# Supervised, semisupervised, and unsupervised learning of the Domany-Kinzel model

Kui Tuo,[1] Wei Li,[1,*] Shengfeng Deng,[2,†] and Yueying Zhu[3]

[1]*Key Laboratory of Quark and Lepton Physics (MOE) and Institute of Particle Physics, Central China Normal University, Wuhan 430079, China*

[2]*School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710061, China*

[3]*Research Center of Applied Mathematics and Interdisciplinary Science, Wuhan Textile University, Wuhan 430073, China*

(Received 26 September 2023; revised 28 January 2024; accepted 9 July 2024; published 2 August 2024)

The Domany-Kinzel (DK) model encompasses several types of nonequilibrium phase transitions, depending on the selected parameters. We apply supervised, semisupervised, and unsupervised learning methods to studying the phase transitions and critical behaviors of the $(1+1)$-dimensional DK model. The supervised and the semisupervised learning methods permit the estimations of the critical points, the spatial and temporal correlation exponents, concerning labeled and unlabeled DK configurations, respectively. Furthermore, we also predict the critical points by employing principal component analysis and autoencoder. The PCA and autoencoder can produce results in good agreement with simulated stationary particle number density.

DOI: 10.1103/PhysRevE.110.024102

## I. INTRODUCTION

Machine learning (ML) methods have attracted much attention in recent years and have been widely applied to many fields, such as natural language processing [1], face and image recognition [2,3], ecology [4], economics and finance [5], data mining and analysis [6], and electronic games [7], etc. Recently, reinforced learning methods even have solved the previously unfathomable Go games (AlphaGo) [8]. Machine learning methods also show their great advances in boosting multidisciplinary problem solving, especially when the related tasks are data-driven optimization problems.

In the physics realm, some major progresses have also been made with machine learning methods. For example, quantum computing has been combined with machine learning to develop the field of quantum machine learning: In Ref. [9] the quantum annealer has been used to sample the Boltzmann distribution, and Ref. [10] and Ref. [11] have studied the quantum Boltzmann machine and constructed quantum neural network, respectively. In high-energy physics, a machine learning classifier has been constructed to search for new particles of unknown masses [12], using parameterized networks to simplify the training process and enhance the learning performance. The quantum chromodynamics (QCD) phase transition has also been studied by using deep convolutional neural networks [13]. More recently, graph neural networks (GNNs) combined with a HaarPooling operation have been applied to extracting the features of quark-gluon tagging [14], which can enhance the accuracy of quark-gluon tagging, as compared to the weakly supervised learning method proposed earlier [15]. In astrophysics, the machine learning package ASTROML has been developed [16], and machine learning methods have also been utilized to boost cosmological and astrophysics process simulations [17].

Important breakthroughs have also been made in learning different phases of matter. The seminal work by Carrasquilla and Melko in 2016 have demonstrated that the ferromagnetic and paramagnetic phases of the classical Ising model can be classified based on supervised machine learning methods [18], permitting the identification of the critical points and the spatial correlation exponents. This work has since triggered great interest in the application of machine learning methods to the studies of various types of phase transitions. Regardless of the complexity of the target problem, the versatility of machine learning methods allows for learning more complex phases of three-dimensional Ising model [19], or phases with nonlocal and topological (Kosterlitz-Thouless) properties in percolation, XY and generalized XY models [20–22], or even phases of nonequilibrium matter [23], such as many-body localized and topological phases, and the nonequilibrium phase transitions in the directed percolation (DP) [24]. As in Ref. [18], it has been repeatedly demonstrated that one can estimate the critical points and the spatial correlation exponents, which further enhances the possibility for obtaining the entire phase diagram that is consistent with theory [23].

In this work, we study the phase transitions of the $(1+1)$-dimensional Domany-Kinzel (DK) model by machine learning techniques. As will be shown in the next section, the DK model is controlled by two parameters. Along the transition line, the model characterizes several types of phase transitions. Hence the DK model provides an excellent test bed for comparing the capabilities of different learning methods. To that end, supervised, semisupervised, and unsupervised learning methods will be applied to each type of phase transition. For supervised learning, the respective critical exponents are estimated after the phases are learned. We also propose a semisupervised learning method, in which only half probability data of training set with respect to test set are

*Contact author: liw@mail.ccnu.edu.cn

†Contact author: dengsf@snnu.edu.cn

 

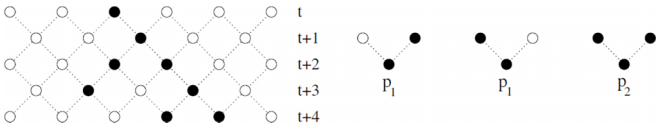


FIG. 1. The $(1+1)$-dimensional Domany-Kinzel model [26]. Occupied sites are marked by black circles. The state $s_{i,t+1}$ of a given site $i$ at time $t+1$ depends on the states of its left and right neighbors $(s_{i-1,t}, s_{i+1,t})$ at time step $t$.

labeled. The trained neural network can then predict the order parameter of the unlabeled DK model and the corresponding critical points. In addition, two unsupervised learning methods, principal component analysis (PCA) and autoencoder, are employed to estimate the critical points.

The remainder of this paper is organized as follows. In Sec. II, we briefly introduce the DK model. Section III presents the supervised learning of $(1+1)$-dimensional DK model, in which the critical points and the correlation length and correlation time exponents are estimated. Section IV gives the semisupervised learning results of $(1+1)$-dimensional DK model. Section V is about the unsupervised learning results of $(1+1)$-dimensional DK model, via autoencoder and PCA. Section VI summarizes the main findings of this work.

## II. DOMANY-KINZEL MODEL

The Domany-Kinzel (DK) [25,26] model is a stochastic cellular automaton that exhibits nonequilibrium active-to-absorbing type phase transition, controlled by two parameters. In $(1+1)$ dimensions, the model is defined on a one-dimensional array, on which site $s_i$ can be either occupied ($s_i = 1$) or empty ($s_i = 0$). As illustrated in Fig. 1, the state of each site is then updated with time with respect to its nearest neighbors according to the following rule:

$$s_i(t+1) = \begin{cases} 1 & \text{if } s_{i-1}(t) \neq s_{i+1}(t) \quad \text{and } r_i(t) < p_1 \\ 1 & \text{if } s_{i-1}(t) = s_{i+1}(t) = 1 \text{ and } r_i(t) < p_2 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $0 \leqslant r_i(t) \leqslant 1$ is a random number generated from a uniform distribution, and $0 \leqslant p_1 \leqslant 1$ and $0 \leqslant p_2 \leqslant 1$ are two probabilities used to control the phases of the model. In practice, simulations were performed on a triangular lattice displayed as a square lattice, with the periodic condition being imposed. The time of the lattice counts from $t = 1$ and at each even time step, the one-dimensional configuration is rotated to the right by one position to mimic the two-dimensional triangular space-time lattice structure. Suppose the size of the system is $L$. Due to this rotation, the periodic boundary condition is implemented by adding an extra site $s_{L+1}$. At each step, the DK rules are implemented by first determining the state $s_i(t+1)$ according to $s_i(t)$ and $s_{i+1}(t)$. Then when $t+1$ is odd, we identify the last site with the first site by setting $s_{L+1}(t+1) = s_1(t+1)$, and when $t+1$ is even, the entire row is rotated to the right by one position and the ensuing new first site is identified with the new last site by setting $s_1(t+1) = s_{L+1}(t+1)$.

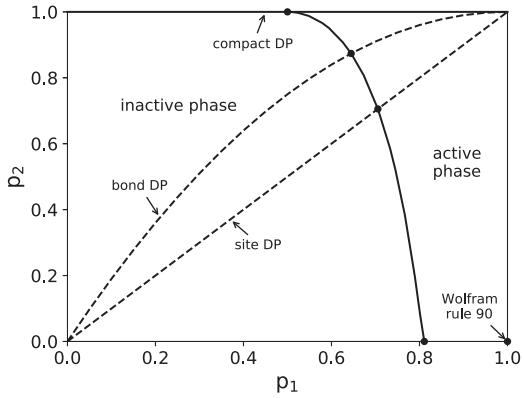


FIG. 2. Phase diagram of the $(1+1)$-dimensional Domany-Kinzel model, plotting data from Ref. [27]. Bond directed percolation corresponds to the line $p_2 = p_1(2-p_1)$. Site directed percolation is obtained for $p_1 = p_2$. For $p_1 = 1$ and $p_2 = 0$, it is equivalent to Wolfram rule 90. For simplicity, we refer to this rule throughout the paper even when $p_1 < 1$, as long as $p_2 = 0$. For $p_2 = 1$, it is a different universal scaling behavior called compact directed percolation.

From the above rule, one can easily imagine that unless all the sites are initially occupied, given the probability $p_2$, if $p_1$ is too small, the proportion of the occupied sites will decrease until only empty sites remain, whereas, for large enough $p_1$ value, the array will become increasingly occupied until a saturated density is reached. Once the system evolves into a fully empty state, there is no way for it to get out of that state. Hence the model displays an active-to-absorbing phase transition.

As depicted in Fig. 2 for the phase diagram, there is a transition line $(p_{1c}, p_{2c})$ separating the active phase and the absorbing one. Depending on the location of the parameters, the DK model includes bond and site DP as special cases. Bond DP corresponds to the line $p_2 = p_1(2-p_1)$, while site DP is obtained for $p_1 = p_2$. For $p_1 = 1$ and $p_2 = 0$, it is equivalent to Wolfram rule 90 [28–30] and for $p_2 = 1$, since an empty site is guaranteed to be filled as long as both its neighbors are occupied, the generated clusters become compact, giving rise to a different universal scaling behavior called the compact directed percolation (CDP). The CDP has two absorbing states, namely the empty and the fully occupied lattice. The CDP is different from the bond DP, the site DP, and the Wolfram rule 90, as the latter ones all belong to the DP universality [31,32]. This is exemplified by the critical clusters of the DK model generated from a single active seed shown in Fig. 3, in which the compact pattern of the CDP cluster is quite distinctive from those of the rest ones.

The particle number density of active sites of the DK is defined as

$$\rho(t) = \left\langle \frac{1}{L} \sum_i s_i(t) \right\rangle, \tag{2}$$

which depends on the initial condition and is averaged over all realizations of the dynamics. The order parameter of the DK model is then the long time stationary value $\rho_{\text{stat}}$ of the density $\rho(t)$ as $t \to \infty$, which is characterized by a power law in the

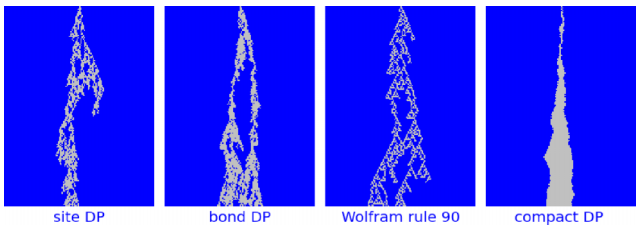


FIG. 3. Critical clusters of DK model generated from a single active seed.

vicinity of the critical point:

$$\rho_{\rm stat} \sim (p - p_c)^{\beta}. \tag{3}$$

## III. SUPERVISED LEARNING OF THE DOMANY-KINZEL MODEL

The inputs for the learning machines are just raw configurations generated from Monte Carlo (MC) simulations of the $(1+1)$-dimensional DK model. According to the phase diagram depicted in Fig. 2, each type of phase transition of the DK model can be controlled by varying the probability $p_1$. We henceforth denote it as $p$ for simplicity. The generated configurations are split into the training set and the test one. In the training set, each configuration is labeled according to the probability $p$ that generated that configuration. The labeling is a prerequisite for the supervised learning. If a configuration $\mathbf{x}_i$ was obtained by a simulation with $p < p_c$, it is in the absorbing phase and the output vector $\mathbf{y}_i$ assigned to it during training is (see below) $\mathbf{y}_i = (0, 1)$, whereas configurations $\mathbf{x}_i$ generated at $p > p_c$ are assigned $\mathbf{y}_i = (1, 0)$. Since knowledge of $p_c$ is required for training, supervised learning cannot be used to determine $p_c$. It can be used to predict the phase of a large number of simulated configurations on the basis of a much smaller amount of training data. In the subsequent sections, we should explore alternative ML methods to partially circumvent this issue.

For supervised learning, we apply the convolutional neural network (CNN) as illustrated in Fig. 4, on which, sigmoid activation function is used for the convolution and the pooling layers, and SOFTMAX activation function is taken in the output layer, producing a binary classification output. For a certain test configuration $\mathbf{x}_i(p)$ fed into the neural network, the two output units take real values between 0 and 1. The value taken by one output unit represents the probability $P_1(\mathbf{x}_i(p))$ that the configuration belongs to the active phase, and the value taken by the other output unit represents the probability $P_0(\mathbf{x}_i(p))$ that the configuration belongs to the absorbing phase.

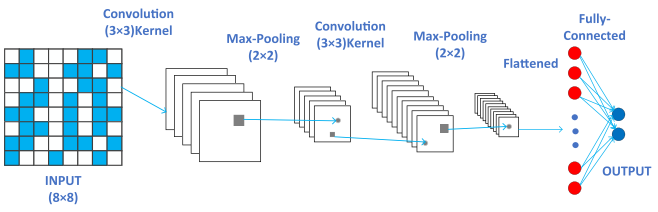


FIG. 4. Schematic structure of CNN.

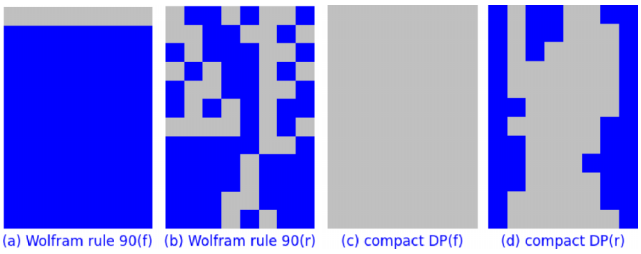


FIG. 5. (a) Cluster of Wolfram rule 90 generated from fully occupied active seeds. (b) Critical cluster of Wolfram rule 90 generated from randomly occupied active seeds. (c) Cluster of CDP generated from fully occupied active seeds. (d) Critical cluster of CDP generated from randomly occupied active seeds. Gray color represents occupied sites, and blue ones represent empty sites.

Since learning machines try to learn features of configuration images of relatively small system sizes and of different phases, it is customary to make use of configurations obtained from fully occupied or randomly occupied (e.g., with a probability of 0.5) initial states instead of starting from a single active seed as shown in Fig. 3, as the latter leaves a large proportion of the sites empty at initial stages. Here, randomly occupied initial states are more preferable because for the Wolfram rule 90 ($p = p_1,\ p_2 = 0$) and the CDP ($p = p_1,\ p_2 = 1$), fully occupied initial states only result in trivial absorbing states, as illustrated in Figs. 5(a) and 5(c), respectively, regardless of how the other parameter ($p_1$ for the Wolfram rule 90 and the CDP) is chosen.

Simulation times $t$ are typically selected with respect to the characteristic time $t_f$. On a finite lattice of nonequilibrium phase transition, there is always a nonvanishing probability of reaching the absorbing configuration, finite-size effects set in after a typical time $t_f$ that grows with the system size as $t_f \sim L^z$. For the DP universality class, $z = 1.58$ [33], while for the CDP, $z = 2$ [25].

The configuration images are of $L \times (t+1)$ dimension. From initial states with randomly occupied sites, for each probability $p$, 1700 labeled configurations are generated for the training set and another 500 configurations for the test set. The CNN output layers are eventually averaged over the test set, giving $P_{0|1}(p) = \frac{1}{500}\sum_{i=1}^{500} P_{0|1}(\mathbf{x}_i(p))$.

To examine the proper system size, we started with $L = 12$ and $L = 8$ for the Wolfram rule 90 and the CDP, which give rise to the relatively low accuracy values of 90.03% and 90.72%. Here, accuracy refers to the percentage of correct predictions of the phases (as compared to the true labels in the test set) of the test set with the trained CNN model. Figures 6(a) and 6(b) show the CNN output layers averaged learning results over a test set as a function of $p$ for the Wolfram rule 90 and the CDP. Therefore, larger sizes will be taken in what follows. Figures 6(c) and 6(d) show the averaged results for the output layers from the above-trained CNN with respect to bond and site DP, respectively. The critical points can be estimated from the crossing point of the two output layers, which is typically around $P_0(p_c) = P_1(p_c) \approx 0.5$. With $L = 32$, we hence estimate $p_c = 0.644 \pm 0.02$ for the bond DP. With $L = 40$, we estimate $p_c = 0.701 \pm 0.02$ for the site DP. These estimations yield accuracy of 99.66% for the

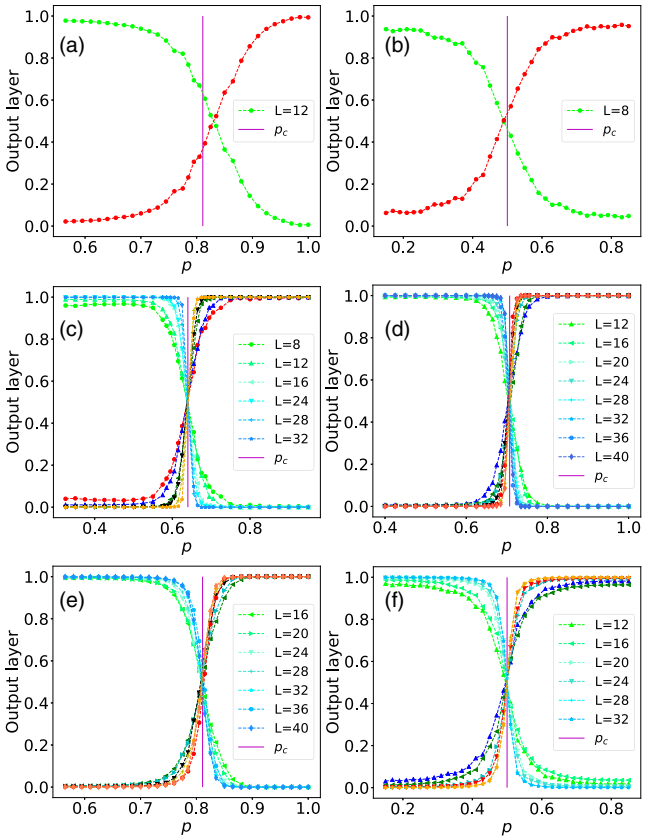


FIG. 6. The CNN output layers averaged learning results $[P_0(p)$ and $P_1(p)]$ over a test set for (a) the Wolfram rule 90 ($L = 12$, $t = 60$); (b) the CDP($L = 8$, $t = 64$); (c) the bond DP ($L = 8$, 12, 16, 20, 24, 28, 32; $t = 50$, 60, 80, 115, 152, 194, 240); (d) the site DP ($L = 12$, 16, 20, 24, 28, 32, 36, 40; $t = 60$, 80, 115, 152, 194, 240, 288, 340); (e) the Wolfram rule 90 ($L = 16$, 20, 24, 28, 32, 36, 40; $t = 80$, 115, 152, 194, 240, 288, 340); and (f) the CDP ($L = 12$, 16, 20, 24, 28, 32; $t = 145$, 256, 400, 576, 784, 1024). The values of $p_c$ used in the learning phase of each case are indicated by the vertical lines; they were taken from Ref. [33] for the bond and the site DP, from Ref. [27] for the Wolfram rule 90, and from Ref. [25] for the CDP.

bond DP critical point, and accuracy of 99.38% for the site DP; see the accuracy data in Table I. Therefore, even though the employed system sizes are relatively small, supervised learning by CNN still allows us to classify the two phases and estimate the associated critical points quite well.

For Figs. 6(e) and 6(f), with $L = 40$, we estimate $p_c = 0.798 \pm 0.02$ for the Wolfram rule 90. With $L = 32$, we estimate $p_c = 0.498 \pm 0.02$ for the CDP. These estimations yield

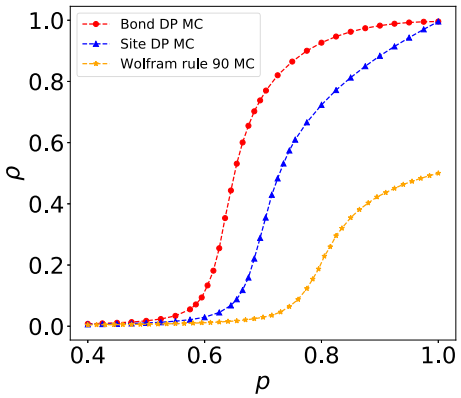


FIG. 7. With $L = 32$, $t = 240$, comparing the particle number density of the bond DP, site DP and Wolfram rule 90.

accuracy of 97.46% for the Wolfram rule 90 critical point, and accuracy of 98.41% for the CDP.

As one can notice from Table I, the accuracy values for the bond DP, the site DP, and the Wolfram rule 90 display a decreasing trend among them for each studied system size. As shown in Fig. 7, the particle number densities of the three models at the same $p$ are actually different and display the same trend. Since these three variants all belong to the DP universality class, here we observe that the nonuniversal lacunarity property (porous structure) of clusters, associated with the particle number density, still affects the learning accuracy, although one may just want to probe the same universality properties.

The features of nonequilibrium phase transitions such as absorbing phase transitions are encoded in the correlations within the spatial configurations and their dynamical evolution. Approaching the critical point, the spatial correlation length $\xi_\perp$ and the temporal correlation length $\xi_\parallel$ diverge as

$$\xi_\perp \sim |p - p_c|^{-\nu_\perp} \quad \text{and} \quad \xi_\parallel \sim |p - p_c|^{-\nu_\parallel}, \tag{4}$$

where $\nu_\perp$ and $\nu_\parallel$ are spatial and temporal correlation exponents, respectively, and $\xi_\parallel \sim \xi_\perp^z$, with $z = \nu_\parallel/\nu_\perp$ being the dynamical exponent. For finite systems simulated within finite times, by noting that finite-size effects set in when $\xi_\perp \sim |p - p_c|^{-\nu_\perp} \sim L$, and $\xi_\parallel \sim |p - p_c|^{-\nu_\parallel} \sim t$, one sees that $x = (p - p_c)L^{1/\nu_\perp}$ and $y = (p - p_c)t^{1/\nu_\parallel}$ are dimensionless quantities, so the functions $\hat{P}_{0|1}(x)$ and $\hat{P}_{0|1}(y)$ are scaling functions of $x$ and $y$, respectively, and we expect data collapse for different system sizes when the scaled variables $x$ and $y$ are used. Hence, it is possible to estimate $\nu_\perp$ and $\nu_\parallel$ by performing data collapse techniques.

TABLE I. The accuracy values of the trained CNNs with respect to different system sizes $L$ for (i) the bond DP, (ii) the site DP, (iii) the Wolfram rule 90, and (iv) the CDP.

| size $L$ | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
|---|---|---|---|---|---|---|---|---|---|
| bond DP | 93.77% | 95.61% | 97.92% | 98.46% | 98.53% | 99.45% | 99.66% | – | – |
| site DP | 90.85% | 95.03% | 96.4% | 97.75% | 98.43% | 98.57% | 99.1% | 99.18% | 99.38% |
| Wolfram rule 90 | 87.51% | 90.03% | 94.63% | 95.09% | 96.2% | 96.31% | 96.93% | 96.99% | 97.46% |
| CDP | 90.72% | 94.76% | 96.43% | 97.06% | 97.82% | 98.23% | 98.41% | – | – |

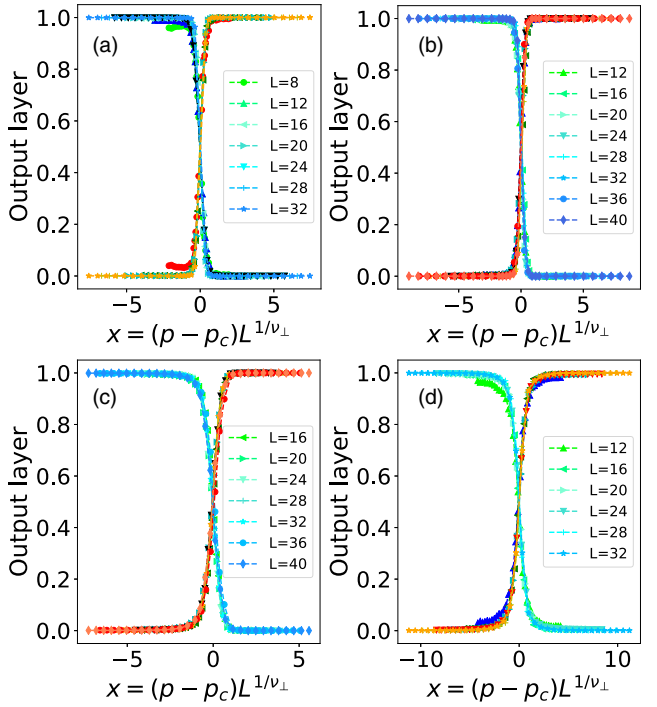


FIG. 8. CNN outputs results as a function of $x = (p - p_c)L^{1/\nu_\perp}$ for (a) the bond DP ($L = 8, 12, 16, 20, 24, 28, 32$; $t = 50, 60, 80, 115, 152, 194, 240$), (b) the site DP ($L = 12, 16, 20, 24, 28, 32, 36, 40$; $t = 60, 80, 115, 152, 194, 240, 288,340$), (c) the Wolfram rule 90 ($L = 16, 20, 24, 28, 32, 36, 40$; $t = 80, 115, 152, 194, 240, 288, 340$), (d) the CDP ($L = 12, 16, 20, 24, 28, 32$; $t = 145, 256, 400, 576, 784, 1024$).

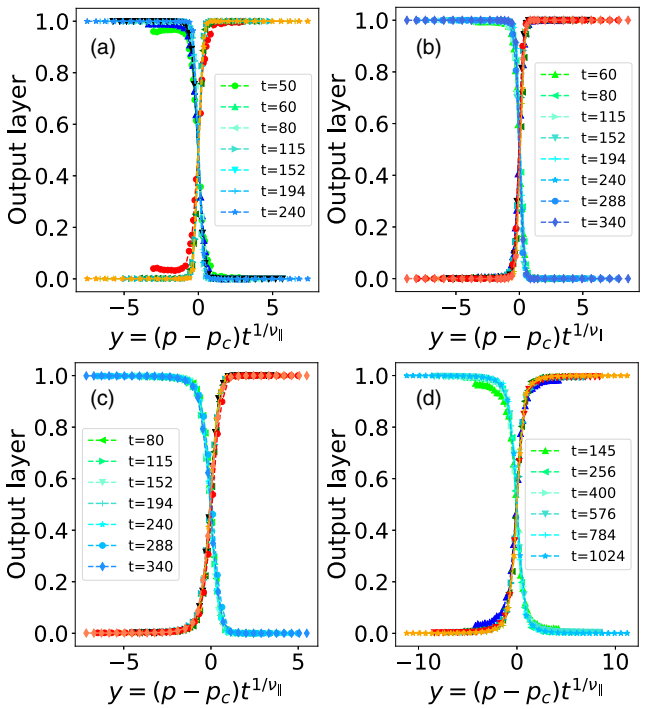


FIG. 9. CNN outputs results as a function of $y = (p - p_c)t^{1/\nu_\parallel}$ for (a) the bond DP ($L = 8, 12, 16, 20, 24, 28, 32$; $t = 50, 60, 80, 115, 152, 194, 240$), (b) the site DP ($L = 12, 16, 20, 24, 28, 32, 36, 40$; $t = 60, 80, 115, 152, 194, 240, 288, 340$), (c) the Wolfram rule 90 ($L = 16, 20, 24, 28, 32, 36, 40$; $t = 80, 115, 152, 194, 240, 288, 340$), (d) the CDP ($L = 12, 16, 20, 24, 28, 32$; $t = 145, 256, 400, 576, 784, 1024$).

Before we proceed, let us note that the bond DP, the site DP, and the Wolfram rule 90 of the $(1 + 1)$-dimensional DK all belong to the $(1 + 1)$-dimenional DP universality class, which is characterized by the correlation exponent $\nu_\perp \approx 1.0968(4)$ and the temporal correlation exponent $\nu_\parallel \approx 1.7338(6)$ [33]. The CDP represents a different universality class (the CDP universality class) of absorbing phase transitions, where the percolation clusters are compact objects, which is characterized by $\nu_\perp = 1$ and $\nu_\parallel = 2$ in $1 + 1$ dimensions [25,34].

Now, rescaling the probability $p - p_c$ by choosing proper $\nu_\perp$ and $\nu_\parallel$ in Fig. 6 should render the output layer curves for different sizes collapse to the scaling functions $\hat{P}_{0|1}(x)$ and $\hat{P}_{0|1}(y)$. As seen in Fig. 8, the curves coincide for different sizes with a suitable choice of $\nu_\perp$. The estimated DK critical exponents are $\nu_\perp = 1.092 \pm 0.02$ for the bond DP, $\nu_\perp = 1.087 \pm 0.03$ for the site DP, and $\nu_\perp = 1.073 \pm 0.03$ for the Wolfram rule 90, which are consistent with the theoretical value $\nu_\perp = 1.0968$. For the CDP, We estimate $\nu_\perp = 0.991 \pm 0.02$, which is again consistent with the theoretical value $\nu_\perp = 1$. Similarly, Fig. 9 shows the data-collapse results for temporal correlations with respect to different simulation times. The estimated DK critical exponents are $\nu_\parallel = 1.727 \pm 0.02$ for the bond DP, $\nu_\parallel = 1.725 \pm 0.03$ for the site DP, and $\nu_\parallel = 1.719 \pm 0.03$ for the Wolfram rule 90, which are consistent with the theoretical value $\nu_\parallel = 1.7338(6)$. We also estimate $\nu_\parallel = 1.967 \pm 0.03$ for the CDP, agreeing with the theoretical value $\nu_\parallel = 2$. According to $z = \nu_\parallel/\nu_\perp$, the estimated DK dynamical exponents are $z = 1.5815 \pm 0.02$ for the bond DP, $z = 1.587 \pm 0.03$ for the site DP, $z = 1.602 \pm 0.03$ for the Wolfram rule 90, $z = 1.985 \pm 0.03$ for the CDP.

Machine learning has been well applied to studying equilibrium phase transition models, but applying it to studying nonequilibrium phase transitions is a new research field, having attracted much attention in recent years. Previously, it has been demonstrated that spatial correlation exponents of nonequilibrium phase transitions can be extracted in a similar manner as in the nonequilibrium case [24]. Here, we explore further and find that the CNN output layer also contains temporal correlation information, which permits the extraction of temporal correlation exponents.

## IV. SEMISUPERVISED LEARNING OF THE DOMANY-KINZEL MODEL

Semisupervised learning is a kind of learning paradigm, which combines supervised learning with unsupervised learning or utilizes the capability of the trained model through supervised learning to predict values (labels) that do not occur in the training set. Hence, there are two slightly different semisupervised learning settings, namely, inductive and transductive semisupervised learning. Inductive learning assumes that the unlabeled samples in the training data are not the data to be predicted, in the hope that the trained model can be applied to data not observed during the training phase.

Transductive learning, on the other hand, assumes that the unlabeled samples considered in learning are exactly the data to be predicted—the unlabeled samples constitute the test set [35,36]. In what follows, we will take the more straightforward latter approach. In this way, one may substantially reduce the cost for data labeling, or, for more practical applications, one only needs to label data with the most certainty and leaves those less certain ones unlabeled and treat them as the test set.

In semisupervised learning, basically the same convolutional neural network as illustrated in Fig. 4 is being used, only that the output layer contains only one neuron, intending to predict the stationary particle number density $\rho_{\text{stat}}$ through its output value. Therefore, instead of labeling the raw configuration data according to the phase they are in, raw configurations are labeled by their stationary particle number densities $\rho_{\text{stat},i}(p)$ (for the $i$th sample with respect to probability $p$), inferred from MC configurations directly. To test the ability of semisupervised learning, the training set, labeled by $\rho_{\text{stat},i}(p)$, only uses a sparser probability selection [e.g. $p = (0.1, 0.3, 0.5, 0.7, 0.9)$], while the test set includes unlabeled data for more extensive $p$ values [e.g. $p = (0, 0.1, 0.2, 0.3, \ldots, 0.9, 1)$]. For the DK model, simulations are run on arrays of size $L = 16$, up to $t = 120$ steps. For each chosen probability $p$, 2000 labeled configurations are generated for the training set and another 1000 configurations for the test set. Once the CNN is fully trained with respect to the training set, the CNN output should predict the stationary particle number density $\rho_{\text{stat},i}(p)$ for the $i$th sample in the test set for probability $p$.

Figures 10(c)–10(f) show the results for the semisupervised learning on the test sets, indicating that semisupervised learning indeed can predict the stationary particle number density of the DK model, complying with the counterpart from the Monte Carlo simulations well. From the peak of the variance, the estimated values of critical points are $p_c = 0.637 \pm 0.02$ for bond DP, $p_c = 0.695 \pm 0.02$ for site DP, $p_c = 0.496 \pm 0.02$ for compact DP, and $p_c = 0.793 \pm 0.02$ for Wolfram rule 90.

We remark that even for the DK model, the selection of the labeled portion in the training set could be quite arbitrary. This opens a possible avenue for the study of more intricate phase transitions such as topological phase transitions. Previously, it had been demonstrated that the unsupervised learning methods (PCA) are not suitable for extracting the critical points of the XY model [37]. It would be interesting to study if one can infer the full phase information for topological phase transitions via semisupervised learning by utilizing only partial information of these transitions.

## V. UNSUPERVISED LEARNING OF THE DOMANY-KINZEL MODEL

Unsupervised learning is a type of machine learning that learns from unlabeled data. In certain scenario, labeled data is scarce or expensive to obtain and unsupervised learning allows machines to automatically discover hidden patterns or sample classification through data analysis without explicit guidance. In this section, two well-known unsupervised learning methods, i.e., autoencoder and principal component analysis (PCA), will be applied to detect the phases of the DK model. For both methods, the outputs will be restricted to two- and one-dimensional spaces, which are the most physically relevant, as will be demonstrated in this section.

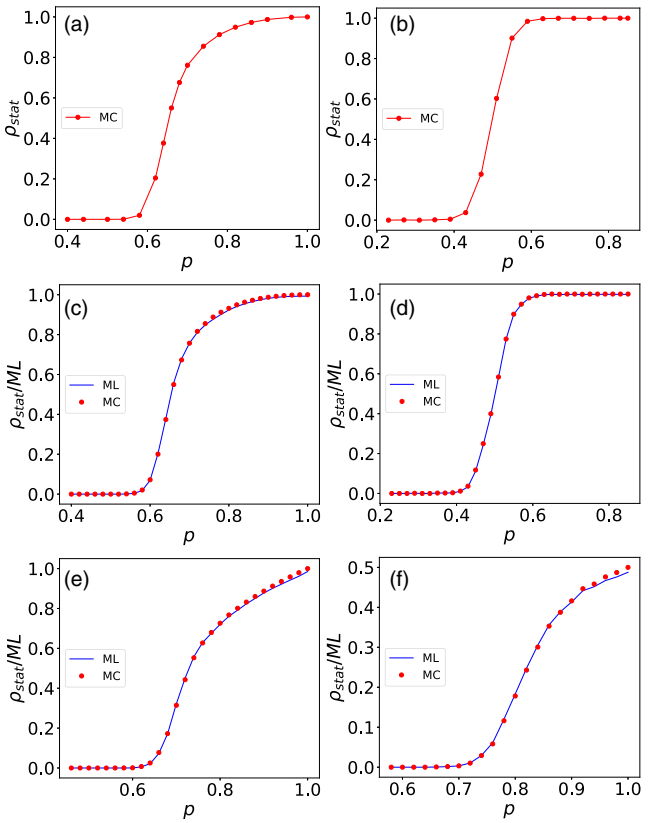


FIG. 10. Stationary particle number density $\rho_{\text{stat}}(p)$ for the labeled training sets of (a) the bond DP and (b) the CDP, as well as semisupervised learning results averaged over a test set as a function of $p$ for of the particle number density with partially labeled training sets for (c) the bond DP, (d) the CDP, (e) the site DP, and (f) the Wolfram rule 90. Red dots represent the stationary particle number density of the DK model and blue lines represent the CNN outputs. The predicted results were obtained after averaging over all the output results for the test set.

### A. Autoencoder results of the DK model

Autoencoders are simple generative models, which can produce random outputs that are similar to the inputs [38–40]. As illustrated in Fig. 11, the fully connected autoencoder architecture that we used includes an input layer, an encoder, a latent layer of hidden neurons, a decoder, and an output layer. The inputs for the autoencoders are just raw DK configurations $x_i$. The model is trained until the L2 loss function

$$L(\phi, \theta) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - D_\theta(E_\phi(x_i))||_2^2 \tag{5}$$

is minimized with respect to the encoder parameters $\phi$ and the decoder parameters $\theta$, with $E_\phi$ and $D_\theta$ being the encoder and the decoder functions, respectively. In this way, an effective representation, which preserves the most important informa-

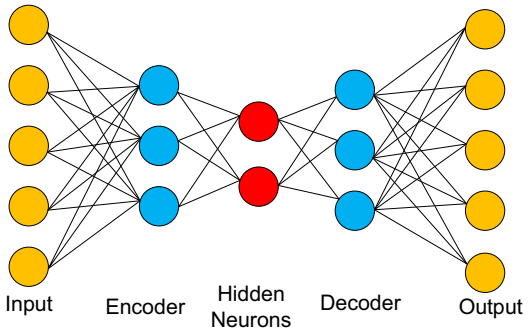


FIG. 11. Schematic structure of autoencoder. The fully connected autoencoder architecture includes an input layer, an encoder (four layers of neurons), a latent layer of hidden neurons (one or two neurons), a decoder (four layers of neurons), and an output layer, with number of neurons (640, 256, 128, 64, 16, 2|1, 16, 64, 128, 256, 640) and reLU activation functions for the layers.

tion of the input data, is achieved in the latent layer through data compression, which further permits data reconstruction via data decompression with the decoder.

First, we examine the autoencoder with two hidden neurons in the latent layer. For the DK model, simulations are run on arrays of size $L = 16$ and $t = 120$, with the configurations from the last 40 time steps ($\Delta t = 40$) being taken. For each probability $p$, 2000 configurations are generated for the training set, and another 1000 configurations for the test set. Since the autoencoder output is limited to two hidden neurons, the DK configurations are compressed into two dimensions. Once the autoencoder is trained, each input $\mathbf{x}_i$ from the test set then gives rise to one point $(h_{i1}, h_{i2})$ on a two-dimensional plane corresponding to the degree of freedom of the latent layer.

As shown in Fig. 12, the used autoencoder roughly classifies the DK configurations into two clusters, although the absorbing phase and the active phase are not completely separated. Especially, configurations drawn from the same probability $p$ are closely clustered together, so the fuzzy

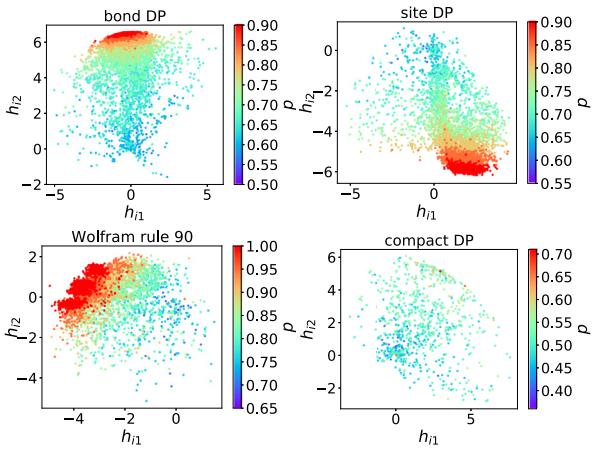


FIG. 12. The autoencoder output is linked with two hidden neurons, projecting the configurations of the bond DP, the site DP, the Wolfram rule 90, and the CDP onto two dimensions. The colormap represent the probability $p$.

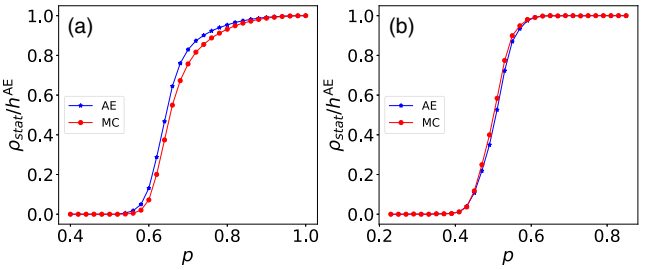


FIG. 13. Encoding the raw bond (a) DP and (b) CDP configurations to a single latent variable, through a single hidden neuron, as a function of the probability $p$. Here, adaptive moment estimation (ADM) is selected as the autoencoder optimizer. The blue and the red curves represent the autoencoder results and the MC results, respectively.

boundary of the two phases means that the transition is of continuous type.

Previously, it was found that in the autoencoder learning of the two-dimensional Ising model, the latent variable and the magnetization are highly correlated [37,41]. It was recently found that the latent variable and the particle density for the $(1 + 1)$ dimensions bond DP are also highly correlated [24]. Hence, analogously, we surmise that if the latent layer is restricted to just one hidden neuron, compressing all the configurations onto representations in one dimension, the learned feature should be the stationary particle density. As a first attempt, we selected adaptive moment estimation (ADM) as the autoencoder optimizer. It can be found from Fig. 13 that the normalized output $h^{\mathrm{AE}}$ of the hidden neuron and the stationary particle number density do not strictly coincide with one another, which is beyond our expectation.

Alternatively, we now choose stochastic gradient descent (SGD) as the autoencoder optimizer. The curve of normalized output $h^{\mathrm{AE}}$ versus the probability $p$ is now comparable to the counterpart of stationary particle number density for the DK model, as illustrated in Fig. 14. This indicates that the optimizer is very crucial for autoencoder.

Based on the data presented in Fig. 14, from the peak of the variance, the estimated values of critical points are $p_c = 0.638 \pm 0.02$ for the bond DP, $p_c = 0.696 \pm 0.02$ for the site DP, $p_c = 0.495 \pm 0.02$ for the compact DP, and $p_c = 0.793 \pm 0.02$ for the Wolfram rule 90. The Euclidean distance, defined as $\sqrt{\sum_{i=1}^{n} (h_i^{\mathrm{AE}} - \rho_i^{\mathrm{MC}})^2}$, can be utilized to quantify the difference between the (normalized) latent variable $h^{\mathrm{AE}}(p)$ and the stationary particle number density $\rho_{\mathrm{stat}}(p)$ from the Monte Carlo simulations. The Euclidean distances for the DK model are found to be 0.01445 for the bond DP, 0.01525 for the site DP, 0.01805 for the Wolfram rule 90, and 0.008198 for the compact DP. These values are quite small, suggesting that the single latent variable $h^{\mathrm{AE}}$ of the autoencoder learning can be treated as the stationary particle number density, as we conjectured earlier.

Hence, while two neurons in the latent layer are capable of clustering the DK configurations into two phases, the single latent variable of autoencoder results in good agreement with the DK stationary particle number density. Both these two constructions suggest that autoencoders can capture essential

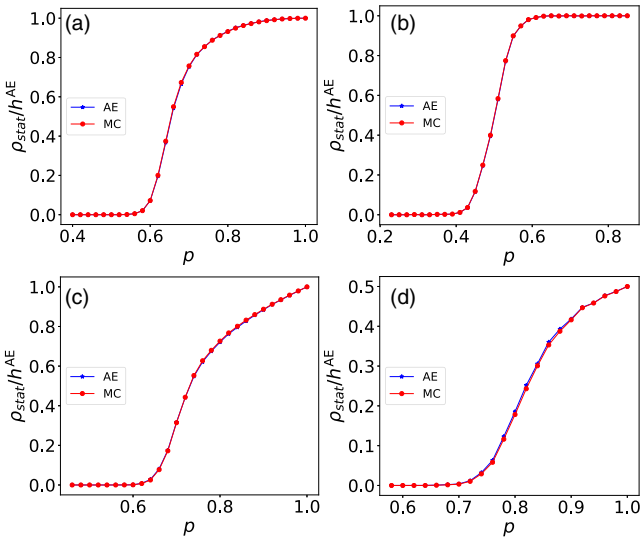


FIG. 14. Learning of the DK model when SGD is chosen as the autoencoder optimizer. The raw DK configurations are encoded on to a latent variable of a single hidden neuron as function of the probability $p$ for (a) the bond DP, (b) the CDP, (c) the site DP, (d) the Wolfram rule 90. The single-variable outputs (blue) of DK autoencoder results are normalized, which are compared to the stationary particle number densities (red) from the Monte Carlo simulations.

information of the input data so as to detect the phases or the critical points, without any prior knowledge of the DK model.

### B. PCA results of the DK model

Principal component analysis (PCA) is also an unsupervised learning algorithm, which can be used for data dimensionality reduction [42–44]. PCA performs orthogonal transformation on the data to find the direction of high variance, and converts the variables with possible correlations into linearly uncorrelated ones. The transformed variables are called principal components and here only the first two leading components will be used for the analysis of the DK model. One can intuitively imagine the process as projecting the data points of the original high-dimensional representation onto a lower-dimensional space by selecting proper directions of projection with largest variances, so that the maximum amount of information is still preserved after reduction of the dimensions. Similar to the previous section, we will also be interested in reductions into two- and one-dimensional spaces.

To be more specific, the PCA method is implemented as follows. Simulations are run on arrays of size $L = 16$ and $t = 120$, with the configurations from the last 40 time steps ($\Delta t = 40$) being taken, giving rise to $M = (L \times \Delta t)$-dimensional vectors $\mathbf{x}_i$'s. For each probability $p$, 1000 configurations are generated. Let $N_p$ be the number of $p$ values being studied, the full data set contains a total of $N = 1000 \times N_p$ samples so that an $(N \times M)$-dimensional matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N)^T$ is used to obtain the PCA results. PCA determines the principal components through a linear transformation of the original data $\mathbf{Y} = \mathbf{XW}$, where the

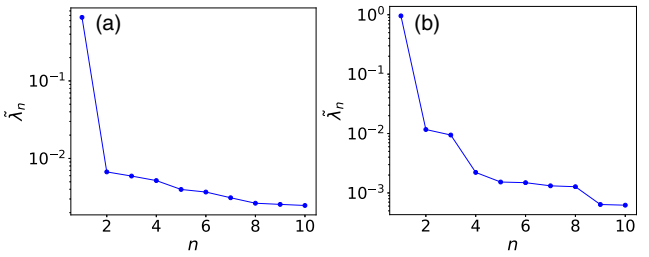


FIG. 15. PCA explained variance ratios of the first 10 principal components for (a) the site DP and (b) the CDP.

$(M \times K)$-dimensional transformation $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K)$ consists of $K$ weighted components (column vectors) $\mathbf{w}_n$ of dimension $M$. If $K \ll M$, a dimensionality reduction can be achieved.

Now by focusing on the real symmetric $(M \times M)$-dimensional covariance matrix $\mathbf{X}^T\mathbf{X}$ and considering $K = M$, the PCA component directions corresponding to descending variance spans can be determined by choosing $\mathbf{w}_n$'s as eigenvectors of $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X}\mathbf{w}_n = \lambda_n \mathbf{w}_n, \tag{6}$$

with eigenvalues sorted in a descending order $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \lambda_M \geqslant 0$, representing variances of the input matrix $\mathbf{X}$ along the directions of the corresponding $\mathbf{w}_n$ vectors. Dimensionality reduction is then achieved by selecting the first few $\mathbf{w}_n$'s corresponding to the largest eigenvalues. In the terminology of PCA, we denote the normalized eigenvalues $\tilde{\lambda}_n = \lambda_n / \sum_{i=1}^{M} \lambda_i$ as the explained variance ratio, which amounts to the weight accounting for the total variance along the direction of the $n$th component. As illustrated in Figs. 15(a) and 15(b), weights of the first few components already account for almost the entire variance.

The two leading components of DK configurations by PCA are illustrated in Fig. 16. Similar to the autoencoder results, this reveals that PCA can also roughly classify the DK

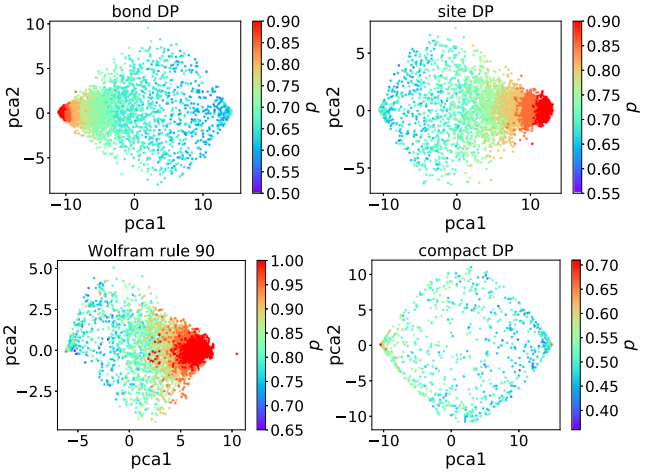


FIG. 16. PCA results for the bond DP, the site DP, the Wolfram rule 90, and the CDP, with projection of the raw configurations onto the plane of the two leading principal components. The colormap represent the probability $p$.

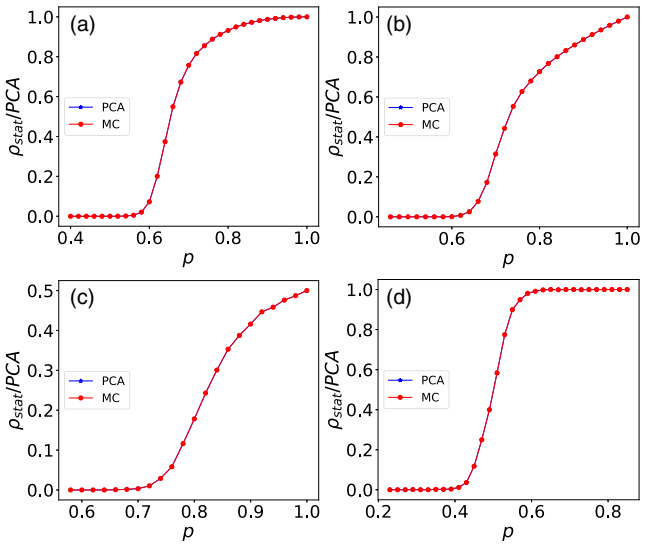


FIG. 17. PCA results for (a) the bond DP, (b) the site DP, (c) the Wolfram rule 90, (d) the CDP. The normalized first leading component (blue) varies as a function of the probability $p$, which is compared to the stationary particle number density (red) from the Monte Carlo simulations; both averaged over 1000 samples for each $p$ value.

configurations into two phases. Since the boundary of these phases is not distinct, the transition is of continuous nature, as expected. It is interesting to note that almost all data points for high $p$ values are projected to very few red dots for the CDP. This can be understood as a consequence of projecting very similar compact objects to the PCA plane given that the last 40 rows of the CDP configurations ($t = 81$-$120$) were taken and the clusters are mostly active for high $p$ values.

It is also of interest to examine the normalized first leading component of the DK configurations and compare it to the particle number density from the Monte Carlo simulations. As presented in Fig. 17, the two are close to each other for any given probability $p$ and the Euclidean distances from the MC results are 0.00012 for the bond DP, 0.000094 for the site DP, 0.00022 for the Wolfram rule 90, and 0.00011 for the compact DP. These distance values are quite small, suggesting that the first leading component can be interpreted as the stationary particle number density. This interpretation of the first leading component is then quite in resemblance to the magnetization analogy of the first leading component of the PCA in the two-dimensional Ising model [37,45]. It is found that the first leading component and the particle density for the $(1+1)$-dimensions bond DP are also highly correlated [46]. From the peak of the variance, the estimated values of critical points are $p_c = 0.639 \pm 0.02$ for the bond DP, $p_c = 0.698 \pm 0.02$ for the site DP, $p_c = 0.496 \pm 0.02$ for the compact DP, and $p_c = 0.795 \pm 0.02$ for the Wolfram rule 90. The estimated values are in good agreement with both those obtained from autoencoder in the previous section and the MC results [33].

Although PCA is just based on linear transformations of the input data, we show that it is still effective in extracting features of the DK phase transitions. Since the PCA could achieve similar results as compared to autoencoder without needing to train a model firstly, it costs less and is more convenient to apply.

## VI. SUMMARY

In this paper, we applied supervised, semisupervised, and unsupervised learning methods to study the phase transitions and critical behavior of the Domany-Kinzel model. With supervised learning, the critical points were estimated from the neural network outputs. By further collapsing the outputs for different sizes, the correlation exponents $\nu_\perp$ and $\nu_\parallel$ were estimated, which are consistent with reference values in the literature. Previously, it has been demonstrated that, similar to the equilibrium case, the spatial correlation exponent $\nu_\perp$ of nonequilibrium phase transitions can be extracted. Here, we explore further and find that the CNN output layer also contains temporal correlation information, which permits the extraction of the temporal correlation exponent $\nu_\parallel$. The achieved high accuracy values even for rather small system sizes suggest that the applied learning machine could learn the features of the phases quite well, so that the computation overheads in the MC simulation end can be substantially reduced.

The unsupervised learning methods, PCA and autoencoder, are able to roughly separate the phases into two clusters if the output is two dimensional. Once the output is restricted to one dimension, both methods yield an output that is in good agreement with the stationary particle number density and the estimated critical points are also in good agreement with the literature values within error margins. Since learning through PCA is simpler, PCA generally is more efficient as compared to autoencoder.

In semisupervised learning, even though only half of the training set was labeled, by setting the output to just one neuron, we found that the network predicts the stationary particle density of the test set quite well, permitting us to estimate the critical points of the DK model as well. Given these features of semisupervised and unsupervised learning methods, it is advisable to use these methods to study more intricate phase transitions if data labeling becomes costly.

Finally, we remark that even though only universal property of the DK model should matter along the DP critical transition line, the nonuniversal lacunarity property (porous structure) of clusters affects the learning accuracy. It is observed that learning machines generally learn the phase features of the DK models with denser clusters (e.g., the bond DP) better than models with sparser clusters (e.g., the Wolfram rule 90).

## ACKNOWLEDGMENTS

We thank Jianmin Shen and Longfeng Zhao for valuable discussions. Wolfram rule 18 should be replaced with the correct Wolfram rule 90, we thank an anonymous referee for pointing this out. This work was partially supported by the Fundamental Research Funds for the Central Universities, China (Grant No. CCNU19QN029), the National Natural Science Foundation of China (Grants No. 11505071, No. 61702207 and No. 61873104), and the 111 Project 2.0, with Grant No. BP0820038.

[1] A. Graves, A.-r. Mohamed, and G. Hinton, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, New York, 2013), pp. 6645–6649.

[2] O. Parkhi, A. Vedaldi, and A. Zisserman, *Deep Face Recognition* (British Machine Vision Association, Durham, 2015), pp. 1–12.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems*, Vol. 25, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Curran Associates, Inc., New York, 2012).

[4] S. Christin, É. Hervet, and N. Lecomte, Methods Ecol. Evol. **10**, 1632 (2019).

[5] P. Gogas and T. Papadimitriou, Comput. Econ. **57**, 1 (2021).

[6] Z. Ge, Z. Song, S. X. Ding, and B. Huang, IEEE Access **5**, 20590 (2017).

[7] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, IEEE Transactions on Games **12**, 1 (2019).

[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, Nature (London) **529**, 484 (2016).

[9] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Phys. Rev. A **94**, 022308 (2016).

[10] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Phys. Rev. X **8**, 021050 (2018).

[11] E. Farhi and H. Neven, arXiv:1802.06002.

[12] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson, Eur. Phys. J. C **76**, 235 (2016).

[13] L.-G. Pang, K. Zhou, N. Su, H. Petersen, H. Stöcker, and X.-N. Wang, Nat. Commun. **9**, 210 (2018).

[14] F. Ma, F. Liu, and W. Li, Phys. Rev. D **108**, 072007 (2023).

[15] L. M. Dery, B. Nachman, F. Rubbo, and A. Schwartzman, J. High Energy Phys. **05** (2017) 145.

[16] J. VanderPlas, A. J. Connolly, Ž. Ivezić, and A. Gray, in *2012 Conference on Intelligent Data Understanding* (IEEE, New York, 2012), pp. 47–54.

[17] F. Villaescusa-Navarro, D. Anglés-Alcázar, S. Genel, D. N. Spergel, R. S. Somerville, R. Dave, A. Pillepich, L. Hernquist, D. Nelson, P. Torrey *et al.*, Astrophys. J. **915**, 71 (2021).

[18] J. Carrasquilla and R. G. Melko, Nature Phys. **13**, 431 (2017).

[19] R. Zhang, B. Wei, D. Zhang, J.-J. Zhu, and K. Chang, Phys. Rev. B **99**, 094427 (2019).

[20] W. Zhang, J. Liu, and T.-C. Wei, Phys. Rev. E **99**, 032142 (2019).

[21] M. J. S. Beach, A. Golubeva, and R. G. Melko, Phys. Rev. B **97**, 045207 (2018).

[22] J. F. Rodriguez-Nieva and M. S. Scheurer, Nature Phys. **15**, 790 (2019).

[23] J. Venderley, V. Khemani, and E.-A. Kim, Phys. Rev. Lett. **120**, 257204 (2018).

[24] J. Shen, W. Li, S. Deng, and T. Zhang, Phys. Rev. E **103**, 052140 (2021).

[25] E. Domany and W. Kinzel, Phys. Rev. Lett. **53**, 311 (1984).

[26] H. Hinrichsen, Physica A **369**, 1 (2006).

[27] S. Lübeck, J. Stat. Mech.: Theory Exp. (2006) P09009.

[28] S. Wolfram, Rev. Mod. Phys. **55**, 601 (1983).

[29] G. Zebende and T. Penna, J. Stat. Phys. **74**, 1273 (1994).

[30] Note that in the literature, e.g., Refs. [27,47,48], had attributed this special case to Wolfram rule 18. The correct rule should be Wolfram rule 90 instead, as the state of the site $i$ is irrelevant according to the definition (1). For simplicity, the stochastic version ($p_1 < 1$) is also referred to as Wolfram rule 90 throughout the paper. In the special case in which the system evolves from a single seed, rules 18 and 90 happen to generate exactly the same Sierpiński pattern [28].

[31] H.-K. Janssen, Z. Phys. B **42**, 151 (1981).

[32] P. Grassberger, Z. Phys. B **47**, 365 (1982).

[33] I. Jensen, J. Phys. A: Math. Gen. **32**, 5233 (1999).

[34] J. Essam, J. Phys. A: Math. Gen. **22**, 4927 (1989).

[35] X. Zhu and A. B. Goldberg, *Introduction to Semisupervised Learning* (Springer Nature, Berlin, 2022).

[36] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach *et al.* (Curran Associates, Inc., New York, 2019).

[37] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).

[38] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, Neurocomputing **139**, 84 (2014).

[39] A. Ng *et al.*, CS294A Lecture notes **72**, 1 (2011).

[40] D. Bank, N. Koenigstein, and R. Giryes, arXiv:2003.05991.

[41] C. Alexandrou, A. Athenodorou, C. Chrysostomou, and S. Paul, Eur. Phys. J. B **93**, 226 (2020).

[42] H. Abdi and L. J. Williams, WIREs Computational Stats **2**, 433 (2010).

[43] M. Ringnér, Nature Biotechnol. **26**, 303 (2008).

[44] J. Shlens, arXiv:1404.1100.

[45] L. Wang, Phys. Rev. B **94**, 195105 (2016).

[46] J. Shen, S. Wang, W. Li, D. Xu, Y. Yang, Y. Wang, F. Gao, Y. Zhu, and K. Tuo, arXiv:2311.11741.

[47] S. Lübeck, Int. J. Mod. Phys. B **18**, 3977 (2004).

[48] H. Hinrichsen, Adv. Phys. **49**, 815 (2000).