


RESEARCH ARTICLE | MAY 23 2025

## Numerical simulations on DC-driven low-temperature plasmas by physics-informed neural networks

Wen-Kai Li; Yuan-Tao Zhang 



*J. Appl. Phys.* 137, 203304 (2025)

<https://doi.org/10.1063/5.0253746>



### Articles You May Be Interested In

Physics-informed neural network (PINNs) for convection equations in polymer flooding reservoirs

*Physics of Fluids* (March 2025)

An efficient framework for solving forward and inverse problems of nonlinear partial differential equations via enhanced physics-informed neural network based on adaptive learning

*Physics of Fluids* (October 2023)

Two-dimensional temperature field inversion of turbine blade based on physics-informed neural networks

*Physics of Fluids* (March 2024)



Journal of Applied Physics

## Special Topics Open for Submissions

[Learn More](#)

# Numerical simulations on DC-driven low-temperature plasmas by physics-informed neural networks

Cite as: J. Appl. Phys. **137**, 203304 (2025); doi: [10.1063/5.0253746](https://doi.org/10.1063/5.0253746)

Submitted: 18 December 2024 · Accepted: 2 May 2025 ·

Published Online: 23 May 2025



Wen-Kai Li and Yuan-Tao Zhang<sup>a)</sup>

## AFFILIATIONS

School of Electrical Engineering, Shandong University, Jinan, Shandong Province 250061, People's Republic of China

<sup>a)</sup>Author to whom correspondence should be addressed: [ytzhang@sdu.edu.cn](mailto:ytzhang@sdu.edu.cn)

## ABSTRACT

In recent years, artificial intelligence (AI) technology is empowering various fields, and the combination of AI and numerical simulation of Low-Temperature Plasmas (LTPs) has attracted widespread attention. In this study, the physics-informed deep neural networks (PINNs) are used to solve the fluid model, which describes a one-dimensional DC-driven discharge system. PINNs use a multilayer perceptron to represent the solutions to the poisson equation, continuity equation, and drift-diffusion approximation equations in fluid model and employ automatic differentiation to compute the derivatives of these field variables, thereby constructing the residuals of the physical equations. The training dataset consists of randomly sampled spatial and temporal coordinates along with their corresponding distribution functions, generated from fluid simulations. The gradient descent algorithm minimizes the sum of the residuals of the physical equations and the data loss to update the model parameters, enabling PINNs to fit observed data while satisfying the fluid equations. The simulation results show that this mesh-less method can effectively solve the fluid model at any spatial-temporal resolution instead of discretization method with a fixed spatial-temporal resolution, providing an alternative for numerical solution of fluid models for LTP in the era of AI.

24 May 2025 06:45:25

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0253746>

## I. INTRODUCTION

LTPs with low gas temperature and plenty of reactive species are widely used in industrial applications and emerging fields, such as energy and chemical engineering,<sup>1</sup> agricultural production,<sup>2,3</sup> material preparation and modification,<sup>4,5</sup> plasma medicine,<sup>6,7</sup> and environmental remediation.<sup>8,9</sup> Experimental measurement methods and numerical simulation methods, as primary approaches for investigating the characteristics and evolution of plasma discharges, have been widely applied. Although experimental measurement methods can directly obtain data, the complexity and dynamic nature of plasmas still limit the precision and real-time capabilities of experimental measurements.

Numerical simulation, as another important method for studying LTP, can effectively complement the limitations of experimental diagnostics. It can provide a more intuitive understanding of the multi-scale and multi-physics characteristics of discharge plasmas. Currently, the commonly used plasma numerical simulation methods include Particle-in-cell Monte Carlo (PIC-MCC) and

fluid simulation of plasma.<sup>10,11</sup> Regardless of the method used, they are all based on traditional numerical algorithms, such as the Finite Difference Method (FDM), Finite Element Method (FEM), and Finite Volume Method (FVM). Traditional numerical algorithms, while effective, face several disadvantages in mesh generation, including difficulties in handling complex geometries, high dependence on mesh density and shape, the time and resources required to generate high-quality meshes, challenges in resolving local details and multi-scale problems, low efficiency in parallel computing, and difficulties in accurately representing boundary conditions.<sup>12,13</sup> These drawbacks can limit the accuracy and efficiency of numerical simulations to some extent.

In recent years, the AI technology has been introduced into the fields of plasmas simulation, and the well-trained deep neural networks (DNNs) can be applied to predict the discharge characteristics in almost real time, such as the profiles of discharge current, electron density, electric field,<sup>14,15</sup> even the electron energy distribution function (EEDF) based on the data from the PIC-MCC

can be efficiently obtained in seconds,<sup>16</sup> which usually requires expensive computational resources. In the numerical study on plasma medicine, the DNN is also used to assist in unveiling the interactions of plasma and biological molecules by reducing the computational workload of reactive molecular dynamics simulations.<sup>17</sup> It is evident that the data-driven models particularly those represented by DNNs have exhibited substantial computational advantages in plasma simulations.<sup>18,19</sup> Physics-informed neural networks (PINNs) have also become an efficient new tool for solving forward and inverse problems involving partial differential equations (PDEs). The core idea of PINNs is to integrate the governing equations of physical systems into the loss function of neural networks, ensuring that the networks minimize PDE residuals while approximating the given conditions or observed data. Dissanayake and Phan-Thien<sup>20</sup> pioneered the use of neural networks to solve PDEs in the 20th century. Compared with mesh-based numerical methods such as finite element (FEM), finite difference (FDM), and finite volume (FVM), the advantage of PINNs lies in their ability to directly calculate spatiotemporal derivatives through automatic differentiation,<sup>21,22</sup> enabling a mesh-free approach that avoids mesh generation, which also indicates that PINNs can alleviate the curse of dimensionality to a certain extent.<sup>23,24</sup> PINNs have been demonstrated for various physics phenomena, including fluid mechanics,<sup>25–28</sup> heat transfer,<sup>29,30</sup> fluid–structure interaction,<sup>31</sup> electromagnetic propagation,<sup>32</sup> and quantum chemistry.<sup>33</sup> Although the validity of PINNs in LTPs computing is still not well explained, multi-hidden layer neural networks that follow universal approximation theorems are thought to be able to approximate a set of governing equations in the fluid model for LTPs simulations,<sup>14</sup> which lays a solid foundation for the practice of PINNs in the field of computational LTPs. However, due to the complex multi-particle plasma chemical reaction systems and multi-physics coupling relationships, numerical simulation studies of LTPs based on PINNs are still relatively scarce. Therefore, the introduction of PINNs into the numerical simulation of LTPs is highly necessary.

The main contents of this paper are given as follows: In Sec. I, the fluid model is briefly described and used to provide training data, and a general-purpose PINNs model suitable for fluid simulations is constructed. The efficiency and accuracy of the PINNs model are demonstrated by comparing the prediction results with those obtained from a fluid model solved by a traditional discretization method, given in Sec. II. Finally, a conclusion is summarized in Sec. III. The purpose of this paper is only to demonstrate the ability and advantages of PINNs and to provide an alternative for numerical solution of fluid models for LTPs. More reactions and particles will be considered in the PINNs in future investigations.

## II. DESCRIPTION OF NUMERICAL METHODS

### A. Fluid model of LTPs

Fluid model is well accepted in describing the LTPs at atmospheric pressure or low pressure, and the discretization process is commonly used to solve the fluid equations.<sup>34–36</sup> In this study, a one-dimensional direct current barrier discharge system controlled by the semiconductor material as an example to illustrate how PINNs solve the fluid equations. For this direct current discharge system, its structure is shown in Fig. 1, and  $V_0$  is the externally

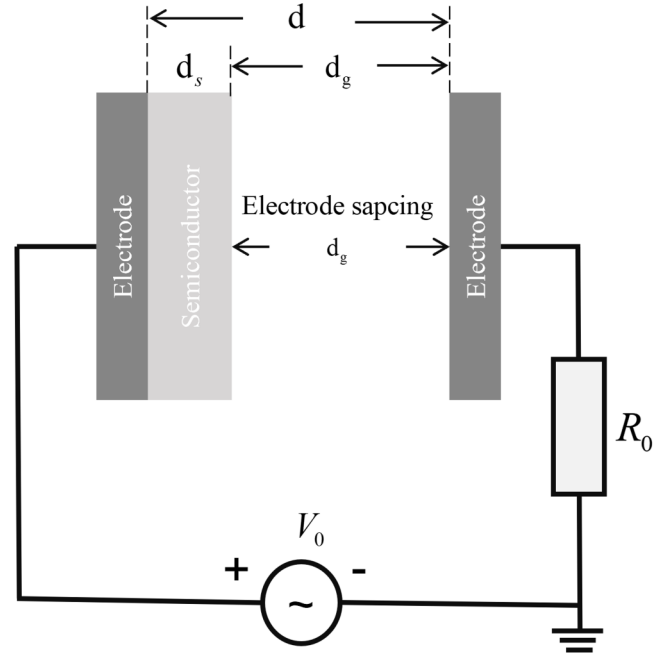


FIG. 1. The geometry of the one-dimensional DC discharge model.

applied DC voltage, and  $d_g$  and  $d_s$  represent the discharge space and the insulator plate thickness, respectively, which have been explored in detail experimentally and numerically.<sup>37–39</sup> In the fluid model, Poisson's equation is used to calculate the electric field, and the generation and loss of charged and neutral particles and neutral particle are generally discussed by continuity equations. The motion of particles is described using drift-diffusion approximation at high pressure. The one-dimensional form of those equations is given as follows:<sup>34–36,40,41</sup>

$$\frac{\partial E(x, t)}{\partial x} = \frac{q}{\epsilon} (N_i(x, t) - N_e(x, t)), \quad (1)$$

$$\frac{\partial N_{i,e}(x, t)}{\partial t} + \frac{\partial j_{i,e}(x, t)}{\partial x} = S(x, t), \quad (2)$$

$$j_{i,e}(x, t) = \pm \mu_{i,e} E(x, t) N_{i,e}(x, t) - D_{i,e} \frac{\partial N_{i,e}(x, t)}{\partial x}, \quad (3)$$

where  $i$  and  $e$  represent ions and electrons in the discharge space, respectively.  $N(x, t)$  is the number density of plasma species, and  $j(x, t)$  is the flux density in diffusion-drift approximation.  $\mu$  and  $D$  stand for drift and diffusion coefficients, respectively.  $E(x, t)$  represents the electric field.  $q$  and  $\epsilon$  represent the elementary charge and the permittivity of free space, respectively.  $S$  represents the particle source term, including the generation and disappearance of charged particles, and the specific mathematical expressions are as follows:

24 May 2025 06:45:25

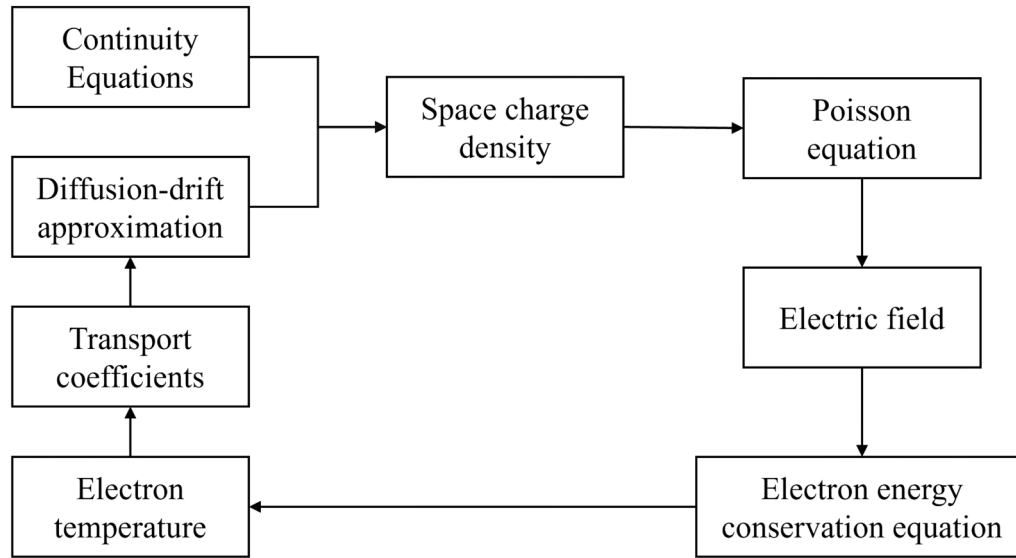


FIG. 2. Schematic diagram of the fluid model.

$$S(x, t) = A\mu_e E(x, t)N_e(x, t)e^{-\frac{Bp}{E(x,t)}}, \quad (4)$$

where  $p$  represents the voltage of the gas gap and  $A$  and  $B$  are parameters that depend on the type of gas and the pressure. The diagram of the specific fluid simulation method is shown in Fig. 2. It is only important to note that extremely frequent particle collisions at low pressure severely limit the choice of spatial grids and time steps in fluid simulations.

In this study, a DC-driven discharge system is investigated with nitrogen as working gas, and only collisional ionization reactions and two types of particles, namely, electrons and nitrogen ions, are considered in the model to quantitatively explore the discharge characteristics. The background gas pressure  $p$  was set to 4 Torr. The gas gap distance  $d_g$  and the thickness of the dielectric plate  $d_s$  were set to 0.1 and 0.15 cm, respectively. The applied voltage is 700 V, and the semiconductor conductivity is given as  $1.0 \times 10^{-6}(\Omega \text{ cm})^{-1}$ . Under the given discharge conditions, a spatially uniform but temporally oscillatory behavior of discharge current density can be generated from the fluid model with discretization method,<sup>36,42</sup> as shown in Fig. 3. The discharge current is approximately 15 mA, with an oscillation frequency of 375 kHz. And each current pulse has a duration of about  $1 \mu\text{s}$ , which agrees well with the experimental observation and computational exploration.<sup>37,39,43</sup>

## B. PINNs for solving fluid model

PINNs are a new paradigm that leverages recent advances in deep learning to infer solutions, parameters, and/or constitutive laws involving PDEs.<sup>3,22,44–46</sup> In this framework, the solution of PDEs is parametrized by a neural network that is trained to match the measurements of the system, while being constrained to approximately satisfy the underlying physical laws, and the requirements of the

amount of data were reduced. In general, the PDEs are given as the following form:

$$\mathcal{N}[u](x, t) = f(x, t), \quad x \in \Omega, t \in (0, T], \quad (5)$$

$$\mathcal{B}[u](x, t) = g(x, t), \quad x \in \partial\Omega, t \in (0, T], \quad (6)$$

$$u(x, 0) = h(x), \quad x \in \bar{\Omega}, \quad (7)$$

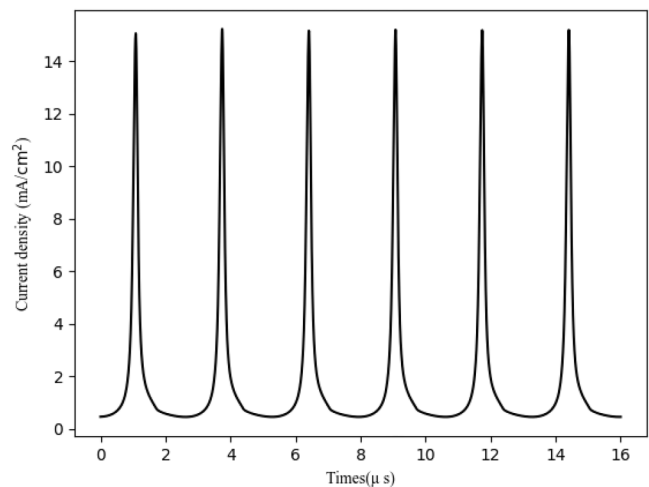


FIG. 3. Temporal evolution of discharge current density evolution.

24 May 2025 06:45:25

where  $\mathcal{N}[\cdot]$  is a differential operator and  $\mathcal{B}[\cdot]$  corresponds to Dirichlet, Neumann, Robin, or periodic boundary conditions. In addition, for time-dependent problems, we consider time  $t$  as a special component of  $x$ , and  $\Omega$  then also contains the temporal domain. In that case, initial conditions can be simply treated as a special type of boundary condition on the spatiotemporal domain. Following the original work given in Ref. 22,  $u(x, t)$  is approximated by a deep neural network  $u_\theta(x, t)$ , where  $\theta$  denotes all tunable parameters of the network (e.g., weights and biases). Then, a physics-informed model can be trained by minimizing the following composite loss function,

$$\mathcal{L}(\theta) = \lambda_s \mathcal{L}_s(\theta) + \lambda_r \mathcal{L}_r(\theta) + \lambda_{ub} \mathcal{L}_{ub}(\theta) + \lambda_{ic} \mathcal{L}_{ic}(\theta), \quad (8)$$

where  $\mathcal{L}_s$  is the loss term corresponding to sample data (if any), while  $\mathcal{L}_r$ ,  $\mathcal{L}_{ub}$ , and  $\mathcal{L}_{ic}$  are loss terms corresponding to not satisfying the PDE (5), the boundary condition (6), and the initial condition (7), respectively. Moreover, the parameters  $\{\lambda_s, \lambda_r, \lambda_{ub}, \lambda_{ic}\}$  correspond to weight coefficients in the loss function that can effectively assign a different learning rate to each individual loss term. These weights may be user-specified or tuned automatically during network training.<sup>47,48</sup>

In PINNs, the loss function typically consists of two components, namely, the physics constraint term and the data fitting term. These two parts work together to ensure that the neural network not only fits the known data points but also satisfies the given physical laws. In this study, the process of solving fluid equations based on deep learning technology will be given in details, showing that PINNs can be applied as a new tool to numerically solve the fluid model but without discretization. For Eqs. (1)–(4), various particle densities, electric fields, and flow densities are coupled through the Poisson equation, continuity equations, and drift-diffusion approximation equations, and the complex coupling relationship presents significant challenges for the training of PINNs for fluid model. Therefore, the fluid equations are combined, and the following residues are defined as follow:

$$res_1(x, t) := \frac{\partial E}{\partial x} - \frac{q}{\epsilon}(N_i - N_e), \quad (9)$$

$$res_2(x, t) := \frac{\partial N_e}{\partial t} - \mu_e \left( E \frac{\partial N_e}{\partial x} + N_e \frac{\partial E}{\partial x} \right) - D_e \frac{\partial^2 N_e}{\partial x^2} - A \mu_e \exp \left( -\frac{Bp}{E} \right) E N_e, \quad (10)$$

$$res_3(x, t) := \frac{\partial N_i}{\partial t} + \mu_i \left( E \frac{\partial N_i}{\partial x} + N_i \frac{\partial E}{\partial x} \right) - D_i \frac{\partial^2 N_i}{\partial x^2} - A \mu_i \exp \left( -\frac{Bp}{E} \right) E N_e. \quad (11)$$

Then, the spatial distributions of the fundamental fields described in the fluid model, such as electron density, ion density, and electric fields, can be solved using the aforementioned equations. This constitutive relation is coupled to Poisson's equation, continuity equations, and the drift-diffusion approximation. We aim to exploit

via using PINNs in order to infer the basic field's spacetime distribution from area measurements. These residuals can be then used as constraints during the training of the neural networks  $f(x, t; \theta)$  in order to encourage them to produce physically consistent predictions. For the first and second derivatives of the field, such as  $\frac{\partial N}{\partial x}$ ,  $\frac{\partial^2 N}{\partial x^2}$ , automatic differentiation is used to compute them within the neural network. Automatic differentiation dynamically computes derivatives through the program's execution path, avoiding truncation and round-off errors present in traditional numerical methods (such as finite difference methods). Additionally, automatic differentiation can efficiently compute gradients for high-dimensional problems, which significantly reduces computational time and memory usage.<sup>22</sup>

This part of the loss function corresponds to the collocation points. These are points that we randomly choose inside the domains using a latin-hypercube sampling strategy,<sup>49</sup>

$$\mathcal{L}_{residual} = \frac{\lambda_1}{N_{pde}} \sum_{i=1}^{N_{pde}} (res_1(x_i, t_i; \theta))^2 + \frac{\lambda_2}{N_{pde}} \sum_{i=1}^{N_{pde}} (res_2(x_i, t_i; \theta))^2 + \frac{\lambda_3}{N_{pde}} \sum_{i=1}^{N_{pde}} (res_3(x_i, t_i; \theta))^2, \quad (12)$$

where  $N_{pde}$  represents the number of collocation points.  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the weights for different losses, set to 1, 2, and 1, respectively. We increased  $\lambda_2$  to 2 (from its initial value of 1) to mitigate the gradient dominance issue caused by the smaller magnitude of electron density (one order of magnitude lower than ion density and electric field even after normalization). This adjustment ensures that the network prioritizes physical consistency for electron density prediction.

For the next step, a given neural network  $f(x, t; \theta)$  should be defined to represent the solution to Eqs. (9)–(11), which aims to approximate the following mapping:

$$[x, t] \mapsto [N_e(x, t), N_i(x, t), E(x, t)].$$

This part of the loss function ensures that the network fits the simulation data, including initial conditions, boundary conditions, and some interior points. The measurement of loss function has the form

$$\mathcal{L}_{measurements} = \frac{w_1}{N_E} \sum_{i=1}^{N_E} |E(x_i, t_i) - f(x_i, t_i; \theta)|^2 + \frac{w_2}{N_{N_e}} \sum_{i=1}^{N_{N_e}} |N_e(x_i, t_i) - f(x_i, t_i; \theta)|^2 + \frac{w_3}{N_{N_i}} \sum_{i=1}^{N_{N_i}} |N_i(x_i, t_i) - f(x_i, t_i; \theta)|^2, \quad (13)$$

where  $N_E$ ,  $N_{N_e}$ , and  $N_{N_i}$  represent the number of measurement points for electric field, electron density, and ion density.  $w_1$ ,  $w_2$ , and  $w_3$  are the weights for different losses, set to 1, 1, and 1, respectively.  $w$  is initially set to 1 for all fields initially to ensure equal priority in data fitting.

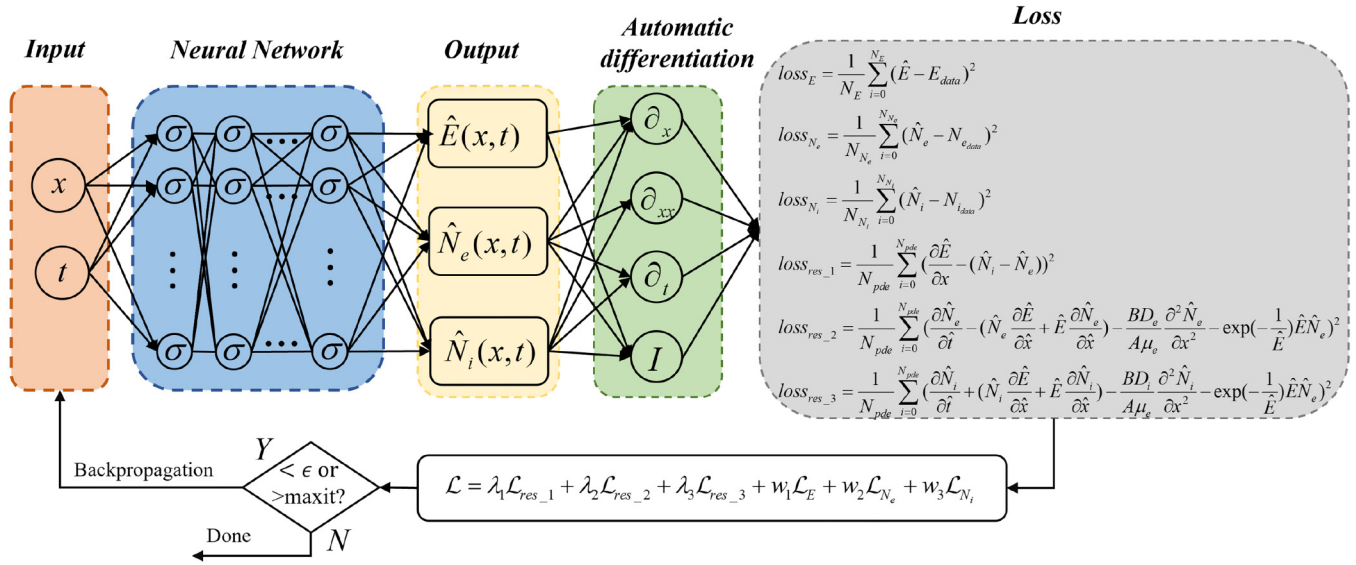


FIG. 4. Schematic diagram of PINNs structure for solving the fluid model.

The total loss function is finally given as

$$\mathcal{L} = \mathcal{L}_{\text{residual}} + \mathcal{L}_{\text{measurements}}, \quad (14)$$

which is imposed to encourage the neural networks to find a particular set of parameters that make their predictions consistent with the underlying fluid equations by translating to have the minimum residual. Thus, based on the structure of PINNs for fluid model shown in Fig. 4, by satisfying the given loss function together with matching the measurements at particular points, the solution of the fluid equations, namely, the spatiotemporal profiles of electron and ion density and electric field can be obtained.

In this study, one cycle of the discharge was selected for investigation. The electron density, ion density, and electric field were sampled once every 100 time steps, resulting in a total of 200 datasets. And the discharge space was divided into 80 grid cells, with one sample taken at each grid point. Their spatiotemporal distributions are shown in Fig. 5. In this case, the architecture of PINNs is configured as follows: there are seven hidden layers, each containing 60 neurons. We tested configurations ranging from six to eight hidden layers (with 50–80 neurons per layer) and selected the simplest architecture (seven hidden layers, 60 neurons each) that achieved optimal validation accuracy, balancing computational efficiency and model performance. Each layer uses the tanh function

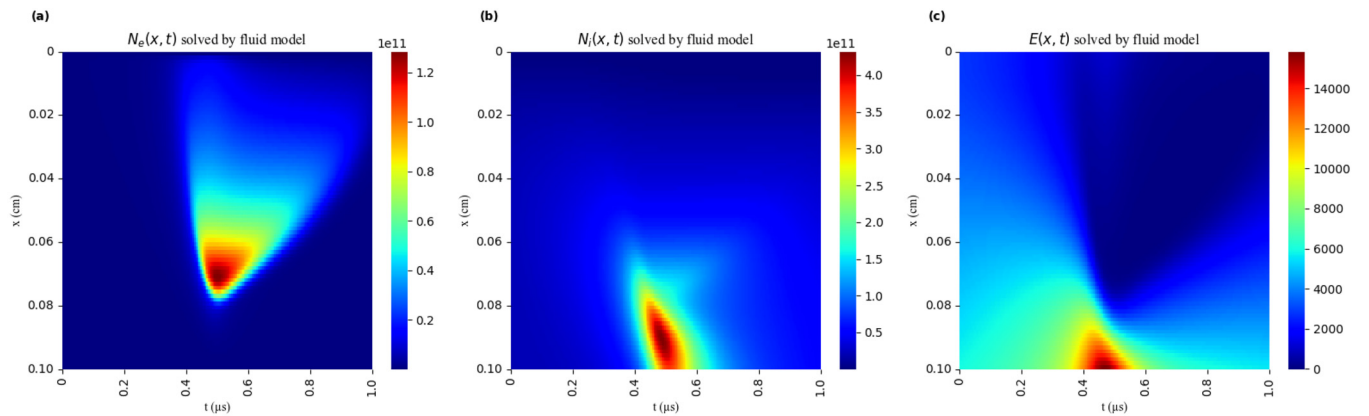


FIG. 5. Spatiotemporal distributions of electron density (a), ion density (b), and electric field (c) over one cycle from the fluid model.



as the activation function. The tanh activation function was chosen for its zero-centered output symmetry, which mitigates gradient vanishing issues in deeper layers compared to sigmoid or ReLU, particularly given the bounded nature of our input data. The model employs the Adam optimizer for parameter updates, with an initial learning rate set at 0.001 and a stepwise decay schedule that multiplies the rate by 0.8 every 1000 training steps, strategically balancing convergence speed and numerical stability. Weights were initialized using the Xavier/Glorot method, which scales initial values based on the number of input and output units to maintain consistent variance across layers, thereby avoiding saturation in tanh activations. The input consists of spatial-temporal coordinates, and the outputs are the electric field, electron density, and ion density. For sampling the density and electric field, a uniform sampling method is employed at the boundary conditions and initial conditions, totaling 480 points. Additionally, to better learn the density and electric field, 216 points are uniformly sampled within the domain according to time and space. The initial learning rate is set to  $10^{-3}$  and it decays by half every 5000 iterative steps during the training process, with a total of 60 000 training iterations. To accelerate model training and validation, we used a single NVIDIA GeForce RTX 4060Ti GPU. The training process took approximately 0.5 h on a single GPU. Both the training and validation of the model were completed using the aforementioned hardware and software configurations.

### C. Data preprocessing for PINNs

In Eqs. (13)–(15), the order of magnitude of the different physical variables, such as electron density, ion density area, and electric field, has a significant relative difference, e.g.,  $N_e \sim 10^{10}/\text{cm}^{-3}$ ,  $N_i \sim 10^{11}/\text{cm}^{-3}$ , and  $E \sim 10^4 \text{ V}/\text{cm}^{-3}$ , which casts great difficulty on the training of the neural network.<sup>22</sup> The substantial disparity in the magnitudes of the parameters poses a systematic challenge for training the PINNs, as this scale difference significantly affects the magnitude of the gradients back-propagated to update the network parameters during training. To overcome this problem, we employ a non-dimensionalization technique with the purpose of scaling the input and the output of the neural networks in a proper scale for training the neural networks more efficiently.

By dimensional analysis, the independent dimensionless parameters of the model are identified. It is convenient to introduce the following dimensionless times, lengths, and fields:<sup>50</sup>

$$\hat{x} = \frac{x}{x_0}, \quad \hat{t} = \frac{t}{t_0}, \quad \hat{E} = \frac{E}{E_0}, \quad \hat{N}_e = \frac{N_e}{N_0}, \quad \hat{N}_i = \frac{N_i}{N_0},$$

where

$$x_0 = \frac{1}{Ap}, \quad E_0 = Bp, \quad t_0 = \frac{x_0}{\mu_e E_0}, \quad N_0 = \frac{\varepsilon_0 E_0}{qx_0}.$$

Thus, the residues now take the form

$$res_1(\hat{x}, \hat{t}) := \frac{\partial \hat{E}}{\partial \hat{x}} - (\hat{N}_i - \hat{N}_e), \quad (15)$$

$$res_2(\hat{x}, \hat{t}) := \frac{\partial \hat{N}_e}{\partial \hat{t}} - \left( \hat{E} \frac{\partial \hat{N}_e}{\partial \hat{x}} + \hat{N}_e \frac{\partial \hat{E}}{\partial \hat{x}} \right) - \frac{B D_e}{A \mu_e} \frac{\partial^2 \hat{N}_e}{\partial \hat{x}^2} - \exp\left(-\frac{1}{\hat{E}}\right) \hat{E} \hat{N}_e, \quad (16)$$

$$res_3(\hat{x}, \hat{t}) := \frac{\partial \hat{N}_i}{\partial \hat{t}} + \frac{\mu_i}{\mu_e} \left( \hat{E} \frac{\partial \hat{N}_i}{\partial \hat{x}} + \hat{N}_i \frac{\partial \hat{E}}{\partial \hat{x}} \right) - \frac{B D_i}{A \mu_e} \frac{\partial^2 \hat{N}_i}{\partial \hat{x}^2} - \exp\left(-\frac{1}{\hat{E}}\right) \hat{E} \hat{N}_e. \quad (17)$$

By using the non-dimensional form, all the variables are scaled to order  $O(1)$ , and the training processes of the PINNs with complex setting and parameters can be performed to finally predict the new data. Although there could be a way to weight the components of the loss function to mitigate the bias casted into the loss function due to this discrepancy across scales, this process would require a lot of guess-work and tuning.

### III. RESULTS AND DISCUSSION

By introducing PINNs, the Poisson equation, continuity equation, and drift-diffusion approximation equation can be solved without discretization, and the DC-driven discharge system can be fully described by the profiles of electron density, ion density, and electric fields. The comparison of the results from PINNs and those from the fluid simulations shows the effectiveness of the PINNs for solving the fluid model.

Figure 6 demonstrates the spatiotemporal distributions of electron density, ion density, and electric field predicted by PINNs, along with their  $L_2$  errors compared to fluid simulations. During the pre-discharge phase ( $t < 0.4 \mu\text{s}$ ), both electron and ion densities remain uniformly distributed below  $10^8 \text{ cm}^{-3}$  [Figs. 6(a) and 6(b)], while the electric field exhibits a quasi-linear spatial profile [Fig. 6(c)]. In this low-gradient regime, PINNs leverage their continuous function approximation capability to achieve prediction errors [ $L_2 < 0.001$ , Figs. 6(d) and 6(f)] comparable to numerical algorithm precision, confirming their robustness in modeling smooth physical processes. As the discharge transitions to the intense ionization phase ( $0.4 \mu\text{s} < t < 0.6 \mu\text{s}$ ), a striking electron density accumulation emerges in the anode region [Fig. 6(a)], reaching concentrations three orders of magnitude higher than local ion density and forming a characteristic anode sheath. Concurrently, the electric field peaks at  $14.7 \text{ kV}/\text{cm}$  [Fig. 6(c)]. Within the sheath, extreme gradients manifest: Electron density surges two orders of magnitude within  $0.5 \text{ mm}$  near the , while the electric field undergoes violent distortion over the same spatial scale. This multi-physics-coupled state challenges PINNs to simultaneously resolve millimeter-scale boundary-layer effects through mesh-free constraints and enforce conservation laws governed by PDEs via residual loss terms. The resulting nonlinear dynamics drive local prediction errors to their cycle maximum ( $L_2 \approx 0.01$ ), with error peaks precisely localized at the sheath's steepest gradient zones. Remarkably, the network still captures the space-charge-dominated field-particle coupling mechanism, as evidenced by error distribution patterns that synchronize with the sheath boundary motion in fluid

24 May 2025 06:45:25

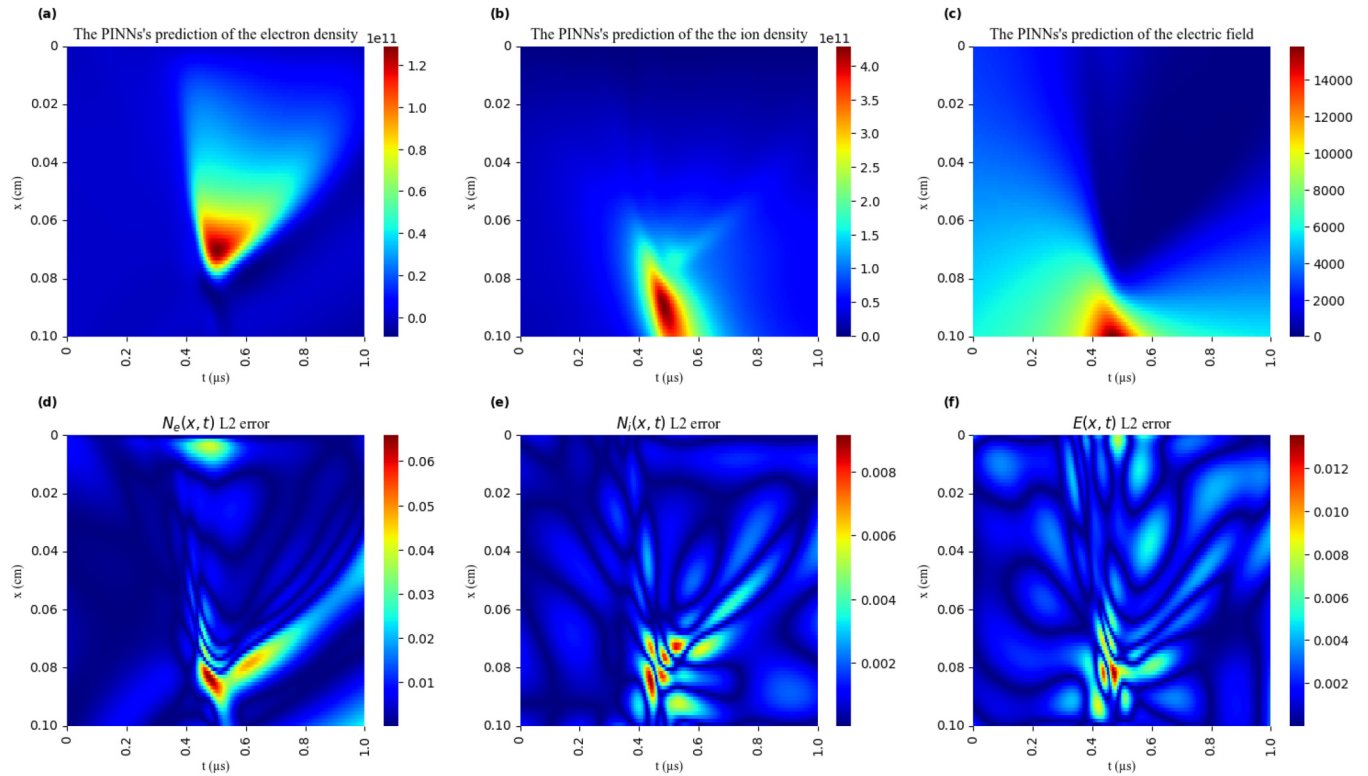


FIG. 6. PINNs-predicted electron density (a), ion density (b), and electric field strength (c), with  $L_2$  error relative (d)–(f) to fluid simulation results.

simulations [Figs. 6(d)–6(f)]. Following discharge extinction ( $t > 0.6 \mu\text{s}$ ), quasi-neutral particle distributions are restored [Figs. 6(a) and 6(b)], and the electric field regains gradual gradients [Fig. 6(c)]. Prediction errors rapidly decay to 0.001  $L_2$  levels, matching the pre-breakdown phase and validating PINNs' numerical stability throughout the entire discharge cycle.

Figure 7 presents the spatiotemporal distributions of electron density, ion density, and electric field predicted by the purely data-driven neural network (NN without  $\mathcal{L}_{\text{residual}}$ ). It is evident that the data-driven NN struggles to accurately resolve the spatial profiles of density and electric field due to the scarcity of high-precision training data. In contrast, the PINNs incorporating PDE residual

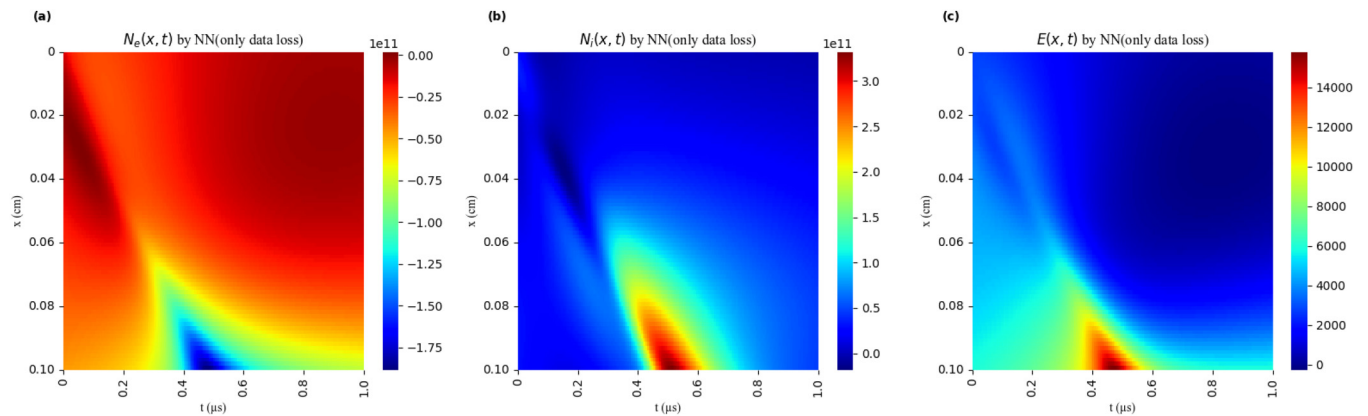
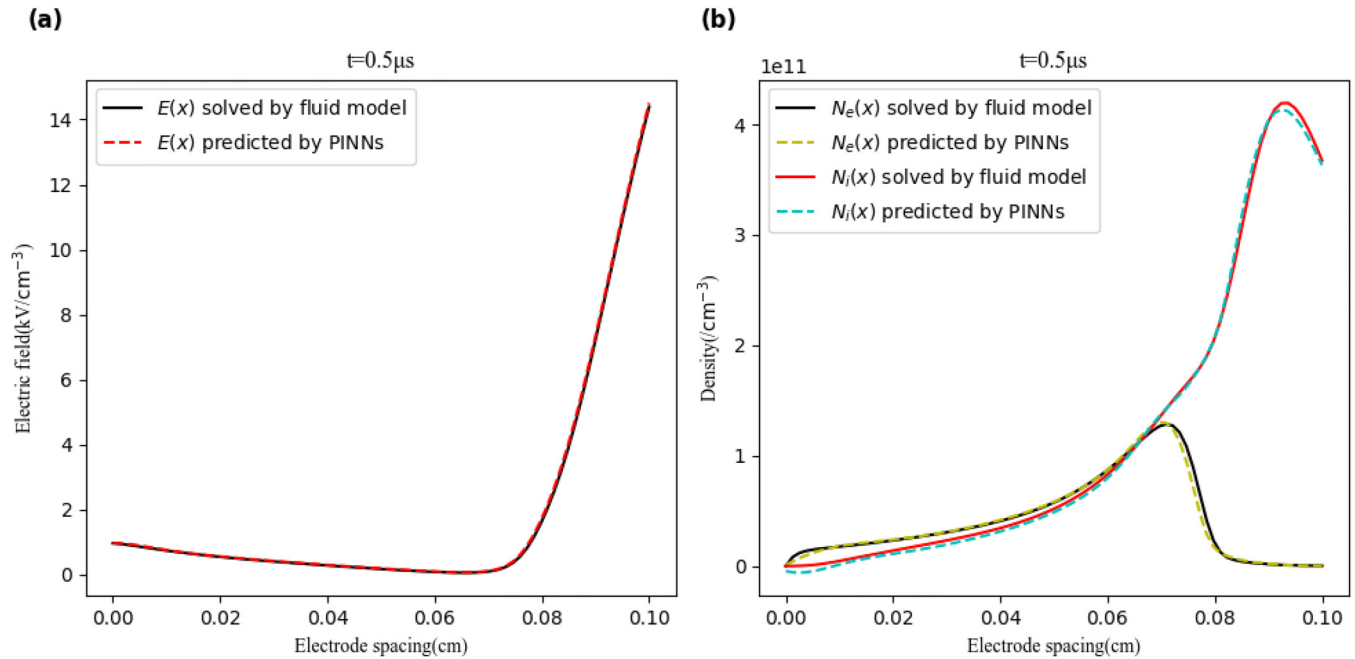


FIG. 7. PINNs-predicted electron density (a), ion density (b), and electric field strength (c) without PDE constraints.

24 May 2025 06:45:25



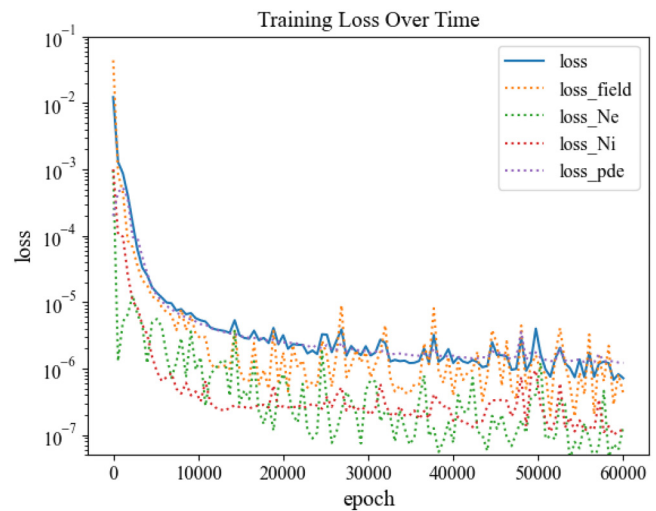


**FIG. 8.** Comparison of spatial distributions of electric field (a) and particle density (b) at the moment when discharge current gets peak value from fluid simulation and PINNs.

loss  $[\mathcal{L}_{residual}]$ , as shown in Figs. 6(a)–6(c)] embeds coupled drift-diffusion equations, continuity equations, and Poisson equations through deep physical encoding. This approach maintains strict physical consistency even in data-sparse regions ( $<1.5\%$  sensor coverage): Anode sheath sharpness  $L_2$  errors remain below 0.01, and more notably, the  $L_2$  error during the pre-discharge phase remains as low as 0.01. By enforcing physical constraints through the PDE residual terms, the PINNs ensure consistent and accurate predictions of electron density, ion density, and electric field intensity, maintaining alignment with the underlying conservation laws and multi-physics coupling mechanisms.

The PINNs' predictions for the spatial distributions of electron density, ion density, and electric field at the moment when the discharge current reaches its peak value are also compared with the simulation data in Fig. 8, which also have been discussed in Ref. 39, indicating a good agreement. The PINNs-predicted profiles closely match the simulation results, particularly in the sheath region where steep gradients dominate. At peak current (Fig. 8), the low-pressure DC discharge exhibits a distinct cathode drop region on the right electrode side, where PINNs accurately resolve the ion density gradient (reaching  $4 \times 10^{11}/cm^3$ ) surpassing electron density by three orders of magnitude, while capturing the electric field peak of 14.7 kV/cm. The enhanced electric field in the cathode sheath, precisely predicted by the PINNs through embedded drift-diffusion-Poisson constraints. Adjacent to the sheath (right side,  $x > 0.07$  cm), the field rapidly diminishes to maximum [Fig. 8(b)], coinciding with the electron density maximum [Fig. 8(a)], while the rest of the electrode spacing ( $x < 0.07$  cm) is identified as the positive column,

which is a large quasi-neutral plasma region where the electron density and ion density are almost equal with a relatively lower electric field. All features of glow-mode discharge can be reproduced by PINNs without traditional discretization. These predicted



**FIG. 9.** Loss function and test errors for electron density, ion density, and electric field during training.

24 May 2025 06:45:25

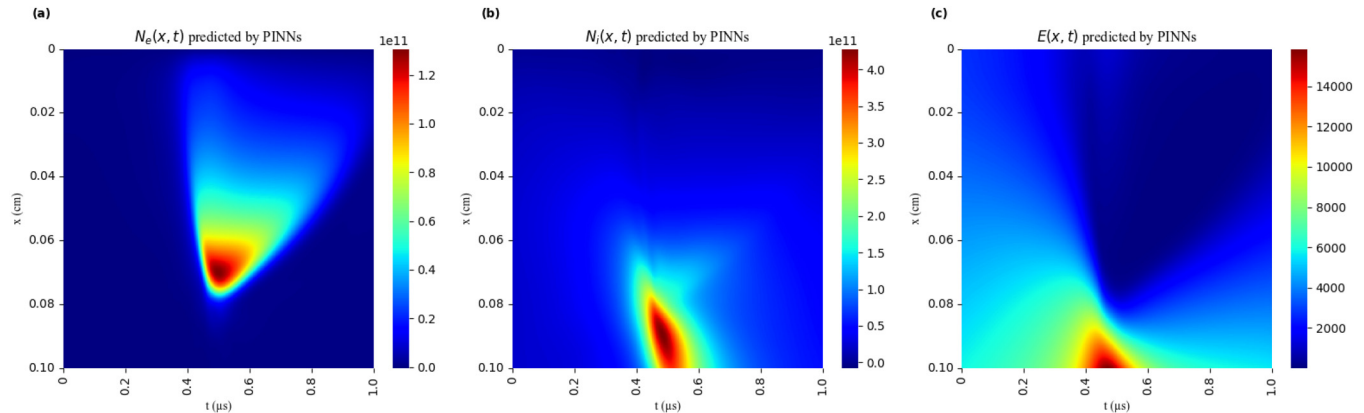


FIG. 10. Spatiotemporal distributions of electron density (a), ion density (b), and electric field (c) with a resolution of  $8000 \times 800$  from PINNs.

data from the PINNs clearly show a good consistency with the simulation results from the traditional discretization method and precisely capture the key dynamics behaviors of the low-pressure DC discharges, which demonstrates the effectiveness and accuracy of the PINNs, providing an alternative for numerical solution of fluid models for LTP with great advantages.

The training process of the PINNs involves monitoring the loss function and the test errors of various quantities as the training progresses. Figure 9 illustrates how the loss function and the test errors evolve over the course of training. Typically, the loss function decreases as the training proceeds, indicating that the model is improving its fit to the fluid equations and the given data. Similarly, the test errors for quantities such as electron density, ion density, and electric field should also decrease, reflecting the model's increasing accuracy in predicting these physical variations. Additionally, during the training process, the  $L_2$  test error can reach between  $10^{-4}$  and  $10^{-5}$ . This indicates a high level of accuracy in the predictions made by the PINNs model. However, it is clearly evident from the training process that the test error for electron density is significantly higher than the test errors for ion density and electric field strength. The reason for this may be that, despite our efforts to non-dimensionalize the equations to minimize the differences in magnitude between electron density, particle density, and electric field strength, the processed electron density remains much smaller than the ion density when the actual discharge is very weak. Small-scale data can be ignored or given lower weights during the training process of neural networks. This occurs because neural networks tend to focus more on larger-scale data when handling features of different magnitudes.

Figure 10 illustrates the capability of PINNs to output electron density, ion density, and electric field at any resolution. Figure 10 provides the spatiotemporal distribution of particle density and electric field on 800 spatial grid points and 8000 time steps. If desired, we can even output the spatiotemporal distribution of particle density and electric field at a resolution of 80 000 spatial grid points by 8000 time steps, or even higher. For traditional methods, achieving output at the same high resolution would result in a substantial increase in computational cost, often scaling much more

steeply with increases in resolution. This highlights a significant advantage of PINNs, which can efficiently generate high-resolution solutions with relatively less computational effort compared to conventional approaches.

#### IV. CONCLUSION

In this study, PINNs were applied to solve the fluid model, which describe a DC-driven discharge system, instead of the discretization method. The details of PINNs for fluid model were given to show how the PINNs to get the electron density, ion density, and electric field at any spatial-temporal resolution in the simulation. The equations in the fluid model were combined as the physical constraints for PINNs with data from experimental observation or numerical simulation; thus, the predicted results by PINNs are consistent with the physical laws. The training process of PINNs minimizes a composite loss function that includes both the residuals of the fluid model equations and the constraints from simulation data, which ensures that the model satisfies both physical laws and experimental observations. The computational results show that after training, the  $L_2$  relative error between the PINNs solutions and the outputs from traditional fluid simulations can be approximated with  $10^{-3}$  accuracy, which validates the feasibility and superiority of PINNs in numerical simulations of LTPs, offering new tools and methods for simulation and applications in plasma physics.

Compared to traditional grid-based methods (e.g., finite volume/element schemes), PINNs offer distinct advantages in solving fluid models for discharge plasmas. Mesh-free formulation eliminates grid generation challenges near complex geometries and dynamic phenomena like sheath layers; continuous spatiotemporal representation avoids interpolation errors during transient phases by inherently resolving smooth field transitions. While adaptive optimization through loss weighting was not explicitly implemented in this work, it represents a promising future direction to enhance accuracy in high-gradient regions (e.g., cathode sheath) without manual mesh refinement. We acknowledge that training efficiency currently lags behind mature discretization tools, but

24 May 2025 06:45:25

ongoing work on transfer learning and physics-guided initialization aims to bridge this gap. For LTPs, the complex physicochemical processes involved in multi-particle systems, multi-physics, and multi-scale coupling present a significant challenge for both the solution and optimization of PINNs. Furthermore, many choices during training, such as network architecture parameters, activation functions, data preprocessing, and the form of the equations, can significantly affect the training effects. Therefore, AI-based numerical simulation methods for LTPs require further optimization to address these challenges.

## ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 12375201).

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

## Author Contributions

**Wen-Kai Li:** Data curation (equal); Investigation (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Yuan-Tao Zhang:** Funding acquisition (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal).

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES

- <sup>1</sup>S. J. Yoon and J. Goo Lee, *Energy Fuels* **26**, 524 (2012).
- <sup>2</sup>S. Liu, D. Mei, Z. Shen, and X. Tu, *J. Phys. Chem. C* **118**, 10686 (2014).
- <sup>3</sup>N. Khadir, K. Khodja, and A. Belasri, *Plasma Sci. Technol.* **19**, 095502 (2017).
- <sup>4</sup>J. Yang, T. Li, C. Zhong, X. Guan, and C. Hu, *J. Electrochem. Soc.* **163**, E288 (2016).
- <sup>5</sup>X. Pei, D. Gidon, Y.-J. Yang, Z. Xiong, and D. B. Graves, *Chem. Eng. J.* **362**, 217 (2019).
- <sup>6</sup>V. Tu, J. Jeong, A. Schütze, S. Babayan, G. Ding, G. Selwyn, and R. Hicks, *J. Vac. Sci. Technol. A* **18**, 2799 (2000).
- <sup>7</sup>L. Zhu, W. Teng, H. Xu, Y. Liu, Q. Jiang, C. Wang, and Y. Qiu, *Surf. Coat. Technol.* **202**, 1966 (2008).
- <sup>8</sup>X. Lu, G. V. Naidis, M. Laroussi, S. Reuter, D. B. Graves, and K. Ostrikov, *Phys. Rep.* **630**, 1 (2016).
- <sup>9</sup>G. Lloyd, G. Friedman, S. Jafri, G. Schultz, A. Fridman, and K. Harding, *Plasma Processes Polym.* **7**, 194 (2010).
- <sup>10</sup>S. Lee, F. Iza, and J. K. Lee, *Phys. Plasmas* **13**, 057102 (2006).
- <sup>11</sup>Y. Hong, M. Yoon, F. Iza, G. Kim, and J. Lee, *J. Phys. D: Appl. Phys.* **41**, 245208 (2008).
- <sup>12</sup>R. V. Garimella, *Int. J. Numer. Methods Eng.* **55**, 451 (2002).
- <sup>13</sup>A. El Kacimi and O. Laghrouche, *Int. J. Numer. Methods Eng.* **84**, 330 (2010).
- <sup>14</sup>X.-C. Wang and Y.-T. Zhang, *J. Appl. Phys.* **133**, 143301 (2023).
- <sup>15</sup>Y.-T. Zhang, S.-H. Gao, and Y.-Y. Zhu, *J. Appl. Phys.* **133**, 053303 (2023).
- <sup>16</sup>Z.-B. Liu, X.-C. Wang, and Y.-T. Zhang, *IEEE Trans. Plasma Sci.* **51**, 1212 (2023).
- <sup>17</sup>Z.-N. Chai, X.-C. Wang, M. Yusupov, and Y.-T. Zhang, *Plasma Processes Polym.* **21**, 2300230 (2024).
- <sup>18</sup>R. Li, X. Wang, and Y. Zhang, *Appl. Sci.* **14**, 6855 (2024).
- <sup>19</sup>M. He, R. Bai, S. Tan, D. Liu, and Y. Zhang, *Plasma Processes Polym.* **21**, e2400020 (2024).
- <sup>20</sup>M. G. Dissanayake and N. Phan-Thien, *Commun. Numer. Methods Eng.* **10**, 195 (1994).
- <sup>21</sup>A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, *J. Mach. Learn. Res.* **18**, 1 (2018).
- <sup>22</sup>M. Raissi, P. Perdikaris, and G. E. Karniadakis, *J. Comput. Phys.* **378**, 686 (2019).
- <sup>23</sup>J. Sirignano and K. Spiliopoulos, *J. Comput. Phys.* **375**, 1339 (2018).
- <sup>24</sup>S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, *J. Sci. Comput.* **92**, 88 (2022).
- <sup>25</sup>M. Raissi, A. Yazdani, and G. E. Karniadakis, *Science* **367**, 1026 (2020).
- <sup>26</sup>S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, *Acta Mech. Sin.* **37**, 1727 (2021).
- <sup>27</sup>S. Xu, Z. Sun, R. Huang, D. Guo, G. Yang, and S. Ju, *Acta Mech. Sin.* **39**, 322302 (2023).
- <sup>28</sup>L. Hou, B. Zhu, and Y. Wang, *Acta Mech. Sin.* **39**, 322326 (2023).
- <sup>29</sup>S. A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, *Comput. Methods Appl. Mech. Eng.* **384**, 113959 (2021).
- <sup>30</sup>N. Zobeiry and K. D. Humfeld, *Eng. Appl. Artif. Intell.* **101**, 104232 (2021).
- <sup>31</sup>C. Cheng, H. Meng, Y.-Z. Li, and G.-T. Zhang, *Ocean Eng.* **240**, 109932 (2021).
- <sup>32</sup>Z. Fang and J. Zhan, *IEEE Access* **8**, 24506 (2019).
- <sup>33</sup>D. Pfau, J. S. Spencer, A. G. Matthews, and W. M. C. Foulkes, *Phys. Rev. Res.* **2**, 033429 (2020).
- <sup>34</sup>X. Yuan and L. L. Raja, *IEEE Trans. Plasma Sci.* **31**, 495 (2003).
- <sup>35</sup>Y. T. Zhang, D. Z. Wang, and Y. H. Wang, *Phys. Plasmas* **12**, 103508 (2005).
- <sup>36</sup>Y. T. Zhang, D. Z. Wang, and M. G. Kong, *J. Appl. Phys.* **100**, 063304 (2006).
- <sup>37</sup>C. Strümpel, Y. A. Astrov, and H. Purwins, *Phys. Rev. E: Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* **62**, 4889 (2000).
- <sup>38</sup>C. Strümpel, H.-G. Purwins, and Y. A. Astrov, *Phys. Rev. E* **63**, 026409 (2001).
- <sup>39</sup>D. D. vsijavcic, U. Ebert, and I. Rafatov, *Phys. Rev. E: Stat. Nonlinear Soft Matter Phys.* **71**, 066402 (2005).
- <sup>40</sup>N. Balcon, G. Hagelaar, and J.-P. Boeuf, *IEEE Trans. Plasma Sci.* **36**, 2782 (2008).
- <sup>41</sup>J. He, J. Hu, D. Liu, and Y.-T. Zhang, *Plasma Sources Sci. Technol.* **22**, 035008 (2013).
- <sup>42</sup>A. Kulikovskiy, *J. Comput. Phys.* **119**, 149 (1995).
- <sup>43</sup>C. Yuan, C. Yesil, J. Yao, Z. Zhou, and I. Rafatov, *Plasma Sources Sci. Technol.* **29**, 065009 (2020).
- <sup>44</sup>M. Raissi, P. Perdikaris, and G. E. Karniadakis, *arXiv:1711.10561* (2017).
- <sup>45</sup>A. M. Tartakovsky, C. O. Marrero, P. Perdikaris, G. D. Tartakovsky, and D. Barajas-Solano, *arXiv:1808.03398* (2018).
- <sup>46</sup>M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, *J. Fluid Mech.* **861**, 119 (2019).
- <sup>47</sup>S. Wang, Y. Teng, and P. Perdikaris, *arXiv:2001.04536* (2020).
- <sup>48</sup>S. Wang, X. Yu, and P. Perdikaris, *J. Comput. Phys.* **449**, 110768 (2022).
- <sup>49</sup>M. Stein, *Technometrics* **29**, 143 (1987).
- <sup>50</sup>D. D. Šijačić and U. Ebert, *Phys. Rev. E* **66**, 066410 (2002).