



Determining global property of dusty plasma from single particle dynamics using machine learning

Chen Liang , Dong Huang, Shaoyu Lu, and Yan Feng 

Institute of Plasma Physics and Technology, Jiangsu Key Laboratory of Thin Films, School of Physical Science and Technology, Soochow University, Suzhou 215006, China



(Received 19 May 2023; accepted 21 July 2023; published 7 August 2023)

By machine learning, specifically the convolutional neural network (CNN) technology, the screening parameter κ is determined from single particle dynamics in two-dimensional (2D) dusty plasma simulations and also in an experiment. Independent Langevin simulation runs are performed with different κ values to obtain individual particle dynamical data to train, validate, and test CNNs. By adjusting the training dataset to mimic the application with the experimental data, three different CNNs are designed. From the analyzed results with the test dataset of simulations, all three CNNs have excellent performance in determining the κ value, with the resulting root-mean-square error of the determined κ value of 0.081, 0.279, and 0.155, respectively. While using these trained CNNs with the 2D dusty plasma experimental data, the distribution of the determined κ values has a prominent peak at the κ value, agreeing well with that determined from the phonon spectra. The results presented here clearly demonstrate the feasibility of determining the global property of dusty plasmas purely from single particle dynamics using machine learning, which may be developed further to diagnose more complicated plasma environment conditions in the future.

DOI: [10.1103/PhysRevResearch.5.033086](https://doi.org/10.1103/PhysRevResearch.5.033086)

I. INTRODUCTION

For any physical system, its global property is definitely a key problem that needs to be quantified or investigated. Statistical mechanics aims to explain the global physical properties of matter in bulk, based on the dynamics of its microscopic constituents [1]. In principle, statistical mechanics is applicable to different phases of matter, which has achieved considerable success [1–8]. In statistical mechanics, starting from the position and velocity information of the abundant particles, their detailed dynamics data can be easily averaged, leading to the global property that is able to describe the collective behaviors of the studied system [3]. In some experiments, due to the hardware limitation, there are probably not enough data of particle dynamics captured. In fact, the position and velocity information of particles cannot be accurately and simultaneously obtained experimentally [9,10]. Here, we attempt to determine the global property of an experimental system from individual particle dynamics, or even without the velocity information.

Laboratory dusty plasma, also termed complex plasma, typically refers to partially ionized gas containing micron-size solid dust particles [11–23]. Under the typical laboratory conditions, these dust particles are charged by free electrons

and ions in the plasma, reaching a high negative charge of $\approx 10^4 e$ in the steady state within microseconds [24]. These highly charged dust particles can be suspended and confined by the electric field in the sheath, forming a single layer, i.e., the so-called two-dimensional (2D) dusty plasma [25–32]. The interaction between these dust particles can be accurately described using the Yukawa repulsion [33]. Due to their high charges, these dust particles are strongly coupled [34]; as a result, a collection of them typically exhibits properties or behaviors of both solids and liquids [35–44]. Since each individual dust particle can be directly imaged and tracked experimentally, 2D dusty plasma is an excellent model system to study various physical properties and dynamics in liquids and solids at the kinetic level, such as viscosity [26], viscoelasticity [28], phase transitions [27], and waves [45,46].

Machine learning, leading the recent rapid development of computer technology, becomes a powerful tool for massive data analysis that attracts tremendous attention in various fields [47]. In plasma physics studies, machine learning has been applied to many studies [48–53]. For example, in [49], based on machine learning technology, a robust disruption warning algorithm is developed to predict the disrupted discharges of fusion plasmas in tokamak with overall disruption prediction accuracy of $\geq 90\%$. In [51], machine learning is used to reveal the accuracy boundary between higher-fidelity models and a simple, lower-fidelity model. In [52], machine learning is used to build a turbulence transport prediction model for fusion plasmas, which is able to perform rapid experimental analysis. In [53], using machine learning, an architecture for a tokamak magnetic controller is developed, which is able to autonomously learn to command the full set of control coils.

*Author to whom correspondence should be addressed: fengyan@suda.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Recently, in dusty plasma related investigations, machine learning has also been widely exploited in various investigations [54–63]. In 3D dusty plasma investigations, machine learning is used to distinguish fcc, hcp, and bcc phases [54], obtain the phase transition diagram [55], study the electrorheological transition from the stringlike to the isotropic system [56], as well as identify the interface of binary complex plasma [57]. In [58], machine learning is used to track the microparticle clouds in exploding wires, dusty plasmas, and atmospheric plasmas, so that the data analysis results are improved. In [59], the machine learning algorithm is adopted to reconstruct the three-dimensional positions of particles in a dense cloud of particles in plasma with the successfully reconstructed particle fraction of $\geq 80\%$. In [60], machine learning is also used to enhance the diagnosis of dusty plasma. In [61], the dust ablation trails in tokamaks are automatically identified and analyzed by the developed machine learning code package, with an accuracy of $\geq 99\%$. In [62], the solid-liquid phase transition diagram for 2D dusty plasma is obtained using machine learning, agreeing well with [13]. While using machine learning to analyze the 2D dusty plasma experiment [63], the environmental forces acting on dust particles are extracted from the noisy data with an accuracy of 50% better than conventional methods.

As one of the two key parameters to characterize the global property of dusty plasmas [17], the screening parameter κ is defined as the ratio of the length scale between neighboring dust particles to the environment Debye length λ_D . In 2D dusty plasma experiments, the determination of the κ value is not so straightforward due to the complicated plasma diagnostics and the λ_D variation in different levitation heights in the plasma sheath. One widely used method is based on the stochastic thermal motion of dust particles in the initial highly ordered lattice [27–29], from which the wave spectra are calculated, so that the corresponding longitudinal and transverse dispersion relations are obtained. Then, one may compare these experimentally obtained dispersion relations to the theoretical dispersion relationships of a 2D Yukawa lattice [64,65], where there are two variables of κ and the dusty plasma frequency ω_{pd} [27–29]. By characterizing the discrepancy between the experimental and theoretical dispersion relations while varying both κ and ω_{pd} , a χ^2 surface on the κ and ω_{pd} plane is obtained, so that the κ value and its uncertainty can be determined from the confidence interval of χ^2 , similarly to [66]. Here, we attempt to determine the κ value differently, from the individual particle dynamics data using machine learning, specifically the convolutional neural network (CNN) technology [67]. Either in 2D dusty plasma experiments or simulations, the particle dynamics data are all discrete, so that the time interval between neighboring frames is crucial in the data analysis. In 2D dusty plasma experiments, the time interval between frames is always controlled by cameras, typically in units of milliseconds. While using machine learning, this time interval needs to be connected to dusty plasma dynamics, i.e., typically normalized using ω_{pd} . However, ω_{pd} has not been determined yet, which also varies in different experimental conditions. Thus, this issue needs to be solved in the machine learning data analysis, as presented in detail later.

The rest of this paper is organized as follows. In Sec. II, we briefly introduce our simulation method to mimic 2D dusty

plasmas, and the 2D dusty plasma experiment which will be analyzed below. In Sec. III, we describe the structure and training procedure of our designed CNNs. In Sec. IV, we report the analyzed κ results of the simulation data using our designed CNNs from the single particle dynamics and compare their accuracy. We also use these CNNs to determine the κ value in the 2D dusty plasma experiment, agreeing well with the obtained κ value from a completely different approach. In Sec. V, we provide our interpretation of the performance and generalization ability of our designed CNNs, as well as the future possibility to improve its performance. Finally, a summary is given in Sec. VI.

II. SIMULATION AND EXPERIMENT

To characterize our studied 2D dusty plasmas, the two key parameters of the screening parameter $\kappa = a/\lambda_D$ and the coupling parameter $\Gamma = Q^2/4\pi\epsilon_0ak_B T$ [17] are both used. Here, a is the Wigner-Seitz radius of $(n\pi)^{-1/2}$ [17] as a function of the number density n . The timescale is normalized using the 2D nominal dusty plasma frequency $\omega_{pd} = \sqrt{Q^2/(2\pi\epsilon_0ma^3)}$ [17].

A. Simulation method

To mimic 2D dusty plasmas, we perform Langevin dynamic (LD) simulations of 2D Yukawa systems using LAMMPS. In our simulations, the equation of motion for each dust particle i is

$$m\ddot{\mathbf{r}} = -\nabla \sum_{j \neq i}^N \phi_{ij} - \nu m \mathbf{v} + \xi_r. \quad (1)$$

Here, ϕ_{ij} is the interparticle interaction of the Yukawa repulsion [33] $\phi_{ij} = Q^2 \exp(-r_{ij}/\lambda_D)/4\pi\epsilon_0 r_{ij}$, where Q is the particle charge, r_{ij} is the distance between the particles i and j , λ_D is the Debye length, and ϵ_0 is the vacuum electric permittivity. On the right-hand side of Eq. (1), the second term $-\nu m \mathbf{v}$ is the frictional gas damping, expressed using the Epstein drag model with the coefficient of ν [68]. The last term ξ_r corresponds to the Langevin random kicks, which is chosen to attempt to achieve the desired temperature according to the fluctuation-dissipation theorem [69]. Note that for the training, validation, and test of our CNN, we perform two sets of LD simulations, as described later.

Here are some details of our simulations. For all of our simulation runs, the coupling parameter and the frictional gas damping coefficient are both specified to be constant as $\Gamma = 800$ and $\nu = 0.027\omega_{pd}$, respectively, while the screening parameter κ varies from 0.25 to 2.00, all comparable to those in typical 2D solid dusty plasma experiments [26–29]. The integration time step is chosen to be $\delta t \omega_{pd} = 0.001$, as well justified in [70]. To mimic the typical operation of 2D dusty plasma experiments [26–29], we output the simulated particle positions every 1600 integration steps, i.e., $\omega_{pd} dt = 1.6$, for each simulation run with the total time duration of $t \omega_{pd} = 819.2$.

The first set of our simulations, for the training of our CNN, contains eight independent simulation runs with the κ values changing from 0.25 to 2.00 with the constant step of 0.25. The

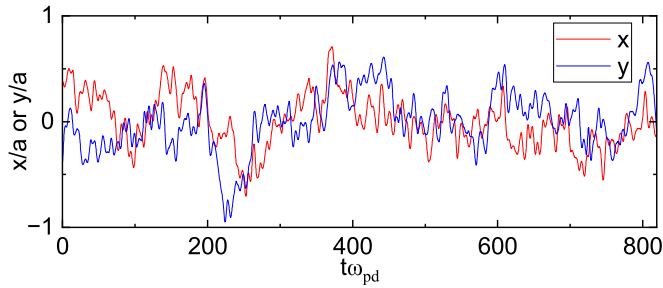


FIG. 1. Typical fluctuations of one particle position during the stochastic thermal motion of simulated 2D dusty plasma under $\kappa = 2$ and $\Gamma = 800$ from the training dataset. The global property of the whole system is determined by the collection of thousands of particles, so that, in principle, the motion of each individual particle should reflect the global property. The main topic of this paper is to determine the global property of the whole system based on the information directly extracted from the particle position fluctuations using machine learning. Note that for the position fluctuation data of one individual particle, the frame number is 512 and the time interval between neighboring frames is $\omega_{pd}dt = 1.6$.

2D simulation box has the dimensions of $243.8a \times 211.1a$, containing simulated 2^{14} particles, with the periodic boundary conditions. The obtained positions of these simulated particles are all packed as the training dataset of our CNN, containing 8×2^{14} particle position data, as one typical example shown in Fig. 1.

The second set of our simulations, for the validation and test of our CNN, contains 15 independent simulation runs with 15 specified κ values of 0.25, 0.40, 0.50, 0.60, 0.75, 0.90, 1.00, 1.10, 1.25, 1.40, 1.50, 1.60, 1.75, 1.90, and 2.00, respectively. The simulation box here has the dimensions of $121.9a \times 105.6a$, i.e., only 1/4 of the first set, containing 2^{12} particles, with the same particle number density and the same periodic boundary conditions. We randomly select 2^9 particles from each simulation run and then pack their position fluctuation data as the validation dataset. Next, we randomly choose the position fluctuation data of other 2^9 particles from each simulation run and pack them as the test dataset.

B. Experimental data

To test the actual performance of our CNN, we also use the particle position data of 2058 consecutive frames images from the 2D dusty plasma experiment of [29]. These images are captured by a top-view camera, with the resolution of 800×600 pixels, operated at the frame rate of 55 frames per second (FPS), with the field of view of $(29.05 \times 21.78) \text{ mm}^2$, containing ≈ 4400 dust particles. These imaged dust particles just vibrate briefly in the highly ordered lattice without any noticeable disturbances, as shown in Fig. 2(a). The Wigner-Seitz radius [17] of this lattice is obtained directly from these images as $a = 0.21 \text{ mm}$. Note that we use the improved moment method [9] to determine particle positions from the experimental images with the typical uncertainty of $< 1/4$ pixel, just corresponding to only about 2% of the lattice constant $b = 0.40 \text{ mm}$.

As shown in Fig. 2, we plot the superposition of the particle positions from the 2D dusty plasma experiment in $t = 2 \text{ s}$ and

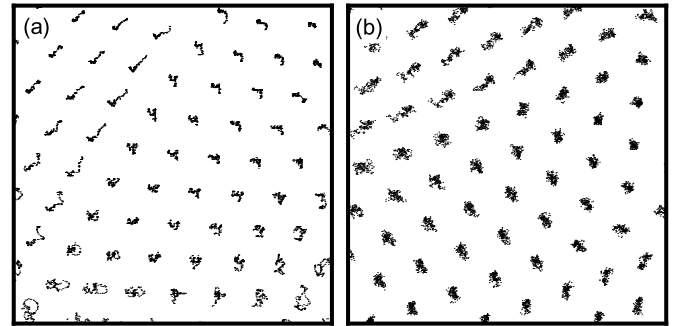


FIG. 2. Superposition of positions of ≈ 60 particles from (a) the 2D dusty plasma experiment of [29] for $t = 2 \text{ s}$ and (b) our performed LD simulation for $t\omega_{pd} = 172$. As described in [29], from the obtained wave spectra of the solid lattice, the values of κ and ω_{pd} in this 2D dusty plasma experiment are determined simultaneously, which are $\kappa = 0.47$ and $\omega_{pd} = 86 \text{ s}^{-1}$, respectively. In (b), the simulation conditions are specified as $\kappa = 0.50$ and $\Gamma = 800$, respectively. The particle position fluctuations in (a) and (b) are quite similar, whose characteristics may be identified using the machine learning studied here. Note that here only about 1.4% of the total field of view in the experiment is presented in (a), while only about 1.5% of the total simulation box is presented in (b).

our simulation within $t\omega_{pd} = 172$, both containing ≈ 60 particles. Clearly, the particle position fluctuations in Figs. 2(a) and 2(b) are quite similar, whose characteristics probably can be identified by machine learning, as we study here. Note that for the simulation data in Fig. 2(b), the conditions are specified as $\kappa = 0.50$ and $\Gamma = 800$.

To determine the values of ω_{pd} and κ , in [29], the wave spectra of the thermal motion of this initial highly ordered lattice are calculated to obtain the longitudinal and transverse dispersion relations. Next, these experimentally obtained dispersion relations are compared to the theoretical dispersion relationships [64,65] with varying values of ω_{pd} and κ . Then, the most matching pair of ω_{pd} and κ , with the least discrepancy between the experimental and theoretical dispersion relations, is chosen as the determined values of ω_{pd} and κ . As described in [29], the determined values from these wave spectra are $\kappa = 0.47$ and $\omega_{pd} = 86 \text{ s}^{-1}$, respectively. Note that the 55 FPS is enough to capture the detailed dynamics for the studied 2D dusty plasma with $\omega_{pd} = 86 \text{ s}^{-1}$; however, both too low and too high frame rates may lead to larger systematic error or noise in the particle tracking [10], which is not helpful in the CNN performance. In this paper, we will determine the value of the screening parameter κ using machine learning, a completely different approach, from the same experiment data.

We make use of the experimental data by extracting them into 72 section movies. We extract the original particle position data every $t = v_f^{-1} = 0.37 \text{ s}$, with the constant time duration of 10 s for each, containing 550 frames. Our current determination of the κ value using the CNN relies on the position fluctuations of these dust particles. However, in the experiment, the time series of these position data for one particle generally contain overall drifts, probably due to the rotation of the total 2D dust suspension. These overall drifts greatly suppress the particle position fluctuations, greatly

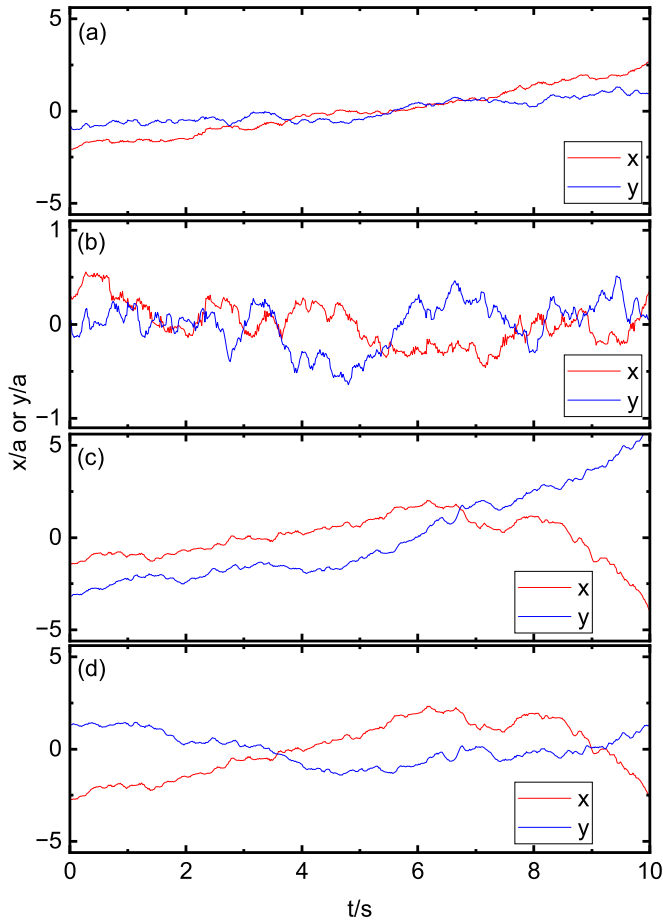


FIG. 3. Typical time series of one particle position in (a),(c) the 2D solid dusty plasma experiment and (b),(d) the corresponding data after attempting to remove their overall drifts. As shown in (a) and (c), the motion of one particle typically contains overall drifts. Here, we attempt to use the average velocity \mathbf{v}_d of each particle to remove the overall drift using $\mathbf{r}' = \mathbf{r} - \mathbf{v}_d t$, where \mathbf{r} is the location of each particle at any time t . For a particle with the fixed drift velocity, as in (a), the removing result, as in (b), meets our expectation that the time series of one particle's position is quite similar to the position fluctuations in Fig. 1. However, for particles with the varying drift velocity, as in (c), the removing result exhibits nonphysical features, as in (d). Thus, we calculate the standard deviation of the time series for the particle positions after removing the overall drift and only keep those with the standard deviation of $\leq a/4$. Finally, we obtain the position fluctuation data of 5998 particles during 10 s in the experiment of [29].

affecting the accuracy of our CNN. Thus, we need to remove the overall drift motion from these time series of particle positions efficiently, as explained next.

To remove the drift motion of each particle in each section movie, we subtract its drift from its locations \mathbf{r} in all times using $\mathbf{r}' = \mathbf{r} - \mathbf{v}_d t$. Here, the drift velocity \mathbf{v}_d is directly determined by the total displacement of this particle in this section movie divided by the total time duration of 10 s. As shown in Fig. 3(a), the time series of one particle position generally exhibits the monotonic linear variation trend, corresponding to the overall drift with a nearly constant velocity. Then, after removing the drift motion, the obtained

particle position fluctuations in Fig. 3(b) are quite similar to the fluctuations in the training data of Fig. 1, which definitely include the global property information for our investigations here. However, not all particles only drift at a constant velocity in all section movies as in Fig. 3(a). For example, in Fig. 3(c), the magnitude and direction of the drift velocity both change and, as a result, our simple removing method above causes the nonphysical feature of \mathbf{r}' in Fig. 3(d). Theoretically, for the highly ordered lattice without any disturbances, the distribution of the particle position fluctuations is pretty narrow, i.e., much smaller than the Wigner-Seitz radius a , as shown in Fig. 1. Thus, for our analysis with CNN, we only keep the experimental data for particles with the standard deviation of the particle fluctuation of $\leq a/4$, as in Fig. 3(b). Other experimental data, such as Fig. 3(d), are all excluded in the latter data analysis. Using this criteria, finally we obtain the position fluctuation data of 5998 particles in total from the 72 section movies extracted from the experiment of [29]. Note that the position fluctuations of each particle contain 550 frames, which are reduced to 512 using the downsampling method [71,72] with the constant new time interval in later CNN analysis.

III. STRUCTURE AND TRAINING PROCEDURE OF CNN

We follow the typical machine learning method [67,73] to design two models of our CNN, as shown in Fig. 4, which both consist of four steps. Model A is designed to analyze time series of particle positions, i.e., the particle position data of consecutive frames with the known or assumed time interval. In step 1 of model A, starting from a two-channel sequence data (2×512) of the particles' x and y positions, we apply four consecutive sets of 1D convolutional layer and one average pooling layer to generate the output of 256 feature vectors (256×2). Here, the kernel sizes of the convolution layers and average pooling layers are 1×5 and 1×4 , respectively, while the activation function is the ReLU function. As described in [67], in step 1 of model A, according to our specified parameters, each 1D convolutional layer increases the channel number of the input data by either four or two times, and then each subsequent average pooling layer reduces the data amount of each channel to one-quarter of its size, leading to the size of the final output data of (256×2) from the initial input data of (2×512). Note that the average pooling layer used here can be regarded as a kind of low-pass filter, which is able to effectively remove the high-frequency noise in the input data. In step 2, to convert the obtained (256×2) feature vectors in the previous step to one feature vector (1×512) here, we apply a flatten layer [74]. Meanwhile, to prevent overfitting, a dropout layer [75] is added here. In step 3, a linear layer with the ReLU activation function is used to further extract features from the feature vector (1×512), leading to a new feature vector (1×128). In step 4, we apply a linear layer without an activate function, leading to the final output (1×1) of model A, which just corresponds to the final determined value, i.e., the expected screening parameter κ_{NN} in our current investigation.

Model B is designed to analyze independent or uncorrelated particle position data, which are not necessary to be a time series. Thus, the fluctuation of particle position is not

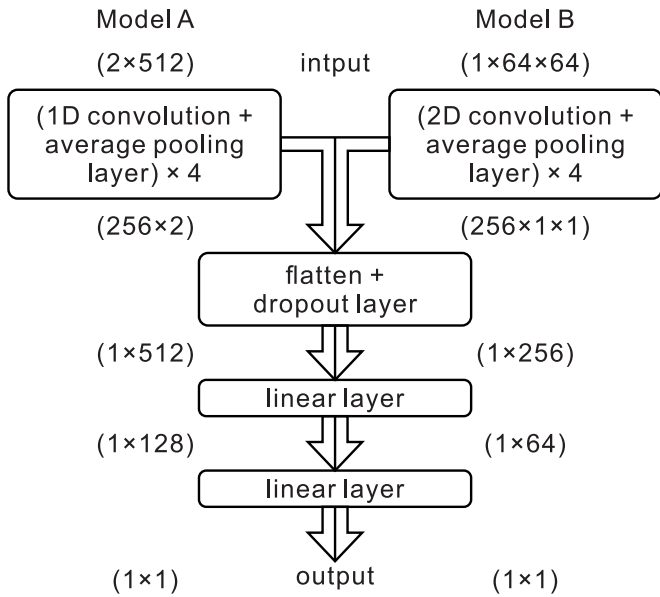


FIG. 4. Four steps of the structure for our convolutional neural network (CNN). In model A, first, four consecutive sets of one 1D convolutional layer combined with one average pooling layer are used with the input of two-channel sequence data (2×512), leading to the output of 256 feature vectors (256×2). Second, a flatten layer is used to convert these feature vectors into a new feature vector (1×512), and a dropout layer is added to prevent the overfitting. Third, we apply a linear layer with the rectified linear unit (ReLU) activate function to obtain a new feature vector (1×128). Finally, a linear layer without an activate function is applied, leading to the final output (1×1) of model A, which just corresponds to the determined screening parameter κ_{NN} . In model B, the only difference in the structure is in the first step, where the input data are single-channel images ($1 \times 64 \times 64$), so that the four 1D layers of model A are replaced by four 2D convolutional layers here, so that 256 feature maps ($256 \times 1 \times 1$) are obtained. The remaining three steps are the same for both models, while the parameters are different as shown.

longer used; instead, we completely rely on the static distribution of particle positions in each snapshot. The only difference between models A and B is in step 1, as shown in Fig. 4. In step 1 of model B, we apply four 2D convolutional layers due to the input of single-channel image data ($1 \times 64 \times 64$), as described in Sec. IV B in detail. Then, after the same three steps as in model A, we obtain the final output (1×1) of model B, which also corresponds to the determined screening parameter κ_{NN} .

We follow [67,73] to train our CNN using the first set of our simulations. First, we randomly divide the training dataset to several batches, each containing the position data of 128 different particles. Second, we input the data of one batch to our CNN and use the Smooth L1 loss function [74] to quantify the difference, which is named as the loss between the output κ_{NN} from our CNN and the specified screening parameter value κ_{MD} in our simulations. Third, based on the loss of this batch, the Adam optimizer [76] with the default parameters is used to update all learnable parameters in our CNN. Fourth, we repeat the above two steps until all batches are used. Fifth, we repeat all the above steps 50 times for the specified epoch number.

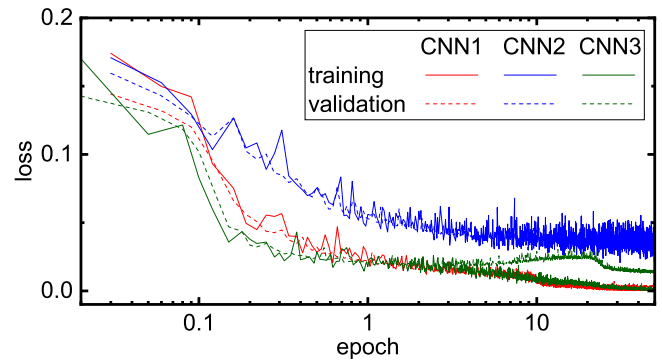


FIG. 5. Obtained loss results during the training procedure of our CNNs. As the epoch number increases, the training and validation loss results of each CNN gradually decay, nearly to zero, finally, indicating that our CNNs converge well without the overfitting feature, so that they all have sufficient generalization ability.

Note that whenever we input one batch into the model, we train the model once. To improve the training efficiency, after training the model 32 times using 32 batches, we perform the validation test using the validation dataset once, where the loss of this model is calculated. Thus, our training program outputs the losses of both the corresponding batch and the validation dataset every 32 times of training, as presented next. While using all the batches divided from the entire training dataset, we just finish performing one complete iteration through the training dataset during the training process of the model, which is called one epoch. However, the entire training dataset is not needed to be completely used at the same time. For example, if only 10% of the entire training dataset is used to train the model, then one may use 0.1 epoch to describe the current training progress, as shown in Fig. 5.

IV. RESULTS

In this paper, we design three different CNNs, labeled as CNN1, CNN2, and CNN3, respectively, and train them using the same method as described in Sec. III. After training them, we plot their training and validation losses, respectively, in Fig. 5. Figure 5 clearly indicates that they all converge well without the overfitting feature, so that they all have sufficient generalization ability. Let us explain the difference between these three CNNs in detail next.

A. Validation by CNN1

To validate the feasibility of the machine learning method in determining the global property of the whole system completely based on the fluctuations of individual particle positions, we use model A with the simulation data first. We follow the training procedure described in Sec. III to train model A using the first set of our simulations, i.e., the training dataset. When the training is finished, we label the trained model A as CNN1.

The determined κ values, κ_{NN} from our CNN1, agree well with the specified κ_{MD} values in the test dataset of our simulations, as shown in Fig. 6. We use CNN1 to analyze the position fluctuation data for each particle in the test dataset described

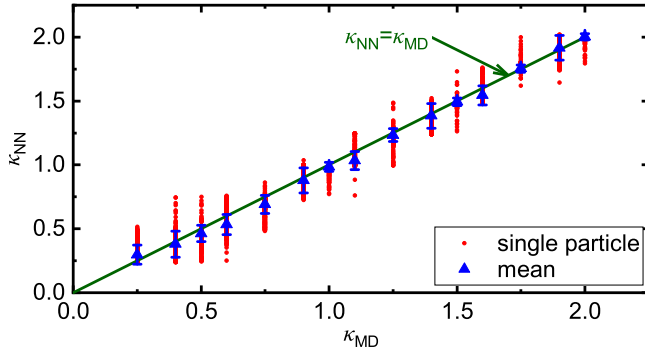


FIG. 6. Our determined κ_{NN} results using the trained CNN1 with the test simulation dataset. For each run of the test simulation dataset, our CNN1 determines the κ_{NN} value for each particle. The mean of all determined κ_{NN} values is almost the same as the specified screening parameter κ_{MD} value in the simulation, with the root-mean-square error (RMSE) of only 0.081.

in Sec. II A. The determined screening parameter κ_{NN} value from each particle is plotted as one dot in Fig. 6. Thus, for each value of κ_{MD} , there are 512 dots of the determined κ_{NN} value in Fig. 6. Then, for each κ_{MD} , we calculate the mean and standard deviation of the determined κ_{NN} , respectively, as marked in Fig. 6, also presented in Table I. From the straight line in Fig. 6, clearly, for each specified value of κ_{MD} , the mean of the determined κ_{NN} is almost the same as κ_{MD} , with a small standard deviation. In short, using CNN1, we validate that the fluctuations of individual particle positions can be used to accurately determine the global property of the κ value for the whole system.

TABLE I. The analysis results of the test simulation dataset from our three CNNs. First, for each value of κ_{MD} , we calculate the mean and standard deviation σ of the determined κ_{NN} values from our CNNs, respectively. Then, for each value of κ_{MD} , we perform a Gaussian fitting on the distribution of these determined κ_{NN} values for CNN2 and CNN3, respectively, resulting in the corresponding mean and standard deviation σ presented here. Finally, to characterize the accuracy of the determined κ_{NN} values from our CNNs, we calculate the RMSE for all determined κ_{NN} values of each CNN, as listed here. The obtained RMSE values clearly indicate that within these three CNNs, the performance of CNN2 is good and CNN3 is better, while CNN1 is the best.

κ_{MD}	CNN1		CNN2				CNN3			
	mean	σ	mean	σ	Gaussian's mean	Gaussian's σ	mean	σ	Gaussian's mean	Gaussian's σ
0.25	0.30	0.08	0.52	0.13	0.48	0.09	0.44	0.15	0.43	0.17
0.40	0.39	0.09	0.70	0.27	0.60	0.16	0.58	0.13	0.58	0.11
0.50	0.47	0.08	0.74	0.26	0.65	0.20	0.57	0.14	0.57	0.14
0.60	0.54	0.08	0.65	0.23	0.55	0.14	0.63	0.11	0.63	0.12
0.75	0.70	0.07	0.84	0.25	0.77	0.20	0.71	0.15	0.70	0.15
0.90	0.87	0.10	0.92	0.28	0.88	0.31	0.85	0.13	0.85	0.11
1.00	0.98	0.03	0.98	0.26	0.91	0.22	0.92	0.18	0.92	0.19
1.10	1.02	0.06	1.09	0.29	1.04	0.31	1.07	0.16	1.09	0.17
1.25	1.23	0.05	1.18	0.28	1.17	0.32	1.18	0.15	1.25	0.01
1.40	1.37	0.10	1.22	0.30	1.22	0.35	1.37	0.12	1.27	0.03
1.50	1.49	0.02	1.42	0.26	1.45	0.27	1.48	0.11	1.50	0.03
1.60	1.54	0.07	1.60	0.22	1.63	0.22	1.57	0.11	1.50	0.02
1.75	1.75	0.02	1.65	0.22	1.70	0.20	1.74	0.10	1.75	0.01
1.90	1.94	0.09	1.82	0.15	1.90	0.07	1.87	0.11	1.75	0.01
2.00	2.00	0.03	1.87	0.12	1.92	0.05	1.93	0.10	2.02	0.01
RMSE	0.081		0.279				0.155			

To characterize the accuracy of the obtained κ_{NN} values from our CNN, we calculate the root-mean-square error (RMSE) [77]. As a measure of the deviation of the obtained data to the true value, the RMSE is calculated using $\sqrt{\frac{1}{N} \sum_{i=1}^N (\kappa_{NN} - \kappa_{MD})^2}$, where N is the total number of the obtained κ_{NN} values, i.e., the total number of the analyzed particles. For our obtained κ_{NN} values from the CNN1 in Fig. 6, the calculated RMSE value is only 0.081 when κ_{MD} varies from 0.25 and 2.00. This low value of RMSE undoubtedly indicates the pretty satisfactory performance of our CNN1, further demonstrating the feasibility of our machine learning method in determining the global property of the κ value purely from the information of individual particle position fluctuations.

However, our CNN1 with the satisfactory performance above cannot be directly used in the experimental data analysis. This is because the κ_{NN} determination using CNN1 above is based on the test of the simulation data with the time interval between two consecutive frames of $1.6 \omega_{pd}^{-1}$, which is exactly the same as that of the training simulation data, i.e., the training and test datasets have the same time duration between neighboring frames. For dusty plasma experiments, the time interval between frames can typically be easily controlled in milliseconds, i.e., in the real time unit, while the corresponding value in units of ω_{pd}^{-1} is really unknown since ω_{pd} is still not determined. In fact, the ω_{pd} value varies for different experiment conditions. To solve this problem, we propose two more CNNs, so that the κ value in experimental data can also be determined, as described next.

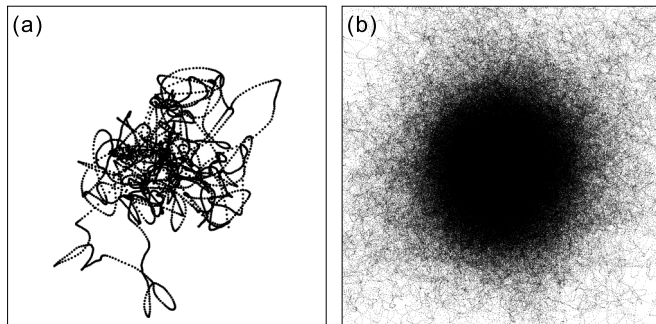


FIG. 7. Spatial distributions of position fluctuations of (a) a single particle and (b) 256 particles for the time duration of $t\omega_{pd} = 819.2$, from our 2D dusty plasma simulation under $\kappa = 2$ and $\Gamma = 800$. Clearly, while the sampling is large enough as in (b), the spatial distribution of particle positions tends to be close to Gaussian. However, if the sampling is not enough, as in (a), the distribution scatters more randomly.

B. Determination by CNN2

Since in experiments the time interval between neighboring frames in units of ω_{pd}^{-1} is unknown, we may completely ignore the time information in determining the κ_{NN} value only from the spatial distribution of individual particle positions, as one example shown in Fig. 7. As shown in Fig. 7(b), with enough sampling data, the spatial distribution of individual particle positions tends to be close to Gaussian. Of course, for one single particle position in a limited time duration, the corresponding distribution tends to scatter more randomly, as in Fig. 7(a), from which we expect to use the machine learning to extract some useful information. Here, following the same training process as in Sec. III, we train model B using the spatial distribution of all individual particle positions in the training dataset, comparable to the information in Fig. 7(b). The final trained model B is labeled as CNN2. Next, we use CNN2 to determine the κ_{NN} value from the distribution of position fluctuations for each individual particle, such as in Fig. 7(a).

To characterize the distribution of particle position fluctuations, we convert the spatial distribution to a 64×64 matrix. The positions of each particle are converted from the time series of fluctuations into a spatial distribution, as shown in Fig. 7(a). For one particle, we create a square of $-a \leq x(y) \leq a$, and divide it into 64×64 subregions with the size of $a/32 \times a/32$ each. Then, we count the number of times this particle appears in each subregion for its position fluctuations. Finally, we plot the obtained counts using the Max-Min normalization, leading to the position spatial distribution data of each particle. When these matrices are obtained, the latter data analysis can be easily performed.

As shown in Fig. 8(a), the determined values of κ_{NN} from CNN2 are generally consistent with the specified values of κ_{MD} . As compared with that of CNN1 in Fig. 6, the distribution of the determined κ_{NN} values from CNN2 in Fig. 8(a) is much wider because the time-related information is completely removed. For each specified κ_{MD} value, we calculate the mean of the obtained κ_{NN} values and also fit them to Gaussian expression, with the results presented in Table I; both roughly agree with the specified κ_{MD} value in simula-

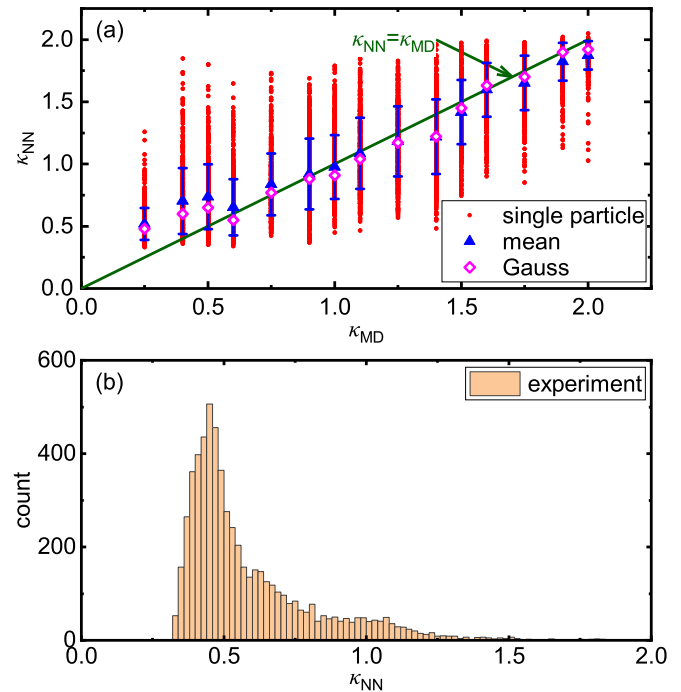


FIG. 8. Our determined κ_{NN} results using the trained CNN2 with (a) the test simulation dataset and (b) the experiment data. In (a), although the RMSE of about 0.279 is significantly larger than that of CNN1, the mean of κ_{NN} and the average from the Gaussian fitting of the determined κ_{NN} values from our trained CNN2 are still both approximately the same as with the specified κ_{MD} values in the test simulation dataset. In (b), while using our CNN2 with the experiment data, there is a prominent peak at ≈ 0.455 in the distribution of the determined κ_{NN} values, agreeing well with $\kappa = 0.47$ obtained from the phonon spectra in [29].

tions. This agreement indicates that CNN2 is able to extract the spatial distribution characteristics of particle position fluctuations under different screening parameters, even without the time information, seemingly beyond the scope of our current physics understanding. The RMSE of CNN2 is about 0.279, about 3.5 times that of the CNN1, which is reasonable since the time information is completely lost.

Our CNN2 is able to be directly used to analyze experiment data, as demonstrated in Fig. 8(b). As explained in Sec. II B, we have the position fluctuations of 5998 particles, extracted from the 2D solid dusty plasma experiment [29]. Next, our CNN2 is able to determine one κ_{NN} from the fluctuations of each particle. Then, we plot the distribution of these obtained 5998 values of κ_{NN} in Fig. 8(b), which has a prominent peak at ≈ 0.455 . In [29], using the widely accepted phonon spectra analysis method, the κ value was found to be 0.47. The location of the peak of our CNN2 determined κ_{NN} values agrees well with the κ value determined from a completely different approach, clearly indicating that even without the time information, the global property of the system is still able to be accurately determined by our CNN2.

C. Determination by CNN3

Although the exact ω_{pd} value of one 2D dusty plasma suspension is unidentified, the typical varying range of ω_{pd} is

often known, especially for the experiments performed on the same setup, which may be included in machine learning to reduce the final uncertainty. For the 2D dusty plasma experiment [29] analyzed here, we know that the typical range of ω_{pd} is $70 \leq \omega_{pd} \leq 90 \text{ s}^{-1}$ from the previous experiments [27–29]. If the time duration of the analyzed experiment movie is 10 s, then the corresponding duration of the simulation run should be $700 \leq t\omega_{pd} \leq 900$. To mimic the experiment analysis, let us include this information of the adjustable time duration into our simulation data analysis next.

To obtain new datasets containing the particle position fluctuations with the adjustable time duration of $700 \leq t\omega_{pd} \leq 900$, we perform two new sets of simulation runs, for the training and test (also including validation) purposes, respectively. As in Sec. II A, the first set of the training data contains eight independent simulation runs, while the second set of the validation and test data contains another 15 independent simulation runs. The detailed simulation settings are the same as in Sec. II A, with only two different points. First, for each simulation run here, the time duration is always specified to $900 \omega_{pd}^{-1}$, where we output the simulated particle positions every 1200 integration steps, i.e., $\omega_{pd} dt = 1.2$. Second, the simulation box size for the eight independent simulation runs of the training set is $121.9 a \times 105.6 a$, only 1/4 of that used in Sec. II A. For each simulation run of the training dataset, we extract the particle position fluctuations for the different values of the time duration 700, 750, 800, 850, and $900 \omega_{pd}^{-1}$, corresponding to 583, 625, 667, 708, and 750 frames, respectively. To mimic the experiment operation with the unknown time duration, for each simulation run of the validation and test dataset, we extract the particle position fluctuations using four arbitrary values of the time duration between 700 and $900 \omega_{pd}^{-1}$. Then, as in Sec. II A, we pack the position fluctuation data of all particles in five different values of the time duration from the first set of simulations as the new training dataset. Meanwhile, we randomly choose 2^9 particles from each simulation run of the second set of simulations, and pack their position fluctuation data with the four arbitrary values of the time duration as the validation and test datasets. Note that no matter how long the time duration is, the frame number of each particle is always reduced to 512 using the downsampling method [71,72] with the constant new time interval for the later CNN operations. We train model A using this new training dataset following the same procedure as in Sec. III, resulting in a new trained model A, named as CNN3.

From Fig. 9(a), the determined κ_{NN} values from CNN3 agree with the specified κ_{MD} values, better than the results of CNN2 in Fig. 8(a). For each specified κ_{MD} , the calculated mean of κ_{NN} and the average from the Gaussian fitting are both approximately the same as the κ_{MD} value, as shown in Table I. As compared with Fig. 8(a), the distribution of the determined screening parameter values κ_{NN} is significantly more concentrated. We also calculate the RMSE of the determined κ_{NN} results from CNN3, which is about 0.155, much smaller than that of CNN2. These results clearly indicate that the performance of CNN3 is better than that of CNN2.

We also use CNN3 to analyze the same experimental data used in Sec. IV B to determine the κ_{NN} value, as shown in Fig. 9(b). As in Sec. IV B, we analyze the position fluctuations of the same 5998 particles using CNN3, leading to 5998

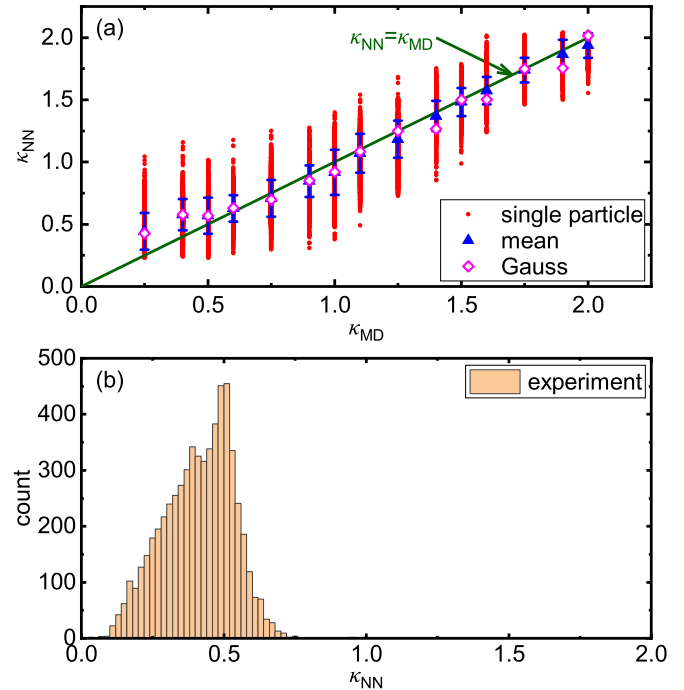


FIG. 9. Our determined κ_{NN} results using trained CNN3 with (a) the test simulation dataset and (b) the experiment data, respectively. In (a), for all runs of the test simulation dataset, our trained CNN3 determines the κ_{NN} values for all particles, with the RMSE of about 0.155, significantly smaller than that of CNN2 in Fig. 8. The mean of κ_{NN} and the average from the Gaussian fitting of κ_{NN} are still both nearly the same as the specified screening parameter value κ_{MD} in the test simulation dataset. In (b), using our CNN3 with the experiment data, the prominent peak in the distribution of the determined κ_{NN} values is located at ≈ 0.498 , agreeing well with $\kappa = 0.47$ obtained from the phonon spectra in [29].

determined values of κ_{NN} . The distribution of these obtained 5998 κ_{NN} values from CNN3, plotted in Fig. 9(b), clearly exhibits a prominent peak at ≈ 0.498 , also agreeing well with $\kappa = 0.47$ from the phonon spectra [29]. Clearly, the distribution of the obtained κ_{NN} values in Fig. 9(b) is much more concentrated than that from CNN2 in Fig. 8(b). For example, the count for $\kappa_{NN} \geq 0.8$ is nearly zero in Fig. 9(b); however, in Fig. 8(b), the tail of the count for $\kappa_{NN} \geq 0.8$ is still substantial. In short, the performance of CNN3 in accurately determining the κ value in the 2D dusty plasma experiment from the single particle information is better than CNN2 since the adjustable time information is incorporated in CNN3.

V. DISCUSSIONS

In fact, due to the same time interval between neighboring frames, the particle position fluctuation data used in CNN1 and CNN3 do imply the particle velocity information, which is completely lacking while using CNN2. From our understanding, among these three CNNs, the best performance of CNN1 with the least RMSE is mainly due to the exactly same time duration between neighboring frames in the training and test datasets. While using CNN3, to mimic the operations in typical dusty plasma experiments, the time duration between

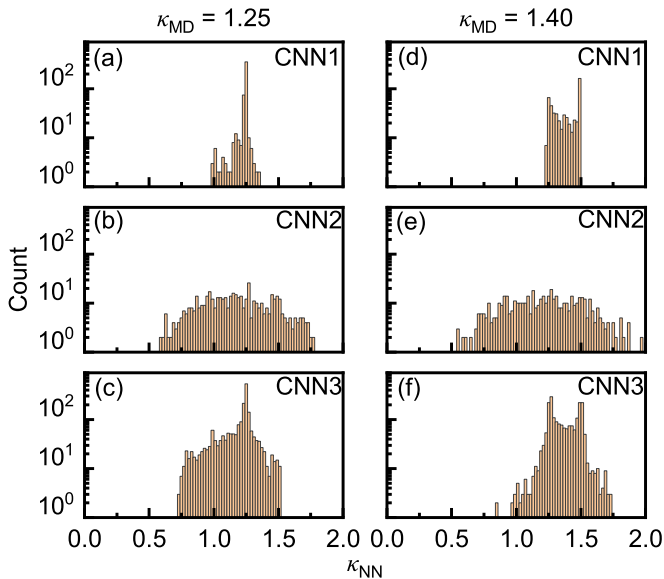


FIG. 10. The distribution of the determined κ_{NN} values from CNN1, CNN2, and CNN3, for the test simulation data with the specified κ_{MD} values of 1.25 and 1.40, respectively. Note that in the training dataset used here, $\kappa = 1.25$ is included, while $\kappa = 1.40$ is not. From (a) and (c), for the particle position fluctuation data of $\kappa_{MD} = 1.25$, the distribution of the determined κ_{NN} values exhibits a prominent peak at $\kappa_{MD} = 1.25$. However, from (d) and (f), for the data of $\kappa_{MD} = 1.40$, the distribution of the determined κ_{NN} values exhibits two distinctive peaks at $\kappa_{MD} = 1.25$ and 1.50 , respectively, just the two closest κ_{MD} values in the training dataset. For (b) and (e), due to the larger uncertainty of the determined κ_{NN} value and the wider distribution, the peak feature is not as clear. Note that (a), (b), (d), and (e) contain 512 κ_{NN} values determined by either CNN1 or CNN2 from the position fluctuation data of 512 different particles in the test simulations, while (c) and (f) contain 4×512 κ_{NN} values determined by CNN3 from the position fluctuation data of 512 different particles with four random values of the time duration in the test simulation dataset.

neighboring frames of the training data is not exactly the same as that of the test data, leading to a slightly larger RMSE, i.e., about twice the value of CNN1. When we use CNN2, the time-related information is completely removed and the spatial distribution of the particle position fluctuations is used to determine κ_{NN} , reasonably leading to the largest RMSE, which is about 3.5 times the value of CNN1.

In addition to reporting the mean of the determined κ_{NN} values above, in Fig. 10, we plot their distribution from these three CNNs for two test simulation runs of $\kappa_{MD} = 1.25$ and 1.40, respectively. From Figs. 10(a)–10(c), for each CNN, the distribution of the determined κ_{NN} values always exhibits a prominent peak around the κ_{MD} , also very close to its mean value, especially for CNN1 and CNN3. However, from Figs. 10(d) and 10(f), for either CNN1 or CNN3, the corresponding distribution of the determined κ_{NN} values surprisingly exhibits two distinctive peaks at the two sides of κ_{MD} , where the weighted mean value agrees well with the specified κ_{MD} value of 1.40. In Fig. 10(e), the distribution of the determined κ_{NN} values does not have a significant two-peak feature, unlike Figs. 10(d) and 10(f).

From the results in Fig. 10, we believe that the generalization ability of these three CNNs is really excellent. From Table I, in the test dataset, there are seven κ_{MD} values which do not exist in the training dataset, including $\kappa_{MD} = 1.4$ in Figs. 10(d)–10(f). Although none of these CNNs are trained with these seven κ_{MD} values, including $\kappa_{MD} = 1.40$, the mean of the determined κ_{NN} values is still very close to each κ_{MD} value, indicating well their excellent generalization ability. For the test data with the same κ value as the training data, due to their similarity, it is not surprising that the distribution of the determined κ_{NN} values is centered at the specified κ_{MD} value, as shown in Figs. 10(a) and 10(c). However, for the test data with the κ value that is not included in the training dataset, both CNN1 and CNN3 search the training data with all existing κ values to find the two closest κ values, leading to the exhibited two-peak feature as in Figs. 10(d) and 10(f), with the weighted mean value agreeing with the specified κ value in the test simulations. For CNN2, the uncertainty of the determined κ_{NN} value is larger and the distribution is also much wider [Figs. 10(b) and 10(e)], so that the peak feature is also not as clear.

As shown in Table I, when κ_{MD} is close to 0.25 or 2.00, the mean of the determined κ_{NN} values deviates from κ_{MD} the most, especially for CNN2 and CNN3. We think that this feature is probably caused by two facts. First, the distribution of the determined κ_{NN} values is broadened, especially for CNN2 and CNN3, as shown in Fig. 10. Second, in our current investigation, κ_{MD} varies only in the range of [0.25, 2.00], resulting in the same range of the determined κ_{NN} values. From the first fact, when κ_{MD} is close to 0.25, the value of the determined κ_{NN} values should be distributed to a wider range of both smaller and larger than κ_{MD} . However, since κ_{MD} only varies in the range of [0.25, 2.00], the obtained κ_{NN} cannot be smaller than 0.25. As a result, the mean of the obtained κ_{NN} values is more strongly biased by the obtained κ_{NN} larger than κ_{MD} , as shown in Table I. Similarly, when κ_{MD} is close to 2.00, the mean of the obtained κ_{NN} values is significantly smaller than κ_{MD} since it is more biased by those smaller than κ_{MD} . Meanwhile, due to the small broadening of the distribution of the determined κ_{NN} from CNN1, this feature is the least obvious for CNN1. By either expanding the value range of κ_{MD} or further upgrading the structure of CNN2 and CNN3 to improve their performance, this feature may be reduced in the future.

Here, we would like to discuss some complicated problems related to some 2D dusty plasma experiments. If the interparticle interaction in the 2D dusty plasma experiment deviates from pure Yukawa repulsion, for example, including the significant ion wake effect, our current CNNs trained by the LD simulation described in Sec. II can no longer be used to analyze this experiment. Instead, the CNNs should be trained using a new simulation dataset, in which the ion wake effect is taken into account. Note that for some 2D dusty plasma experiment data with the significant nonuniform problem not reported here, we may divide the field of view of the experimental images into different subregions; as a result, one may perform the same machine learning method reported here for each subregion, where the lattice constant is still nearly uniform.

In fact, our CNN3 may be further updated to improve its performance. From our understanding, the current

performance of CNN3 is not the best because its structure is not complicated enough, so that our CNN is not able to fully learn and remember enough features of the particle position fluctuation data with the adjustable time duration. In the future, we may probably deepen our CNN model, or even change its structure, so that it has the ability to learn more complicated features in the particle fluctuation data. Meanwhile, simply increasing the amount of the training data may also improve the performance of CNN3.

VI. SUMMARY

To summarize, using three CNNs of machine learning, we successfully determine the global property of the screening parameter in the 2D dusty plasma simulations and the experiment, purely by using the individual particle position fluctuations. For the individual particle position fluctuation data, the time interval between neighboring frames in units of the inverse of the dusty plasma frequency is a key parameter. If this time interval of the test data is the same as that of the training data, our CNN1 that is used here is able to accurately determine the screening parameter value, with the RMSE of only 0.081. In dusty plasma experiments, although the time interval between neighboring frames in the unit of s is clearly known, the dusty plasma frequency varies for different conditions. As a result, in experiments, the time interval between neighboring frames in units of the inverse of the dusty plasma frequency ω_{pd}^{-1} is generally unknown, so that it is impossible to match this time interval in the simulation training data. To mimic using machine learning with experimental data analysis, we may completely ignore the time information in the training dataset by converting the particle position fluctuations into the particle position distribution, as used in CNN2. The

obtained κ_{NN} value from CNN2 is also close to the specified κ_{MD} in the test dataset, with the RMSE of 0.279. Another way to solve this unknown time interval in units of ω_{pd}^{-1} is to include an adjustable time range to rescale the particle position fluctuations in the training dataset, as used in CNN3, leading to a more accurately determined κ_{NN} value, with the RMSE of 0.155. In the analysis with the 2D dusty plasma experimental data, the distribution of the determined κ_{NN} values from either CNN2 or CNN3 has a prominent peak located at the κ_{NN} value, agreeing well with the κ value determined from the phonon spectra fitting in [29].

All of the results above clearly demonstrate that the global property of 2D dusty plasma can be accurately determined from individual particle position fluctuations using machine learning. One advantage of this machine learning method is its high efficiency in the data analysis. For example, our trained CNN2 and CNN3 are able to accurately determine the κ values for thousands of analyzed particles in the 2D solid dusty plasma experiment within a few seconds, greatly facilitating our analysis of experiment data. For future dusty plasma investigations, machine learning may be applied to diagnose the complicated environment conditions, such as the magnetic and electric fields in tokamaks, based on the detailed dynamics of the experimentally observed individual dust particles there.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Contract No. 12175159, the National MCF Energy R&D Program of China, Grant No. 2018YFE0310100, and the corresponding Supporting Project from the Suzhou Science and Technology Bureau.

-
- [1] R. K. Pathria and P. D. Beale, *Statistical Mechanics* (Academic, London, 2021).
 - [2] D. J. Evans and G. P. Morriss, *Statistical Mechanics of Nonequilibrium Liquids* (Academic, London, 1990).
 - [3] P. Eastman, *Introduction to Statistical Mechanics*, <https://web.stanford.edu/~peastman/statmech/>.
 - [4] A. Piel, *Plasma Physics* (Springer, Heidelberg, 2010).
 - [5] J. P. Hansen and I. R. McDonald, *The Theory of Simple Liquids* (Academic, Amsterdam, 1986).
 - [6] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon Press, Oxford, 1987).
 - [7] R. Lakes, *Viscoelastic Materials* (Cambridge University Press, Cambridge, 2009).
 - [8] P. M. Chaikin and T. C. Lubensky, *Principles of Condensed Matter Physics* (Cambridge University Press, Cambridge, 2000).
 - [9] Y. Feng, J. Goree, and B. Liu, Accurate particle position measurement from images, *Rev. Sci. Instrum.* **78**, 053704 (2007).
 - [10] Y. Feng, J. Goree, and B. Liu, Errors in particle tracking velocimetry with high-speed cameras, *Rev. Sci. Instrum.* **82**, 053707 (2011).
 - [11] H. M. Thomas and G. E. Morfill, Melting dynamics of a plasma crystal, *Nature (London)* **379**, 806 (1996).
 - [12] L. I. W.-T. Juan, C.-H. Chiang, and J. H. Chu, Microscopic particle motions in strongly coupled dusty plasmas, *Science* **272**, 1626 (1996).
 - [13] A. Melzer, A. Homann, and A. Piel, Experimental investigation of the melting transition of the plasma crystal, *Phys. Rev. E* **53**, 2757 (1996).
 - [14] M. S. Murillo, Critical Wave Vectors for Transverse Modes in Strongly Coupled Dusty Plasmas, *Phys. Rev. Lett.* **85**, 2514 (2000).
 - [15] W.-S. Duan and Y.-R. Shi, The effect of dust size distribution for two ion temperature dusty plasmas, *Chaos Solitons Fractals* **18**, 321 (2003).
 - [16] R. L. Merlino and J. A. Goree, Dusty plasmas in the laboratory, industry, and space, *Phys. Today* **57(7)**, 32 (2004).
 - [17] G. J. Kalman, P. Hartmann, Z. Donkó, and M. Rosenberg, Two-Dimensional Yukawa Liquids: Correlation and Dynamics, *Phys. Rev. Lett.* **92**, 065001 (2004).
 - [18] V. E. Fortov, A. V. Ivlev, S. A. Khrapak, A. G. Khrapak, and G. E. Morfill, Complex (dusty) plasmas: Current status, open issues, perspectives, *Phys. Rep.* **421**, 1 (2005).
 - [19] G. E. Morfill and A. V. Ivlev, Complex plasmas: An interdisciplinary research field, *Rev. Mod. Phys.* **81**, 1353 (2009).

- [20] M. Bonitz, C. Henning, and D. Block, Complex plasmas: A laboratory for strong correlations, *Rep. Prog. Phys.* **73**, 066501 (2010).
- [21] A. Melzer, A. Schella, J. Schablinski, D. Block, and A. Piel, Analyzing the liquid state of two-dimensional dust clusters: The instantaneous normal mode approach, *Phys. Rev. E* **87**, 033107 (2013).
- [22] E. Thomas, U. Konopka, R. L. Merlino, and M. Rosenberg, Initial measurements of two- and three-dimensional ordering, waves, and plasma filamentation in the magnetized dusty plasma experiment, *Phys. Plasmas* **23**, 055701 (2016).
- [23] Y.-F. He, B.-Q. Ai, C.-X. Dai, C. Song, R.-Q. Wang, W.-T. Sun, F.-C. Liu, and Y. Feng, Experimental Demonstration of a Dusty Plasma Ratchet Rectification and Its Reversal, *Phys. Rev. Lett.* **124**, 075001 (2020).
- [24] J. Goree, Charging of particles in a plasma, *Plasma Sources Sci. Technol.* **3**, 400 (1994).
- [25] Y.-N. Wang, L.-J. Hou, and X. Wang, Self-Consistent Nonlinear Resonance and Hysteresis of a Charged Microparticle in a rf Sheath, *Phys. Rev. Lett.* **89**, 155001 (2002).
- [26] V. Nosenko and J. Goree, Shear Flows and Shear Viscosity in a Two-Dimensional Yukawa System (Dusty Plasma), *Phys. Rev. Lett.* **93**, 155004 (2004).
- [27] Y. Feng, J. Goree, and B. Liu, Solid Superheating Observed in Two-Dimensional Strongly Coupled Dusty Plasma, *Phys. Rev. Lett.* **100**, 205007 (2008).
- [28] Y. Feng, J. Goree, and B. Liu, Viscoelasticity of 2D Liquids Quantified in a Dusty Plasma Experiment, *Phys. Rev. Lett.* **105**, 025002 (2010).
- [29] D. Huang, S. Lu, X.-Q. Shi, J. Goree, and Y. Feng, Fluctuation theorem convergence in a viscoelastic medium demonstrated experimentally using a dusty plasma, *Phys. Rev. E* **104**, 035207 (2021).
- [30] Y. Feng, J. Goree, and B. Liu, Observation of Temperature Peaks due to Strong Viscous Heating in a Dusty Plasma Flow, *Phys. Rev. Lett.* **109**, 185002 (2012).
- [31] Y. Feng, J. Goree, B. Liu, and E. G. D. Cohen, Green-Kubo relation for viscosity tested using experimental data for a two-dimensional dusty plasma, *Phys. Rev. E* **84**, 046412 (2011).
- [32] K. Qiao, J. Kong, J. Carmona-Reyes, L. S. Matthews, and T. W. Hyde, Mode coupling and resonance instabilities in quasi-two-dimensional dust clusters in complex plasmas, *Phys. Rev. E* **90**, 033109 (2014).
- [33] U. Konopka, G. Morfill, and L. Ratke, Measurement of the Interaction Potential of Microspheres in the Sheath of a rf Discharge, *Phys. Rev. Lett.* **84**, 891 (2000).
- [34] S. Ichimaru, Strongly coupled plasmas: High-density classical plasmas and degenerate electron liquids, *Rev. Mod. Phys.* **54**, 1017 (1982).
- [35] Z. Donkó, J. Goree, P. Hartmann, and K. Kutasi, Shear Viscosity and Shear Thinning in Two-Dimensional Yukawa Liquids, *Phys. Rev. Lett.* **96**, 145003 (2006).
- [36] C.-L. Chan and L. I, Microstructural Evolution and Irreversibility in the Viscoelastic Response of Mesoscopic Dusty-Plasma Liquids, *Phys. Rev. Lett.* **98**, 105002 (2007).
- [37] T. Ott and M. Bonitz, Is Diffusion Anomalous in Two-Dimensional Yukawa Liquids?, *Phys. Rev. Lett.* **103**, 195001 (2009).
- [38] M. S. Murillo, Strongly coupled plasma physics and high energy-density matter, *Phys. Plasmas* **11**, 2964 (2004).
- [39] A. Melzer, A. Schella, J. Schablinski, D. Block, and A. Piel, Instantaneous Normal Mode Analysis of Melting of Finite Dust Clusters, *Phys. Rev. Lett.* **108**, 225001 (2012).
- [40] P. Hartmann, A. Douglass, J. C. Reyes, L. S. Matthews, T. W. Hyde, A. Kovács, and Z. Donkó, Crystallization Dynamics of a Single Layer Complex Plasma, *Phys. Rev. Lett.* **105**, 115004 (2010).
- [41] S. Khrapak and L. Couëdel, Dispersion relations of Yukawa fluids at weak and moderate coupling, *Phys. Rev. E* **102**, 033207 (2020).
- [42] P. Bajaj, S. Khrapak, V. Yaroshenko, and M. Schwabe, Spatial distribution of dust density wave properties in fluid complex plasmas, *Phys. Rev. E* **105**, 025202 (2022).
- [43] J. D. Williams, E. Thomas, L. Couëdel, A. V. Ivlev, S. K. Zhdanov, V. Nosenko, H. M. Thomas, and G. E. Morfill, Kinetics of the melting front in two-dimensional plasma crystals: Complementary analysis with the particle image and particle tracking velocimetry, *Phys. Rev. E* **86**, 046401 (2012).
- [44] S. Jaiswal, T. Hall, S. LeBlanc, R. Mukherjee, and E. Thomas, Effect of magnetic field on the phase transition in a dusty plasma, *Phys. Plasmas* **24**, 113703 (2017).
- [45] L. Couëdel, V. Nosenko, M. Rubin-Zuzic, S. Zhdanov, Y. Elskens, T. Hall, and A. V. Ivlev, Full melting of a two-dimensional complex plasma crystal triggered by localized pulsed laser heating, *Phys. Rev. E* **97**, 043206 (2018).
- [46] L. Couëdel and V. Nosenko, Stability of two-dimensional complex plasma monolayers in asymmetric capacitively coupled radio-frequency discharges, *Phys. Rev. E* **105**, 015210 (2022).
- [47] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, *Science* **349**, 255 (2015).
- [48] C. Rea, R. S. Granetz, K. Montes, R. A. Tinguely, N. Eidietis, J. M. Hanson, and B. Sammuli, Disruption prediction investigations using machine learning tools on DIII-D and Alcator C-Mod, *Plasma Phys. Control. Fusion* **60**, 084004 (2018).
- [49] C. Rea and R. S. Granetz, Exploratory machine learning studies for disruption prediction using large databases on DIII-D, *Fusion Sci. Technol.* **74**, 89 (2018).
- [50] K. J. Montes, C. Rea, R. S. Granetz, R. A. Tinguely, N. Eidietis, O. M. Meneghini, D. L. Chen, B. Shen, B. J. Xiao, K. Erickson *et al.*, Machine learning for disruption warnings on Alcator C-Mod, DIII-D, and EAST, *Nucl. Fusion* **59**, 096015 (2019).
- [51] M. S. Murillo, M. Marciante, and L. G. Stanton, Machine Learning Discovery of Computational Model Efficacy Boundaries, *Phys. Rev. Lett.* **125**, 085503 (2020).
- [52] H. Li, Y. Fu, J. Li, and Z. Wang, Machine learning of turbulent transport in fusion plasmas with neural network, *Plasma Sci. Technol.* **23**, 115102 (2021).
- [53] J. Degraeve, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas *et al.*, Magnetic control of tokamak plasmas through deep reinforcement learning, *Nature (London)* **602**, 414 (2022).
- [54] C. Dietz, T. Kretz, and M. H. Thoma, Machine-learning approach for local classification of crystalline structures in multiphase systems, *Phys. Rev. E* **96**, 011301(R) (2017).
- [55] C. Dietz, R. Bergert, B. Steinmüller, M. Kretschmer, S. Mitic, and M. H. Thoma, fcc-bcc phase transition in plasma crystals using time-resolved measurements, *Phys. Rev. E* **97**, 043203 (2018).

- [56] C. Dietz, J. Budak, T. Kamprich, M. Kretschmer, and M. H. Thoma, Phase transition in electrorheological plasmas, *Contrib. Plasma Phys.* **61**, e202100079 (2021).
- [57] H. Huang, M. Schwabe, and C.-R. Du, Identification of the interface in a binary complex plasma using machine learning, *J. Imag.* **5**, 36 (2019).
- [58] Z. Wang, J. Xu, Y. E. Kovach, B. T. Wolfe, E. Thomas, H. Guo, J. E. Foster, and H.-W. Shen, Microparticle cloud imaging and tracking for data-driven plasma science, *Phys. Plasmas* **27**, 033703 (2020).
- [59] M. Himpel and A. Melzer, Fast 3D particle reconstruction using a convolutional neural network: Application to dusty plasmas, *Mach. Learn.: Sci. Technol.* **2**, 045019 (2021).
- [60] Z. Ding, J. Yao, Y. Wang, C. Yuan, Z. Zhou, A. A. Kudryavtsev, R. Gao, and J. Jia, Machine learning combined with Langmuir probe measurements for diagnosis of dusty plasma of a positive column, *Plasma Sci. Technol.* **23**, 095403 (2021).
- [61] C. Liang, Z. Ma, Z. Sun, X. Zhang, X. You, Z. Liu, G. Zuo, J. Hu, and Y. Feng, Demonstration of object location, classification, and characterization by developed deep learning dust ablation trail analysis code package using plasma jets, *Rev. Sci. Instrum.* **94**, 023506 (2023).
- [62] H. Huang, V. Nosenko, H.-X. Huang-Fu, H. M. Thomas, and C.-R. Du, Machine learning in the study of phase transition of two-dimensional complex plasmas, *Phys. Plasmas* **29**, 073702 (2022).
- [63] W. Yu, J. Cho, and J. C. Burton, Extracting forces from noisy dynamics in dusty plasmas, *Phys. Rev. E* **106**, 035303 (2022).
- [64] X. Wang, A. Bhattacharjee, and S. Hu, Longitudinal and Transverse Waves in Yukawa Crystals, *Phys. Rev. Lett.* **86**, 2569 (2001).
- [65] S. Nunomura, J. Goree, S. Hu, X. Wang, A. Bhattacharjee, and K. Avinash, Phonon Spectrum in a Plasma Crystal, *Phys. Rev. Lett.* **89**, 035001 (2002).
- [66] Z. Donkó, J. Goree, P. Hartmann, and B. Liu, Time-correlation functions and transport coefficients of two-dimensional Yukawa liquids, *Phys. Rev. E* **79**, 026401 (2009).
- [67] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature (London)* **521**, 436 (2015).
- [68] B. Liu, J. Goree, V. Nosenko, and L. Boufendi, Transverse optical mode in a one-dimensional Yukawa chain, *Phys. Plasmas* **10**, 9 (2003).
- [69] Y. Feng, B. Liu, and J. Goree, Rapid heating and cooling in two-dimensional Yukawa systems, *Phys. Rev. E* **78**, 026415 (2008).
- [70] B. Liu and J. Goree, Shear Viscosity of Two-Dimensional Yukawa Systems in the Liquid State, *Phys. Rev. Lett.* **94**, 185002 (2005).
- [71] J. Yang, J. Wright, T. S. Huang, and Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Process.* **19**, 2861 (2010).
- [72] <https://opencv.org> (unpublished).
- [73] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks* **61**, 85 (2015).
- [74] <https://pytorch.org> (unpublished).
- [75] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* **15**, 1929 (2014).
- [76] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [77] C. J. Willmott and K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Clim. Res.* **30**, 79 (2005).