

Graph dynamical networks for forecasting collective behavior of active matter

Yanjun Liu(刘彦君), Rui Wang(王瑞), Cai Zhao(赵偲), and Wen Zheng(郑文)

Citation : Chin. Phys. B, 2022, 31 (**11**): 116401. DOI: 10.1088/1674-1056/ac7cce

What follows is a list of articles you may be interested in

Nonvanishing optimal noise in cellular automaton model of self-propelled particles

Guang-Le Du(杜光乐) and Fang-Fu Ye(叶方富)

Chin. Phys. B, 2022, 31 (**8**): 086401. DOI: 10.1088/1674-1056/ac67c4

Ionospheric vertical total electron content prediction model in low-latitude regions based on long short-term memory neural network

Tong-Bao Zhang(张同宝), Hui-Jian Liang(梁慧剑),Shi-Guang Wang(王时光), and Chen-Guang Ouyang(欧阳晨光)

Chin. Phys. B, 2022, 31 (**8**): 080701. DOI: 10.1088/1674-1056/ac4487

Memristor-based multi-synaptic spiking neuron circuit for spiking neural network

Wenwu Jiang(蒋文武), Jie Li(李杰), Hongbo Liu(刘洪波), Xicong Qian(钱曦聪), Yuan Ge(葛源), Lidan Wang(王丽丹), and Shukai Duan(段书凯)

Chin. Phys. B, 2022, 31 (**4**): 040702. DOI: 10.1088/1674-1056/ac380b

Metastable phase separation and rapid solidification of undercooled $Co_{40}Fe_{40}Cu_{20}$ alloy

Xiaojun Bai(白晓军), Yaocen Wang(汪姚岑), Chongde Cao(曹崇德)

Chin. Phys. B, 2018, 27 (**11**): 116402. DOI: 10.1088/1674-1056/27/11/116402

Subsolidus phase relation in the Bi_2O_3–Fe_2O_3–La_2O_3 system

Hu Qi-Chang (胡启昌), Chen Ye-Qing (陈业青), Lü Pei-Wen (吕佩文), Huang Feng (黄丰), Wang Xian (王娴)

Chin. Phys. B, 2014, 23 (**2**): 026402. DOI: 10.1088/1674-1056/23/2/026402

Graph dynamical networks for forecasting collective behavior of active matter

Yanjun Liu(刘彦君)[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* 030060, *China*
[2] *Center of Information Management and Development, Taiyuan University of Technology, Taiyuan* 030060, *China*
[3] *Center for Healthy Big Data, Changzhi Medical College, Changzhi* 046000, *China*

(Received 18 February 2022; revised manuscript received 13 June 2022; accepted manuscript online 29 June 2022)

After decades of theoretical studies, the rich phase states of active matter and cluster kinetic processes are still of research interest. How to efficiently calculate the dynamical processes under their complex conditions becomes an open problem. Recently, machine learning methods have been proposed to predict the degree of coherence of active matter systems. In this way, the phase transition process of the system is quantified and studied. In this paper, we use graph network as a powerful model to determine the evolution of active matter with variable individual velocities solely based on the initial position and state of the particles. The graph network accurately predicts the order parameters of the system in different scale models with different individual velocities, noise and density to effectively evaluate the effect of diverse condition. Compared with the classical physical deduction method, we demonstrate that graph network prediction is excellent, which could save significantly computing resources and time. In addition to active matter, our method can be applied widely to other large-scale physical systems.

Keywords: active matter, graph network, improvement of Vicsek, collective motion

PACS: 64.75.–g, 07.05.Mh, 05.65.+b, 87.64.Aa **DOI:** 10.1088/1674-1056/ac7cce

1. Introduction

Self-driven clustering is a phenomenon in which individuals form a collective force with regular movement. Individuals cooperate and influence each other to accomplish complex, collective tasks in a self-organized manner that cannot be accomplished by individuals. Individuals will overcome extreme environments[1–5] through collective behaviors of avoiding natural predators, fighting danger, foraging for survival, and collaborating. Such clustered movements are universal in natural groups of microorganisms, insects, animals, etc. (e.g., microbial clusters, schools of fish, flocks of birds, and so on). Each individual in such a group with the ability to drive themselves is known as "active matter".[6,7] They generate complex non-equilibrium kinetic phenomena[8,9] that cause processes of phase transition[10] in the system. Extending to human society, the dynamical behavior of active matter in the control coordination and formation control of robot[11] populations, micro-systems, and autonomous multi-intelligent systems[12] helps to explain[13] the generation and change patterns of cluster intelligence behavior. Therefore, how to describe and explore cluster dynamics using theoretical models becomes a crucial proposition.

Among many models for studying cluster motion, the "Vicsek model"[14] is the most classical model with a simple algorithm, which can simulate the natural cluster synchronization phenomenon[15,16] more realistically. The model quantifies[17] the phase transition phenomena in different control modes by controlling the density and noise[18] of particles in the system.[19] Since the introduction of the Vicsek model, scholars in different fields have used it as a base model for improvement and application. The model is gradually diversified and complicated by getting rid of idealization and getting closer to the actual situation of collective motions.[20,21]

However, it is increasingly difficult to use existing dynamical simulation methods to derive the increasingly complex "Vicsek model":[15] the derivation period is lengthy and the calculation process is complicated when calculating the quantitative criteria of the model, namely, the order parameters. It is also limited to small-scale systems and cannot be extrapolated efficiently to realistic large-scale conditional systems. Nowadays, modern machine learning methods and massive data sets have been widely used in multiple fields[22–24] with increasing capabilities[25,26] of recognition, classification, training and prediction. It can effectively improve the speed and accuracy of problem solving, and is expected to solve some problems that are difficult and impossible to solve by existing inference methods. Therefore, using machine learning techniques[27] and massive datasets to explore classical physics theory[28,29] problems has become popular research currently.

Among many machine learning methods, this study attempts to use the graph network[30] model for learning and prediction. This model has developed from the graph neu-

†Corresponding author. E-mail: zhaocai@tyut.edu.cn
‡Corresponding author. E-mail: zhengwen@tyut.edu.cn

© 2022 Chinese Physical Society and IOP Publishing Ltd

ral networks, combining neural network methods with mathematical methods. It provides a more comprehensive and detailed description of graph network and their inference capabilities and extends and generalizes a multitude of neural network methods. Since the emergence of the graph neural networks in the last decade or so, they have been continuously extended and strengthened[31,32] to involve problems in different domains.[33,34] They have been extended to efficiently solve classical problems[35] using various supervised and reinforcement learning methods. In the physical field, graph neural networks are also gradually becoming involved in studying the physical systems,[36,37] cluster system dynamics,[38,39] active matter,[39–41] etc. The DeepMind team has integrated graph neural networks to propose a "graph network model" that is more efficient than other standard machine learning building blocks[42,43] and has been utilized successfully in many models of physically complex systems.[39,43] Thus, graph network can replace the existing molecular dynamics derivation process to study cluster dynamics specific processes through predictive methods.[44,45]

This study aims to apply a novel machine learning approach to a classical physics problem.[46] We hope to obtain the final state of cluster motion at a specified time step using the predictive capability of graph network models with only the input of initial bit patterns. By this method, we can avoid the complex extrapolation process, effectively save the computational cost, and improve the computational efficiency of the traditional physics models.[47,48] In this study, we construct the graph structure of topological relationships between individuals and neighbors, and predict the sequential covariates by building a graph network model. We analyze the variation in the degree of consistency of the active matter system during the phase transition under different conditions.

2. GN-active model framework

The graph network (GN) model is designed in the form of "end-to-end",[42] the input and output are both with graph structures that have the same number of nodes and edge structures. Each time we input a set of graphs with the same condition as system parameter. The initial bit pattern is randomly generated and will be different each time. Through training, it is possible to predict the evolution of cluster motion under system conditions. Graph is represented by the triple $G=(E,V,u)$, E is the set of edge (reflecting the influence of individuals), $E=\{(e_k,r_k,s_k)\}_{k=1:N^{\mathrm{e}}}$; V is the set of nodes (individuals), $V=\{v_i\}_{i=1:N^{\mathrm{v}}}$; u is the global state attribute; N^{v} is the number of nodes; and N^{e} is the number of edges (e_k represents the attribute of the k-th edge, r_k and s_k are respectively the receiving node and the sending node, which are the indexes of nodes connected by the k-th edge).

In the Vicsek model, the graph is constructed by visualizing the individual information embedded in the triad. Individuals of active matter are the nodes, containing the speed of motion of active particles and their own positions; the distance and correlation data between particles exist on the edges, and the rate of motion and positions of individuals in the system form the global state. In this study, the number of particles is fixed. Each individual takes itself as the center of the circle, and the individual particle within the circle area of the specified radius is its neighbor node. The correlation between nodes builds edges. The order parameters can be calculated by using the velocity directions of all the particles in the output system, and the global state can be quantitatively expressed.

The computational core architecture of the GN-active model consists of three modules: encoder (GN_encoder), core computing block (GN_core) and decoder (GN_decoder). The GN-active model first uses the encoder called GN_encoder to encode the input data into particle position, velocity of motion and global parameters (the order parameter that can quantify the degree of system consistency; the predicted model particle number, parameter β, noise value, system particle density and other related parameters), and compresses the input graph into a potential spatial representation. It is then processed by a GN_core block, which is the core computing part of the graph network. The GN_core block is cycled N times and the information is propagated from a single particle in the Vicsek model to the whole graph, with each cycle achieving one update of the whole graph. The result is decoded by the third module GN_decoder and reconstructs the potential spatial representation into another graph. The result still outputs a graph structure with position and velocity information. The specific process of the GN-active model is shown in Fig. 1(a).

In the graph update process of a GN_core block, the computation module is divided into three parts: edge, node, and global. Blocks are closely related, and the calculation result of the previous block will become the input of the subsequent block.

During the graph update process, the calculation unit includes six internal functions: three update functions and three aggregate functions. These six functions would update edges firstly, then update nodes, and finally use the updated edges and nodes to update the state of the whole graph, thus completing a graph update. Update function:

$$e'_k=\Phi^{\mathrm{e}}\left(e_k,v_{r_k},v_{s_k},u\right), \tag{1}$$

$$v'_i=\Phi^{\mathrm{v}}\left(\bar{e}'_i,v_i,u\right), \tag{2}$$

$$u'=\Phi^{\mathrm{u}}\left(\bar{e}',\bar{v}',u\right). \tag{3}$$

Aggregation function:

$$\bar{e}'_i=\rho^{\mathrm{e}\to\mathrm{v}}\left(E'_i\right), \tag{4}$$

$$\bar{e}'=\rho^{\mathrm{e}\to\mathrm{u}}\left(E'\right), \tag{5}$$

$$\vec{v}' = \rho^{\mathrm{v}\to\mathrm{u}}\left(V'\right). \tag{6}$$

The detailed process of the whole update is shown in Fig. 1(b), that is, to realize an update of the graph.

After input the triple structure data of graph $G=(E,V,u)$, the edge block calculates each edge through the edge update function Φ^{e} (formula 1) and outputs e'_k. The set of e'_k is E'. All edges are polymerized to $\bar{e}'$ by the aggregation function $\rho^{\mathrm{e}\to\mathrm{u}}$ (formula 5), which will work when global status updates.

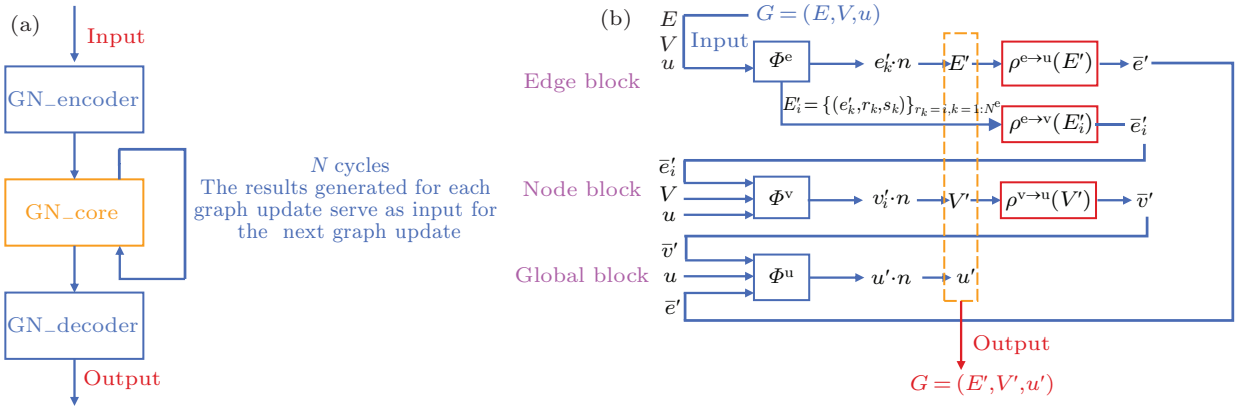


Fig. 1. (a) The core architecture of the GN-active model. The model is divided into three blocks: GN_encoder, GN_core, and GN_decoder. GN_encoder will receive and process input, GN_core module will carry out the circular calculation, and GN_decoder will reconstruct the calculation results and output. (b) The GN_core update process of the GN-active model. The update process is divided into three blocks: edge block, node block, and global block, which are node update, edge update, and global update, respectively. The update operation is carried out through the update function and aggregation function.

Node block aggregates all the directed edges pointing to a vertex to $\bar{e}'_i$ using the aggregation function $\rho^{\mathrm{e}\to\mathrm{v}}$ (formula 4), forming the set $E'_i = \left\{\left(e'_k, r_k, s_k\right)\right\}_{r_k=i,k=1:N^{\mathrm{e}}}$ of all the directed edges pointing to the node with index i. Each node using $\bar{e}'_i$ by updating function Φ^{v} (formula 2) to calculate and output the v'_i, v'_i collection of V', all nodes are aggregated into $\bar{v}'$ by the aggregation function $\rho^{\mathrm{v}\to\mathrm{u}}$ (formula 6).

In the global block, the global attribute will be evaluated by the update function Φ^{e} (formula 3) to yield u', where GN outputs the updated set of all edges, nodes, and global attributes $G'=(E',V',u')$.

GN-active model is improved according to the format of the datasets, which can process the Vicsek model data after the optimization of individual speed. Construct the input graph data based on the location direction of active matter. In the whole process of model construction, the input data is first preprocessed into a graph structure, which is compiled by GN_encoder. The working principle of the graph network is shown by Fig. 1, in which the GN_core cycle update is setup 7 times in order to make the information propagate to the individuals at the edge completely. Finally, the network decodes the generated embedding into the predicted v_{a}, which can quantitatively represent the degree of order of the system. After completing the model training, the sequential covariance prediction is specified for the test set at the time step. The output predictions are sorted and plotted to show the trend of sequential parametric changes at the specified time step during convergence.

3. Results

3.1. Datasets

In fact, in many real clustered systems the individual speed is not constant. In order to further fit the reality, the standard Vicsek model needs to be optimized for individual speed, i.e., individual speed that changes with adjustable parameter; while other conditions remain unchanged, determined the individual's own speed by the environment around. The velocity is adjusted according to the local order parameters of the neighbor individual's direction of motion, so that the Vicsek model is no longer the simplest and most ideal model.[42]

To effectively measure the orderliness of the system, we still refer to the order parameter to reflect the consistency degree of the whole system. The order parameter can be obtained according to

$$v_{\mathrm{a}} = \left| \frac{\sum_{i=1}^{N} \boldsymbol{v}_i}{\sum_{i=1}^{N} |\boldsymbol{v}_i|} \right|, \tag{7}$$

where $\boldsymbol{v}_i$ is the individual velocity of the active matter. In this improved model, the speed and direction of each individual are variable in each time step; v_{a} is between 0 and 1, and we only consider the direction of velocity. When $v_{\mathrm{a}}=1$, the active matters in the system are ordered. When $v_{\mathrm{a}}=0$, the system is unordered (random direction).

In the standard Vicsek model, the velocities of individual particles only change direction. Different from the standard Vicsek model, the speed and direction of individual particles are variable and affected by local order parameters. The updating formula to determine the speed is[12]

$$v_i(t+1) = v_{\max} \cdot \mathrm{e}^{\beta\left[v_{\mathrm{a}}^{i}(t)-1\right]}, \tag{8}$$

where v_a^i is the local order parameter within the circular range taken by the active particle with itself as the center and fixed radius R, which is used to quantify the consistency of local particles. The parameter β can adjust the variation trend of individual speed. When $\beta = 0$, formula (8) is consistent with the Vicsek standard model (v_{max}). When $\beta > 0$, the local consistency of the individual is higher, the particle moves faster, which promotes faster synchronization of the system.

Thus, the individual speed can basically reflect the degree of order within the specified radius. In formula (8), when the degree of consistency is high, the $v_i(t+1)$ tends to v_{max}, and the individual moves at the maximum speed set. When the degree of consistency is low, the $v_i(t+1)$ tends to 0, and the individual basically lingers.

We use python to design iterative programs to calculate the sequential covariates within a specified radius area centered on each individual. Thus, the speed at the next time step of that individual is updated. We designed simulations of $N = 300$ particles in an $L \times L$ region under two-dimensional periodic boundary conditions for active matter dynamics. The initial velocity of each individual is set as 0.03, the parameter β takes 0, 0.1, 0.5, 1, 2, 5, 10, noise η is 0, 1, 2, 3, 4, 5, 6, and the length of the simulated box side L is 5, 10, 15, 25. By setting different environmental parameters, the advantages and disadvantages of the prediction results under different circumstances have been analyzed.

We have trained the models under these environmental conditions, generating 100 independent datasets for each of these conditions as training datasets, and regenerating 100 datasets as test datasets for evaluation of the models. Under each condition, there are 100 sets of training data and 100 sets of prediction data, each group will have T time steps, and the X and Y axis position and direction data of N particles. Meanwhile, we also calculated the system order parameters of each time step.

3.2. Optimal parameter β

In order to observe the effect of parameter β and to analyze its influence, we observe the individual motion of specified time and step under different β, and look for the direct influence of β changes on the cluster.

As can be seen from Fig. 2, when the value of β is larger, the order of magnitude of the difference between the velocity of individuals with high and low velocities is greater. It can be observed that individuals with high-speed show longer lengths in the configuration diagram, and individuals with low speed are closer to the "node" shape. Therefore, it can be seen that their speed is approaching 0, and the individual is in a "hesitation" state.

At the same time, with the increase of β value, there will be more clusters moving in different directions. These clusters will easily change each other's direction when they meet in the process of moving. The time required for the consistent convergence of the whole system is longer. At the same time, the larger the β value is, the more difficult it is for the curvilinear motion to appear in the system. It can even be observed that the two cluster directions are opposite to each other. The direction of the cluster with a small number of individuals will be changed, and the individuals in the area where the two clusters conflict in the first place will gradually slow down to 0, and then move back into the larger cluster.

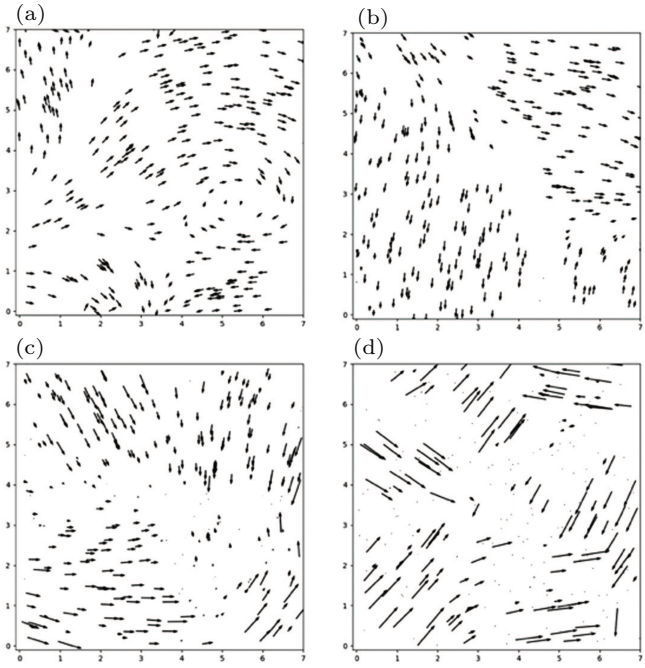


Fig. 2. Motion configuration rendering of the Vicsek model for individual speed optimization. Set $L = 7$, $N = 300$, $\eta = 0.1$, and $R = 1$ as the condition is fixed when the β values are 1, 5, 10, and 20, respectively. In the period before convergence, the configuration diagram at the 10-th step has selected to observe the influence of β change on the configuration change. (a) The configuration diagram of step 10 under the condition of $L = 7$, $N = 300$, $R = 1$, $\eta = 0.1$, $\beta = 1$. (b) The configuration diagram of step 10 under the condition of $L = 7$, $N = 300$, $R = 1$, $\eta = 0.1$, $\beta = 5$. (c) The configuration diagram of step 10 under the condition of $L = 7$, $N = 300$, $R = 1$, $\eta = 0.1$, $\beta = 10$. (d) The configuration diagram of step 10 under the condition of $L = 7$, $N = 300$, $R = 1$, $\eta = 0.1$, $\beta = 20$.

In order to control the variable β in the subsequent prediction and analyze the effects of other parameters on the system, the effects of different β values on the convergence of the system need to be defined first.

Figure 3 shows that with the increase of β, we calculated and recorded the time steps for completing convergence under different β conditions and draw this graph accordingly. In this simulation process, the average value of the order parameter v_a in the last 200 time steps is obtained from the 1000 time steps after the curve is stable. When the v_a of the system reaches this mean value for the first time in the process of moving, it is the "convergence time step". The convergence time T decreases first and then increases, and the best β makes the convergence

speed fastest and the convergence time shortest. In this graph, β, which makes convergence fastest, exists at about 1.0.

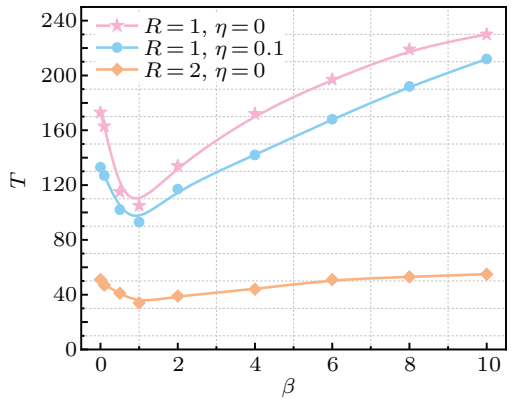


Fig. 3. Finding the optimal parameters β. Set $L=7$, $\eta=0.1$, $R=1$, change η and β, compare and observe the change of the step value when convergence.

Therefore, we fixed $\beta=1$, based on the fastest convergence speed, generated the initial configuration under different conditions, and transformed it into a graph network. Then we carried out the N times of the graph update process in the model and got the prediction results under different parameters.

3.3. Specific applications of prediction

3.3.1. Parameter β affecting speed

In order to observe the prediction ability of the model, we first use the datasets with different β values to conduct training prediction, compare the prediction with the true value curve, and explore the prediction effect after changing parameter β.

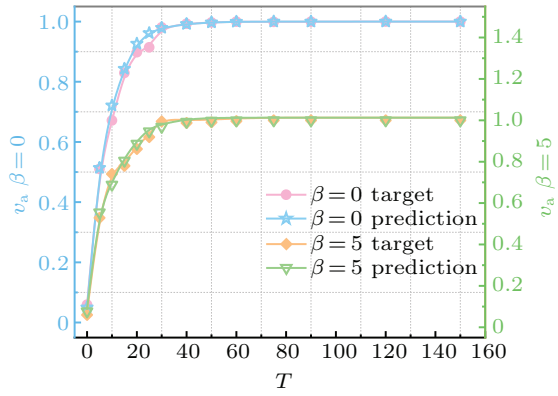


Fig. 4. Parameter β simulation results. Under the condition of changing parameter β, the blue line is the actual value of the combined velocity v_a, and the orange line is the predicted value of the combined velocity v_a. (a) $N=300$, $L=5$, $R=1$, $\eta=0$, $\beta=0$, which is the comparison between the predicted and the true value under the standard Vicsek model. (b) The comparison of $N=300$, $L=5$, $R=1$, $\eta=0$, $\beta=5$. Curve is the fitting of v_a values obtained when the time step takes 0, 5, 10, 15, 25, 30, 35, 40, 45, 50, 55, 60, 75, 90, 120, and 150.

According to the curve comparison in Fig. 4, the prediction effect is still excellent even if the β value changes to a certain extent. The difference of β leads to different convergence efficiency of the system. Under the same noise condition, even though the convergence time is obviously different, the order parameter v_a after convergence tends to the same value.

3.3.2. Analyzing the noise η

By changing the noise η, the influence of variation of the parameter noise on the accuracy of the prediction results of the graph network has been observed. In the datasets parameter design, the control variable β should be discussed at 1.0 when the value of β is no longer the variable in question.

As can be seen from the curve comparison in Fig. 5, when η is smaller, the fitting degree of the two curves is higher. The prediction curve after increasing η value has obvious fluctuation, while the prediction effect is still excellent: the prediction curve is still in the fluctuation range, and the fluctuation range is much smaller than the true value curve. With the increase of η, the order degree of the system will be greatly affected, so the accuracy of the prediction results will be greatly affected.

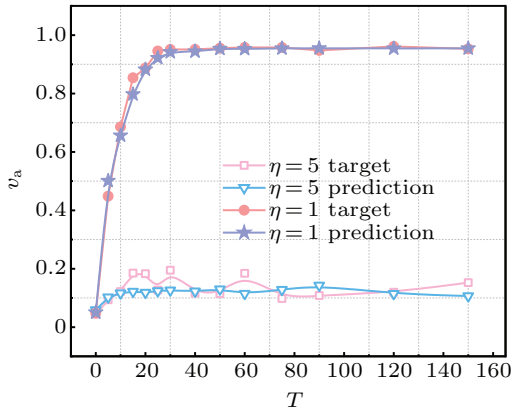


Fig. 5. Noise variation simulation results. Set $N=300$, $L=5$, $R=1$, $\beta=1$, change the predicted and true values of the noise $\eta=1$ and $\eta=5$ under the condition, and take the fitting of v_a values obtained at the time steps of 0, 5, 10, 15, 25, 30, 35, 40, 45, 50, 55, 60, 75, 90, 120, and 150.

3.3.3. Adjusting the density ρ

In order to further verify the predictive power of the model, the system region side length L has been changed to vary the individual density in the system (Fig. 6). We hope to observe the ability of prediction and other interesting phenomena in this way.

In Fig. 6 it can be observed under the condition that the noise η and parameter values are fixed, the prediction curve gets influenced by side length L changing. When L gradually becomes larger, and the number of individuals existing in the system does not change, the density is reduced gradually. This may show that the neighborhood, around which each individual will reduce communication between individuals, will be greatly affected, leading to a much slower convergence speed and significantly increased convergence time. Also, the increase of L reduces the density of the system and leads to its instability. In that case, the prediction accuracy is reduced, but the overall prediction accuracy of the graphical neural network still stays at a high level.

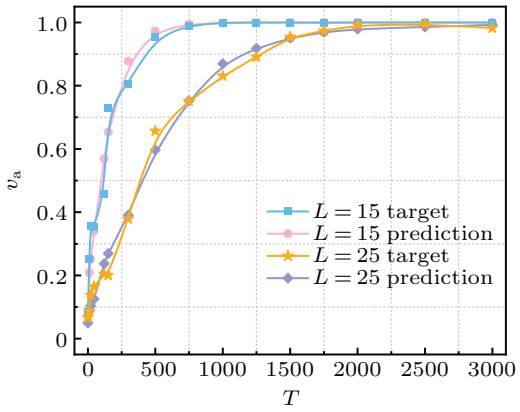


Fig. 6. The simulation results in density change by modifying side length L. Under the fixed conditions $N = 300$, $\eta = 0$, $\beta = 1$, the curve of v_a convergence between the real value and the predicted value has been observed.

4. Conclusions

Our simulation results show that the graph network constitutes a powerful tool to predict the long-term dynamics of active matter, leveraging some of the structure hidden in the local neighborhood of particles. The graph structure is used to describe the relationship between individuals. The graph network trains the model by learning the process of particle position and orientation changes in the system to predict the effect of different environmental settings on the global attribute order parameter v_a and to find the optimal influence parameters. Thus, the influence of global attribute order parameter v_a under different environment settings is predicted, and the best influence parameter is found. We hope that through this method, machine learning can be used to complete the deduction and prediction of large-scale physical systems, and can even be extended to the prediction and interpretation of systems under special conditions.

Active matter systems can generate large and high-quality datasets, which can correspond to complex but controllable physical phenomena that do not require any prior on the relevant physical quantities of the underlying system. Vicsek is an idealized model. Based on this model, this study optimizes the individual velocity close to the reality, so as to make a more realistic and reasonable explanation for the results. We hope to set up more improvements to the Vicsek model to make it more realistic. Datasets with different parameters are designed for training and prediction, and the influences of different variables on prediction results are explored. Each training starts from the input end to the specified time step as the output end, so continuity inference cannot be carried out. Therefore, it is still necessary to deepen the understanding and improvement of the GN-active model and explain the internal process of this "end-to-end" model. At the same time, it is also hoped that new quantitative parameters can be found in the research process to judge the cluster state in the system phase transformation process from different angles. In the process of studying and improving the graph network model, we will apply this model to other complex physical systems and predict more special and large-scale system inference effects.

References

[1] Genkin M M, Sokolov A, Lavrentovich O D and Aranson I S 2017 *Phys. Rev. X* **7** 011029
[2] Jolles J W, Boogert N J, Sridhar V H, Couzin I D and Manica A 2017 *Current Biology* **27** 2862
[3] Bajec I L and Heppner F H 2009 *Animal Behaviour* **78** 777
[4] Dombrowski C, Cisneros L, Chatkaew S, Goldstein R E and Kessler J O 2004 *Phys. Rev. Lett.* **93** 098103
[5] Kumar S, Singh J P, Giri D and Mishra S 2021 *Phys. Rev. E* **104** 024601
[6] Cichos F, Gustavsson K, Mehlig B and Volpe G 2020 *Nat. Mach. Intell.* **2** 94
[7] Keta Y E, Fodor van Wijland F, Cates M E and Jack R L 2021 *Phys. Rev. E* **103** 022603
[8] Zhang J, Zheng W, Zhang S, Xu D, Nie Y, Jiang Z and Xu N 2021 *Sci. Adv.* **7** eabg6766
[9] Speck T 2020 *Soft Matter* **16** 2652
[10] Chepizhko O, Saintillan D and Peruani F 2021 *Soft Matter* **17** 3113
[11] Zhang J, Zhao Y, Tian B, Peng L, Zhang H T, Wang B H and Zhou T 2009 *Physica A* **388** 1237
[12] Chen D 2018 *Research on Evolutionary Analysis and Pattern Control of Collective Dynamic System* (Ph.D. thesis) (Wuhan: Huazhong University of Science and Technolog) (in Chinese)
[13] Shankar S, Souslov A, Bowick M J, Marchetti M C and Vitelli V 2020 arXiv:2010.00364 [cond-mat.soft]
[14] Vicsek T, Czirók A, Ben-Jacob E, Cohen I and Shochet O 1995 *Phys. Rev. Lett.* **75** 1226
[15] Meakin P, Vicsek T and Family F 1985 *Phys. Rev. B* **31** 564
[16] Boccaletti S, Latora V, Moreno Y, Chavez M and Hwang D U 2006 *Phys. Rep.* **424** 175
[17] Liu B, Pu Z, Wu S, Shi L, Wang L and Yang W 2022 *Improved Self-Propelled Swarms Model with Enhanced Convergence Efficiency: Advances in Guidance, Navigation and Control* (Berlin: Springer) pp. 4933–4942
[18] Chen Q S and Ji M 2017 *Chin. Phys. B* **26** 098903
[19] Groß]mann R, Aranson I S and Peruani F 2020 *Nat. Commun.* **11** 1
[20] Czirók A, Stanley H E and Vicsek T 1997 *J. Phys. A: Math. Gen.* **30** 1375
[21] Zhang H T, Chen M Z and Zhou T 2007 arXiv:0707.3402 [physics.data-an]
[22] Hou F, Zhang Y, Fu X, Jiao L and Zheng W 2021 *J. Adv. Transport.* **2021** 9513170
[23] Willard J, Jia X, Xu S, Steinbach M and Kumar V 2020 arXiv:2003.04919 [physics.comp-ph]
[24] Zhang Y, Ren J, Wang R, Fang F and Zheng W 2021 *Water* **13** 2095
[25] LeCun Y, Bengio Y. and Hinton G 2015 *Nature* **521** 436
[26] Sarker I H 2021 *SN Comput. Sci.* **2** 1
[27] Mahesh B 2020 *Int. J. Sci. Res.* **9** 381
[28] Chen K and Meng X 2020 *J. Comput. Res. Develop.* **57** 1971 (in Chinese)
[29] Xu R 2018 *Application of Machine Learning to Phase Transition Problems* (Master thesis) (Xiamen: Xiamen University) (in Chinese)
[30] Zöttl A 2020 *Chin. Phys. B* **29** 074701
[31] Canabarro A, Fanchini F F, Malvezzi A L, Pereira R and Chaves R 2019 *Phys. Rev. B* **100** 045129
[32] Shchur O, Mumme M, Bojchevski A and Günemann S 2018 arXiv:1811.05868 [cs.LG]
[33] Zhang C, Song D, Huang C, Swami A and Chawla N V 2019 *Heterogeneous Graph Neural Network: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '19 (New York, USA: Association for Computing Machinery) pp. 793–803
[34] Xu Y and Xiaodong Q 2020 *Data Analysis and Knowledge Discovery* **4** 119 (in Chinese)
[35] Deng L, Bing G and Wen Z 2021 *International Journal of Web Services Research (IJWSR)* **18** 63
[36] Chang M B, Ullman T, Torralba A and Tenenbaum J B 2016 arXiv:1612.00341 [cs.AI]

[37] Zhang T, Song A and Lan Y 2020 *Sci. Sin. Inf.* **50** 347

[38] Reinhard P G and Suraud E 2008 *Introduction to Cluster Dynamics* (New York: John Wiley & Sons)

[39] Dulaney A R and Brady J F 2021 *Soft Matter* **17** 6808

[40] Wang R, Fang F, Cui J and Zheng W 2022 *Sci. Rep.* **12** 500

[41] Wang R, Cui J, Zhang Y and Zheng W 2021 *J. University of Electronic Science and Technology of China* **50** 768 (in Chinese)

[42] Battaglia P W, Hamrick J B, Bapst V, *et al.* 2018 arXiv:1806.01261 [cs.LG]

[43] Maron H, Ben-Hamu H, Serviansky H and Lipman Y 2019 arXiv:1905.11136 [cs.LG]

[44] Que-Salinas U, Ramírez-Gonzáez P E and Torres-Carbajal A 2021 *Soft Matter* **17** 1975

[45] Dulaney A R and Brady J F 2021 *Soft Matter* **17** 6808

[46] Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J and Battaglia P 2020 arXiv:2002.09405 [cs.LG]

[47] Zheng W, Zhang S and Xu N 2018 *Chin. Phys. B* **27** 066102

[48] Xu Z, Wang R, Cui J, Liu Y and Zheng W 2021 *Chin. Phys. B* **30** 066101