **34** i457
[22] Papadopoulou M, Hildenbrandt H, Sankey D W, Portugal S J and Hemelrijk C K 2022 *PLoS Computational Biology* **18** e100977
[23] Zhang T, Song A and Lan Y 2020 *Scientia Sinica Informationis* **50** 347
[24] Brunton S L, Noack B R and Koumoutsakos P 2020 *Annual Review of Fluid Mechanics* **52** 477

[25] Shlomi J, Battaglia P and Vlimant J R 2020 *Machine Learning: Science and Technology* **2** 021001
[26] Jiang W and Luo J 2022 *Exp. Sys. Appl.* **207** 117921
[27] Zöttl A 2020 *Chin. Phys. B* **29** 074701
[28] Tsitsulin A, Palowitch J, Perozzi B and Müller E 2020 arXiv: 2006.16904
[29] Cao S, Sun X, Bo L, Wei Y and Li B 2021 *Inform. Software Technol.* **136** 106576
[30] Ulices Q S, Pedro E R G and Alexis T C 2021 *Soft Matter* **17** 1975
[31] Pata J, Duarte J, Vlimant J R, Pierini M and Spiropulu M 2021 *Eur. Phys. J. C* **81** 1
[32] Zhang J, Lei Y K, Zhang Z, Chang J, Li M, Han X, Yang L, Yang Y I and Gao Y Q 2020 *J. Phys. Chem. A* **124** 6745
[33] Battaglia P W, Pascanu R, Lai M, Rezende D and Kavukcuoglu K 2016 arXiv: 1612.00222
[34] Dulaney A R and Brady J F 2021 *Soft Matter* **17** 6808
[35] Wang R, Fang F, Cui J and Zheng W 2022 *Sci. Rep.* **12** 1
[36] Hachijo T, Gotoda H, Nishizawa T and Kazawa J 2020 *J. Appl. Phys.* **127** 234901
[37] Ganaie M, Ghosh S, Mendola N, Tanveer M and Jalan S 2020 *Chaos* **30** 063128
[38] Bhaskar D, Manhart A, Milzman J, Nardini J T, Storey K M, Topaz C M and Ziegelmeier L 2019 *Chaos* **29** 123125
[39] Battaglia P W, Hamrick J B, Bapst V, *et al.* 2018 arXiv: 1806. 01261
[40] Bapst V, Keck T, Grabska-Barwińska A, Donner C, Cubuk E D, Schoenholz S S, Obika A, Nelson A W, Back T, Hassabis D, *et al.* 2020 *Nat. Phys.* **16** 448
[41] Li Z and Farimani A B 2022 *Computers & Graphics* **103** 201
[42] Deng L, Zhao C, Xu Z and Zheng W 2020 *Eur. Phys. J. E* **43** 1
[43] Zhang H T, Chen M Z, Zhou T, Cheng Z and Yu P Z 2009 *Complex Sci.* **5** 2159