

# THE MODEL REDUCTION OF THE VLASOV–POISSON–FOKKER–PLANCK SYSTEM TO THE POISSON–NERNST–PLANCK SYSTEM VIA THE DEEP NEURAL NETWORK APPROACH

JAE YONG LEE<sup>1</sup>, JIN WOO JANG<sup>1,2</sup> AND HYUNG JU HWANG<sup>1,\*</sup>

**Abstract.** The model reduction of a mesoscopic kinetic dynamics to a macroscopic continuum dynamics has been one of the fundamental questions in mathematical physics since Hilbert’s time. In this paper, we consider a diagram of the diffusion limit from the Vlasov–Poisson–Fokker–Planck (VPFP) system on a bounded interval with the specular reflection boundary condition to the Poisson–Nernst–Planck (PNP) system with the no-flux boundary condition. We provide a Deep Learning algorithm to simulate the VPFP system and the PNP system by computing the time-asymptotic behaviors of the solution and the physical quantities. We analyze the convergence of the neural network solution of the VPFP system to that of the PNP system *via* the Asymptotic-Preserving (AP) scheme. Also, we provide several theoretical evidence that the Deep Neural Network (DNN) solutions to the VPFP and the PNP systems converge to the *a priori* classical solutions of each system if the total loss function vanishes.

**Mathematics Subject Classification.** 68T20, 35Q84, 35B40, 82C40.

Received October 15, 2020. Accepted July 27, 2021.

## 1. INTRODUCTION

### 1.1. Motivation: a diagram of diffusion limit

The description of physical dynamics in various scales is one of the main questions of interest in the mathematical modeling of complex systems. In kinetic theory, the description of the evolution of gases has been explained *via* the statistical approach on the probabilistic distribution functions on the mesoscopic level, whereas the fluid theory describes the dynamics on the macroscopic level. Each of these interpretations and the asymptotic expansions of the mesoscopic equations to the macroscopic equations have been crucial issues.

The aim of this paper is to establish the commutation of the following diagram of diffusion limit, which provides the reduction of the kinetic equation (the Vlasov–Poisson–Fokker–Planck system) to the fluid equation (the Poisson–Nernst–Planck system) as the perturbation parameter  $\varepsilon$  tends to zero:

---

*Keywords and phrases.* Vlasov–Poisson–Fokker–Planck system, Poisson–Nernst–Planck system, diffusion limit, artificial neural network, asymptotic-preserving scheme.

<sup>1</sup> Department of Mathematics, Pohang University of Science and Technology (POSTECH), Pohang 37673, Republic of Korea.

<sup>2</sup> Institute for Applied Mathematics, University of Bonn, Endenicher Allee 60, 53115 Bonn, Germany.

\*Corresponding author: [hjhwang@postech.ac.kr](mailto:hjhwang@postech.ac.kr)

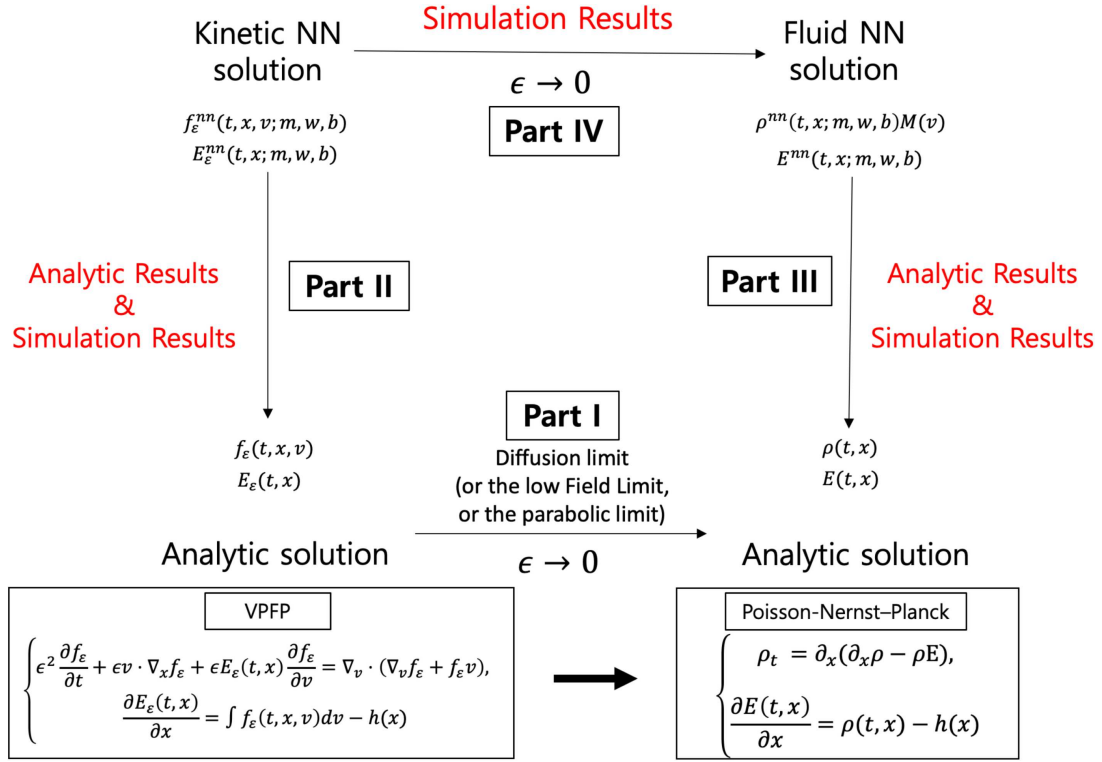


FIGURE 1. The diagram of diffusion limit.

We refer to a theoretical result from [83] to obtain the bottom side (Part I) of the diagram. For the left-hand side (Part II), the right-hand side (Part III), and the upper side (Part IV) of the diagram, we use a Deep Learning method using the Deep Neural Network (DNN) solution to approximate the solutions to the kinetic equation and the fluid equation. We provide large time behaviors and the steady-states of several physical moments of these DNN solutions to show an agreement with the theoretical results. Also, we provide theoretical evidence on the relationship between DNN solutions and the analytic solutions for the left and right-hand sides of the diagram.

There are many numerical studies to simulate an initial-boundary value problem for the kinetic and fluid equations. Especially, it is computationally challenging in a numerical scheme to automatically capture the limit for the asymptotic expansions on the small parameter (*e.g.* the parameter  $\epsilon$  tends to zero as the upper side of Fig. 1). Many numerical schemes have been developed to overcome this challenge. These schemes are the so-called Asymptotic-Preserving (AP) schemes, which have been firstly introduced by Jin [54]. The key idea is to develop a numerical scheme to preserve the asymptotic limits from a mesoscopic to a macroscopic model in the fixed discrete setting.

A Deep Learning method has achieved remarkable success in various areas. Many studies have recently been introduced for learning partial differential equations (PDEs) using the Deep Learning method. These studies approximate the solutions of PDEs using a neural network architecture as a function approximator based on a universal approximation theorem [21]. Along with many numerical methods, this Deep Learning approach has been proposed as a new way to simulate PDE problems.

In this paper, we provide a Deep Learning algorithm to simulate each side of Figure 1. In addition, we prove that the Deep Neural Network solutions converge to the analytic solutions. The simulation results of the Deep

Learning support the model reduction of the Vlasov–Poisson–Fokker–Planck solution to the Poisson–Nernst–Planck solution.

## 1.2. Main results, difficulties and our strategy

In this paper, we establish the commutation of the diagram in Figure 1 on the diffusion limit from the Vlasov–Poisson–Fokker–Planck (VPFP) system (1.1) in a bounded interval with the specular reflection boundary condition (1.8) to the Poisson–Nernst–Planck (PNP) system (1.4) with the no-flux boundary condition (1.10). We provide a Deep Learning Algorithm 1 to simulate the VPFP system and the PNP system. The Deep Learning is a new possible approach for solving partial differential equations (PDEs) with many advantages. The Deep Learning method is a mesh-free approximation method, while the traditional methods in numerical analysis such as the Finite Difference schemes are influenced by the mesh. In this work, we randomly sample the grid points from a given domain in every epoch to train the Deep Neural Network. By randomly sampling each domain, we avoid the need of generating a mesh. We use the PyTorch library and the Adam optimizer for our Deep Learning algorithm.

One of the main advantages of the Deep Learning is that the algorithm can be implemented easily and intuitively by the computation of the gradient of the loss function with the chain rule, so-called the back-propagation algorithm. Namely, this means that it is easy to put the PDE information into the algorithm by adding terms to the loss function. In this regard, we propose the specific loss functions suitable for the VPFP system (4.7) in Section 4.1 and for the PNP system (5.7) in Section 5.1. Also, we use two neural network architectures at the same time to solve the Vlasov–Fokker–Planck system as in Figure 3 and the Nernst–Planck system as in Figure 4, which are coupled with the Poisson equation. In this way, there is a considerable advantage compared to traditional numerical methods, for which we needed to find a numerical scheme for each equation. We also use the Softplus activation function ( $\bar{\sigma}(x) = \ln(1 + e^x)$ ) for the output layer in the neural network structure. Since the Softplus function has the output in the scale of  $(0, +\infty)$ , it makes the neural network output  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  positive.

Meanwhile, there are some weaknesses of the Deep Learning approach. First of all, it is difficult to evaluate the accuracy of Deep Learning while the numerical methods have well-known error bounds. Also, it is difficult to show that the Deep Learning algorithms find the exact global minimum in the optimization aspect. Since the Deep Learning is the gradient-based approach, we cannot guarantee that we could find the global minimum of the loss function, even in our case using the *Adam* optimizer. Instead, in order to show that our DNN solutions well-approximate the solutions of the VPFP system and the solutions of the PNP system, we provide numerical simulations to demonstrate that the neural network solutions satisfy the given theoretical predictions on the asymptotic behaviors of the VPFP system in Section 4.3 and of the PNP system in Section 5.3. We analyze our DNN solutions *via* computing the steady-states for the solutions and *via* computing the physical quantities of the total mass, the kinetic energy, the entropy, the electric energy and the free energy, and their steady-states.

We also provide the theoretical supports for the convergence of the DNN solution to the *a priori* classical solution in two theorems. More precisely, for the VPFP system in Part II (Sect. 4.2), we claim in Theorem 4.5 that we can find a sequence of approximated neural network solutions that reduce the loss function (4.7). Also, we prove an additional theorem (Thm. 4.7) which states that the neural network solution converges to an analytic solution as we minimize the loss function (4.7). In the proof, we use the transformation for the Vlasov–Poisson–Fokker–Planck system motivated by Carrillo [17]. Similarly, we provide Theorem 5.2 for the PNP system in Part III (Sect. 5.2).

In particular, it is hard to capture the asymptotic limit using the fixed numerical discretization in numerical schemes. In this work, we propose a newly devised technique, “Grid Reuse” method, which boosts the Deep Learning by adding the grid points which make the integrand in the loss function the largest in every epoch. The “Grid Reuse” method makes it possible to approximate the neural network solution of the VPFP system with a small Knudsen number  $\varepsilon$  without changing the number of grids sampled. We provide the numerical simulation for the trend of the diffusion limit from the DNN solution of the VPFP system to the DNN solution of the PNP system in Part IV (Sect. 6). To the best of authors’ knowledge, this is the first attempt to use the Deep

Learning method as an AP scheme to see the trend of the diffusion limit of the Vlasov–Poisson–Fokker–Planck system.

The main distinctive of this paper compared to the numerical methods is the use of the neural network approach as a function approximator for the VPFP system, the PNP system, and the AP scheme as the Knudsen number  $\varepsilon$  goes to 0. In this paper, our main goal is to complete the commutation in Figure 1 of the neural network version similar to Figure 2 of the numerical analysis version.

### 1.3. The Vlasov–Poisson–Fokker–Planck equation

In order to study the diffusion limit of the Vlasov–Poisson–Fokker–Planck system in a bounded interval  $\Omega \stackrel{\text{def}}{=} (-1, 1)$ , we need to rescale the VPFP system with the Knudsen number  $\varepsilon$ . The small parameter  $\varepsilon$  represents the ratio of the mean free path of the particles to the typical macroscopic length scale of the particle flow. We are interested in the scaling of the system using the change of variables  $t' = \varepsilon^2 t$  and  $x' = \varepsilon x$ ; see Section 4 of [77] and Section 1 of [83]. With these variables, the VPFP system in a bounded interval  $\Omega = (-1, 1)$  can be written in the dimensionless form as follows:

$$\begin{aligned} \varepsilon^2 \partial_t f_\varepsilon + \varepsilon v \partial_x f_\varepsilon + \varepsilon E_\varepsilon \partial_v f_\varepsilon &= \partial_v (v f_\varepsilon + \partial_v f_\varepsilon), & t \in [0, T], \ x \in \Omega, \ v \in \mathbb{R}, \\ f_\varepsilon(0, x, v) &= f_0(x, v), \\ \partial_x E_\varepsilon &= \int_{\mathbb{R}} f_\varepsilon dv - h(x), & x \in \Omega, \\ E_\varepsilon(0, -1) &= 0, \\ E_\varepsilon(t, x) &= 0, & x \in \partial\Omega, \end{aligned} \tag{1.1}$$

where  $f_\varepsilon = f_\varepsilon(t, x, v) \geq 0$  is the distribution of particles in  $(t, x, v) \in (0, T) \times \Omega \times \mathbb{R}$ , and  $E_\varepsilon = E_\varepsilon(t, x)$  is the self-consistent electric force, where  $\Phi_\varepsilon(t, x)$  is the internal potential of the system with the equation

$$E_\varepsilon(t, x) = -\partial_x \Phi_\varepsilon(t, x).$$

The function  $h(x)$ , which is on the right-hand side of the Poisson equation, stands for the presence of a background charge (*e.g.* ions). We assume the global neutrality condition

$$\int_{\Omega} \int_{\mathbb{R}} f_\varepsilon dv dx - \int_{\Omega} h(x) dx = 0. \tag{1.2}$$

Note that the equations (1.1)<sub>2</sub>, (1.1)<sub>3</sub>, and (1.1)<sub>4</sub> will together imply

$$0 = E_\varepsilon(0, x) = \int_{-1}^x \partial_x E_\varepsilon(0, y) dy = \int_{-1}^x \int_{\mathbb{R}} f_0(y, v) dv dy - \int_{-1}^x h(y) dy. \tag{1.3}$$

The existence and the uniqueness of the VPFP system have been well-studied. Victory and O'Dwyer [78] showed the existence of the classical solutions to the VPFP system in two dimension. Rein and Weckler [73] and Bouchut [11] showed the existence of global solutions to the three-dimensional VPFP system in the whole space. We refer to [17] for the global weak solutions of VPFP system in a bounded domain with absorbing and reflection type boundary conditions. The large time asymptotic solutions to the Vlasov–Fokker–Planck equation has been studied first in [26, 29] in the case that the particles occupy the whole space. They prove that the distribution function  $f(t, x, v)$  tends to a Maxwellian function. This result has been extended by Bouchut and Dolbeault [12] under the more general assumption on the external potential. Also, we refer to [18, 27] in whole space domain. In the case of initial-boundary value problem for the VPFP system, Bonilla *et al.* [10] studied the large time asymptotic behaviors of the solutions with the reflection type boundary condition. The authors in [38] considered the global well-posedness of the nonlinear Fokker–Planck equation with the specular reflection boundary. Also, the well-posedness and regularity for different boundary conditions to the kinetic Fokker–Planck equation has been studied in [44–51].

### 1.4. The Poisson–Nernst–Planck equation

One of the macroscopic models to describe the distribution and the transport of ionic species is the Poisson–Nernst–Planck (PNP) system, where it is also often called the Drift–Diffusion–Poisson (DDP) equation. The PNP system consists of the Nernst–Planck equation that describes the drift and diffusion of ions and the Poisson equation that describes the effect of the self-consistent electric field. In this paper, we consider the following the 1-dimensional Poisson–Nernst–Planck (PNP) system in a bounded interval  $\Omega = (-1, 1)$ :

$$\begin{aligned} \partial_t \rho &= \partial_x (\partial_x \rho - \rho E), & t &\in [0, T], \ x \in \Omega, \\ \rho(0, x) &= \rho_0(x), \\ \partial_x E &= \rho(t, x) - h(x), & x &\in \Omega, \\ E(0, -1) &= 0, \\ E(t, x) &= 0, & x &\in \partial\Omega. \end{aligned} \quad (1.4)$$

Here,  $\rho = \rho(t, x)$  stands for density of particles and  $E(t, x)$  is the self-consistent electric force with the relation  $E(t, x) = -\partial_x \Phi(t, x)$ , similarly to the VPFP system. Then (1.4)<sub>2</sub>, (1.4)<sub>3</sub> and (1.4)<sub>4</sub> will together imply

$$0 = E(0, x) = \int_{-1}^x \partial_x E(0, y) dy = \int_{-1}^x \rho_0(y) dy - \int_{-1}^x h(y) dy. \quad (1.5)$$

We also assume the neutrality condition for the background charge  $h(x)$  as follows:

$$\int_{\Omega} \rho(t, x) dx - \int_{\Omega} h(x) dx = 0. \quad (1.6)$$

The PNP system has a number of applications in many fields, such as electrical engineering, electrokinetics, electrochemistry and biophysics. Therefore, the analytical study of the PNP system has also a long history in the various context. An initial-boundary value problem for a system on the transport of mobile carriers in a semiconductor is studied by Gajewski and Groger [35]. The existence and large time behavior of the PNP equation is studied in [8]. Also, the convergence rate of solutions to the PNP system is studied in [4, 7]. We refer to a review paper [79] for a recent development of generalized PNP systems.

### 1.5. Boundary conditions

#### 1.5.1. Phase boundary and the specular reflection for the VPFP system

Throughout this paper, we will denote the phase boundary in  $\Omega \times \mathbb{R}$  as  $\gamma \stackrel{\text{def}}{=} \partial\Omega \times \mathbb{R}$ . Additionally we split this boundary into an outgoing boundary  $\gamma_+$ , an incoming boundary  $\gamma_-$ , and a singular boundary  $\gamma_0$  for grazing velocities, defined as

$$\begin{aligned} \gamma_+ &\stackrel{\text{def}}{=} \{(x, v) \in \partial\Omega \times \mathbb{R} : n_x \cdot v > 0\}, \\ \gamma_- &\stackrel{\text{def}}{=} \{(x, v) \in \partial\Omega \times \mathbb{R} : n_x \cdot v < 0\}, \\ \gamma_0 &\stackrel{\text{def}}{=} \{(x, v) \in \partial\Omega \times \mathbb{R} : n_x \cdot v = 0\}, \end{aligned} \quad (1.7)$$

where  $n_x$  is the outward normal vector. We define the boundary integration for  $f(x, v)$ ,  $(x, v) \in \partial\Omega \times \mathbb{R}$ ,

$$\int_{\gamma_{\pm}} f d\gamma = \int_{\gamma_{\pm}} f(x, v) |n_x \cdot v| dS_x dv,$$

where  $dS_x$  is the standard surface measure on  $\partial\Omega$  and denote

$$\int_{\gamma} f d\gamma = \int_{\gamma_+} f(x, v) d\gamma - \int_{\gamma_-} f(x, v) d\gamma.$$

We also define the  $L^2(\gamma)$  norm with respect to the measure  $|n_x \cdot v| dS_x dv$ ,

$$\|f\|_\gamma^2 \stackrel{\text{def}}{=} \|f\|_{\gamma_+}^2 + \|f\|_{\gamma_-}^2.$$

In terms of  $f$ , we formulate the specular reflection boundary condition as

$$f(t, x, v)|_{\gamma_-} = f(t, x, R(x)v), \quad (1.8)$$

for all  $x \in \partial\Omega$ , where

$$R(x)v \stackrel{\text{def}}{=} v - 2n_x(n_x \cdot v).$$

One of the well-known *a priori* conservation laws for the Vlasov–Poisson–Fokker–Planck system (1.1) for the specular boundary conditions is the conservation of mass. It means that the total mass, “Mass”, of distribution  $f_\varepsilon(t, x, v)$  is preserved for any time as follows:

$$\text{Mass}(t) \stackrel{\text{def}}{=} \int_\Omega \int_{\mathbb{R}} f_\varepsilon(t, x, v) dv dx \equiv \int_\Omega \int_{\mathbb{R}} f_0(x, v), \quad (1.9)$$

which means  $\frac{d}{dt} \text{Mass}(t) = 0$ .

### 1.5.2. No-flux Boundary Condition for the PNP system

The PNP system is usually posed in a bounded domain with some boundary condition. In this paper, we use the no-flux boundary condition for the PNP equation as follows:

$$(\partial_x \rho - \rho E) \cdot n_x = 0, \quad x \in \partial\Omega. \quad (1.10)$$

This condition is one of the natural boundary conditions for a macroscopic model to explain the diffusion of ions under the effect of potential. With the Dirichlet boundary condition (1.4)<sub>5</sub> for the Poisson equation, it reduces the following boundary conditions: This boundary condition implies that the system (1.1) has the conservation of charges/ions. We can check this property by integrating (1.4)<sub>1</sub> with respect to  $x$  over the whole domain  $[-1, 1]$  as follows:

$$\partial_x \rho(t, x) = \rho(t, x)E(t, x) = 0, \quad x \in \partial\Omega.$$

$$\partial_t \left( \int_\Omega \rho dx \right) = [\partial_x \rho - \rho E]_{x=-1}^1 = 0.$$

This implies that the conservation of total density, that is,

$$\text{Mass}_\rho(t) \stackrel{\text{def}}{=} \int_\Omega \rho(t, x) dx \equiv \int_\Omega \rho_0(x) dx, \quad (1.11)$$

which means  $\frac{d}{dt} \text{Mass}_\rho(t) = 0$ .

## 1.6. The equilibrium state and the macroscopic quantities

### 1.6.1. The equilibrium state and the macroscopic quantities for the VPFP system

It is well-known that the VPFP system has a local equilibrium solution. Bonilla *et al.* [10] introduced the form of the steady-state of the VPFP system in bounded domains with the reflection boundary condition on  $f(t, x, v)$  and the Dirichlet boundary condition for the potential  $\Phi(t, x)$  without a background charge. They remark that they can prove a result analogously with the Neumann boundary conditions instead of the Dirichlet boundary conditions. In this regard, the VPFP system (1.1), which has the background charge as in (1.1)<sub>3</sub>, has the equilibrium state as follows:

$$f_{\varepsilon, \infty}(x, v) = C_{vpfp} M(v) e^{-\Phi_\infty(x)}, \quad (1.12)$$

where  $M(v) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}$  is the normalized Maxwellian and  $\Phi_\infty(x)$  is a weak solution of the Poisson–Boltzmann (PB) equation:

$$\begin{aligned} -\Delta \Phi_\infty(x) &= C_{vpfp} e^{-\Phi_\infty(x)} - h(x), \\ \partial_x \Phi_\infty(x) &= 0 \quad \text{on } x \in \partial\Omega, \end{aligned} \quad (1.13)$$

and

$$C_{vpfp} = \|f_0(\cdot, \cdot)\|_{L^1_{x,v}} \left( \int_{\Omega} e^{-\Phi_\infty(x)} dx \right)^{-1}.$$

The system (1.13) is called the Poisson–Boltzmann (PB) equation. By Theorem 1 of [16], the system (1.13) has a solution  $\Phi_\infty(x)$  unique up to an additive constant. If we assume the background charge  $h(x)$  as constant, we can check that the system (1.13) has a solution  $\Phi_\infty(x) = 0$  by the global neutrality condition (1.2). Therefore, the VPFP system (1.1) has the global equilibrium state  $(f_{\varepsilon,\infty}, E_{\varepsilon,\infty})$  as

$$f_{\varepsilon,\infty}(x, v) = \frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|} M(v), \quad E_{\varepsilon,\infty}(x) = -\partial_x \Phi_\infty(x) = 0. \quad (1.14)$$

We expect that the neural network solutions of the VPFP system reach the steady-state (1.14) (see simulation Sect. 4.3).

The Lyapunov functional  $\eta(t)$  for the VPFP system (1.1) is defined by the relative entropy of the solution  $f_\varepsilon(t, x, v)$  with respect to a non-normalized stationary distribution  $\hat{f}$ . As explained in [10], we define the Lyapunov functional  $\eta(t)$  as

$$\eta(t) \stackrel{\text{def}}{=} \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon \log \left( \frac{f_\varepsilon}{\hat{f}_\varepsilon} \right) dv dx, \quad (1.15)$$

where  $\hat{f}_\varepsilon$  is defined as

$$\hat{f}_\varepsilon \stackrel{\text{def}}{=} \exp \left\{ -\frac{v^2}{2} - \Phi_\varepsilon(t, x) + \frac{1}{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}} \frac{1}{2} \left( \int_{\Omega} E_\varepsilon(t, x)^2 dx \right) \right\}. \quad (1.16)$$

Then,  $\eta(t)$  can be reduced as

$$\begin{aligned} \eta_\varepsilon(t) &= \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon \log \left( \frac{f_\varepsilon}{\hat{f}_\varepsilon} \right) dv dx \\ &= \int_{\Omega} \int_{\mathbb{R}} \left[ f_\varepsilon \log f_\varepsilon + \frac{1}{2} f_\varepsilon v^2 + f_\varepsilon \Phi_\varepsilon - \frac{f_\varepsilon}{2\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}} \left( \int_{\Omega} E_\varepsilon(t, x)^2 dx \right) \right] dv dx \\ &= \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon \log f_\varepsilon dv dx + \frac{1}{2} \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon v^2 dv dx + \int_{\Omega} \left( \int_{\mathbb{R}} f_\varepsilon dv \right) \Phi_\varepsilon dx \\ &\quad - \frac{1}{2} \int_{\Omega} E_\varepsilon(t, x)^2 dx. \end{aligned}$$

If we assume the background charge  $h(x)$  as constant and assume the zero-mean constraint for the  $\Phi_\varepsilon$  (see Rem. 2.1), then we have

$$\begin{aligned} \int_{\Omega} \left( \int_{\mathbb{R}} f_\varepsilon dv \right) \Phi_\varepsilon dx &= \int_{\Omega} (-\partial_{xx} \Phi_\varepsilon + h(x)) \Phi_\varepsilon dx = - \int_{\Omega} \partial_{xx} \Phi_\varepsilon \Phi_\varepsilon dx + \int_{\Omega} h(x) \Phi_\varepsilon dx \\ &= - \underbrace{[(\partial_x \Phi_\varepsilon) \Phi_\varepsilon]_{-1}^1}_{=0} + \int_{\Omega} (\partial_x \Phi_\varepsilon)^2 dx + \underbrace{\int_{\Omega} h(x) \Phi_\varepsilon dx}_{=0} = \int_{\Omega} E_\varepsilon(t, x)^2 dx. \end{aligned}$$

Therefore, it yields that

$$\eta_\varepsilon(t) = \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon \log f_\varepsilon dv dx + \frac{1}{2} \int_{\Omega} \int_{\mathbb{R}} f_\varepsilon v^2 dv dx + \frac{1}{2} \int_{\Omega} E_\varepsilon(t, x)^2 dx \stackrel{\text{def}}{=} -\text{Ent}(t) + \text{KE}(t) + \text{EE}(t),$$

where the entropy of the system “Ent”, the total kinetic energy “KE”, and the electric potential energy “EE” of the system are defined as

$$\text{Ent}(t) \stackrel{\text{def}}{=} - \int_{\Omega \times \mathbb{R}} f_\varepsilon \log f_\varepsilon dx dv, \quad (1.17)$$

$$\text{KE}(t) \stackrel{\text{def}}{=} \frac{1}{2} \int_{\Omega \times \mathbb{R}} |v|^2 f_\varepsilon dx dv, \quad (1.18)$$

and

$$\text{EE}(t) \stackrel{\text{def}}{=} \frac{1}{2} \int_{\Omega} |E_\varepsilon|^2 dx. \quad (1.19)$$

The Lyapunov functional is also called the free energy defined as

$$\text{FE}(t) \stackrel{\text{def}}{=} -\text{Ent}(t) + \text{KE}(t) + \text{EE}(t). \quad (1.20)$$

Since the Lyapunov functional satisfies  $\frac{d}{dt} \eta(t) \leq 0$ , we expect that the free energy (1.20) is a non-increasing function (see Sect. 4.3).

#### 1.6.2. The equilibrium state and the free energy for the PNP system

The steady state of the PNP system (1.4) satisfies

$$\partial_x(\partial_x \rho_\infty - \rho_\infty E_\infty) = 0,$$

from the equations (1.4)<sub>1</sub>. It is reduced to

$$\frac{\partial_x \rho_\infty}{\rho_\infty} - E_\infty = \text{Constant}.$$

Therefore, we have the following steady state:

$$\rho_\infty(x) = C_{pnp} \exp \left( \int_{\Omega} E_\infty(x) dx \right), \quad (1.21)$$

where  $E_\infty(x)$  is a solution of the Poisson–Boltzmann (PB) equation

$$\begin{aligned} \partial_x E_\infty(x) &= C_{pnp} \exp \left( \int_{\Omega} E_\infty(x) dx \right) - h(x), \\ E_\infty(x) &= 0 \quad \text{on} \quad x \in \partial\Omega, \end{aligned} \quad (1.22)$$

with some constant  $C_{pnp}$ . We can express the constant  $C_{pnp}$  using the total density (1.11) as

$$C_{pnp} = \left( \int_{[-1,1]} \rho_0(x) dx \right) \left( \int_{[-1,1]} \exp \left( \int_{\Omega} E_\infty(x) dx \right) dx \right)^{-1}. \quad (1.23)$$

The PB equation (1.22) has a solution  $E_\infty(x) = 0$  similar to the PB equation (1.13) in the VPFP system. Therefore, the PNP system (1.4) has the steady state as follows:

$$\rho_\infty(x) = C_{pnp}, \quad E_\infty(x) = 0, \quad (1.24)$$



with the constant  $C_{pnp}$  which is defined in (1.23). We expect that the neural network solutions of the PNP system reach the steady-state (1.24) (see simulation Sect. 5.3).

Also, the free energy  $\text{FE}_\rho(t)$  of the PNP system (1.4) is defined as follows (similar to [33, 53, 63]):

$$\text{FE}_\rho(t) \stackrel{\text{def}}{=} \int_{\Omega} \left( \rho(t, x) \log \rho(t, x) + \frac{1}{2} E(t, x)^2 \right) dx, \quad (1.25)$$

which has both the entropic part and the interaction part. The first term  $\rho(t, x) \log \rho(t, x)$  on the right-hand side is the entropy related to the Brownian motion of each particles, and the second term  $\frac{1}{2} E(t, x)^2$  is the electric potential energy of the particles.

Under the specific boundary conditions (1.10) and (1.4)<sub>5</sub>, the PNP system has the following relation:

$$\begin{aligned} \int_{\Omega} \Phi(\rho(t, x) - h(x)) dx &= \int_{\Omega} \Phi \partial_x E dx = - \int_{\Omega} \Phi \partial_{xx} \Phi dx \\ &= -[\Phi \partial_x \Phi]_{-1}^1 + \int_{\Omega} (\partial_x \Phi)^2 dx = \int_{\Omega} (\partial_x \Phi)^2 dx = \int_{\Omega} E(t, x)^2 dx, \end{aligned}$$

by multiplying  $\Phi(t, x)$  onto (1.4)<sub>3</sub>, integrating it over the domain  $\Omega$  and using the integration by parts with respect to  $x$ . Therefore, the free energy can be rewritten as

$$\text{FE}_\rho(t) = \int_{\Omega} \left( \rho(t, x) \log \rho(t, x) + \frac{1}{2} (\rho(t, x) - h(x)) \Phi \right) dx.$$

By taking the time derivative of the free energy  $\text{FE}_\rho(t)$ , we can derive

$$\begin{aligned} \frac{d}{dt} \text{FE}_\rho(t) &= \int_{\Omega} (\rho_t \log \rho + \rho_t) dx + \frac{1}{2} \int_{\Omega} ((\rho - h(x)) \Phi_t + \rho_t \Phi) dx \\ &= \int_{\Omega} \rho_t (\log \rho + 1 + \Phi) dx + \frac{1}{2} \int_{\Omega} ((\rho - h(x)) \Phi_t - \rho_t \Phi) dx. \end{aligned}$$

Then, we have

$$\begin{aligned} \frac{d}{dt} \text{FE}_\rho(t) &= \int_{\Omega} \partial_x (\rho_x - \rho E) (\log \rho + 1 + \Phi) dx + \frac{1}{2} \int_{\Omega} (-\Phi_{xx} \Phi_t + \Phi_{txx} \Phi) dx \\ &= - \int_{\Omega} (\rho_x - \rho E) \left( \frac{\rho_x}{\rho} + \Phi_x \right) dx + \frac{1}{2} \int_{\Omega} \partial_x (\Phi \Phi_{tx} - \Phi_t \Phi_x) dx \\ &= - \int_{\Omega} \frac{1}{\rho} (\rho_x - \rho E)^2 dx + \frac{1}{2} [\Phi \Phi_{tx} - \Phi_t \Phi_x]_{-1}^1 = - \int_{\Omega} \frac{1}{\rho} (\rho_x - \rho E)^2 dx. \end{aligned}$$

Therefore, the PNP system (1.4) satisfies the following free energy dissipation law:

$$\frac{d}{dt} \text{FE}_\rho(t) = - \int_{\Omega} \frac{1}{\rho} (\rho_x - \rho E)^2 dx \leq 0. \quad (1.26)$$

We expect that the free energy (1.25) of the PNP system is a non-increasing function (see simulation Sect. 5.3).

### 1.7. Mathematical results on the diffusion limit

In this section, we introduce past results on the diffusion limit of the VPFP system. There are two scalings of the VPFP. The first one is the diffusion limit (or the parabolic limit, or the low field limit), and the second one is the drift limit (or the hyperbolic limit, or the high field limit). In this paper, we consider the first one on the diffusion limit of the VPFP system only. The diffusion limit has been extensively investigated in many works.

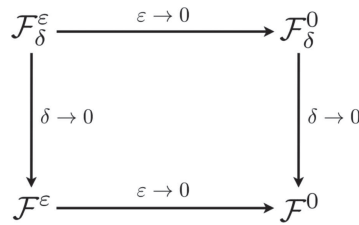


FIGURE 2. Illustration of AP schemes.

Poupaud [70] considers the diffusion limit of the semiconductor Boltzmann equation. The diffusion limit for the VPFP system with a given background was considered by [36, 71] in the two-dimensional case. And later, El Ghani and Masmoudi [30] extend these results to higher dimensional cases in the renormalized sense. The case of multiple-species dynamics is also considered in [41, 83]. A recent paper [83] of Wu *et al.* treats the diffusion limit of the VPFP system in a bounded domain with reflection boundary conditions. We used the results of this paper to show the bottom side of Figure 1. Also, there are many works that deal with the drift limit as in [3, 9, 37, 66].

### 1.8. Existing numerical methods and an Asymptotic Preserving scheme

In this section, we introduce a brief history of the numerical methods to approximate the solutions of the VPFP system and the PNP equation. We also introduce the numerical studies concerning the asymptotic expansions on the small parameters, the so-called Asymptotic Preserving (AP) scheme.

There are many numerical studies to solve the VPFP system and related systems. There is a wide range of literature on numerical analysis for the Fokker–Planck (FP) equation including the finite difference method [19, 74], and its conservative type scheme [6, 13–15, 23]. The particle method [1, 40] is an effective method for the stochastic properties of the Fokker–Planck operator. Also, Wollman and Ozizmir [80–82] provided the deterministic particle method for the VPFP systems in one and two-dimensional cases. Another approach is the spectral method to solve the Fokker–Planck equation. In [68], they develop a new spectral method based on a Fourier spectral approximation for the Boltzmann equation. Filbet and Pareschi [32] extended the method to the nonhomogenous case. The review paper [25] contains the latest references on numerical methods for collisional kinetic equations.

Also, a lot of efforts have been made to the numerical methods for the PNP system. Many of the existing methods have been constructed for both one-dimensional and higher dimensional cases in various chemical and biological contexts. We refer to some recent studies for solving time-dependent PNP systems. Solkalski *et al.* [76] proposed the finite difference scheme for analyzing liquid junction and ion-selective membrane potentials. Hyon *et al.* [53] provided another finite element method with the back-Euler method for the modified PNP system. It is considered to be difficult for numerical schemes to provide the physical properties of the PNP system; namely the nonnegativity principle, the mass conservation, and the free energy dissipation. Regarding these difficulties, Liu and Wang [62] developed a finite difference method for the PNP system. They focus on the development of a free energy satisfying numerical method for the PNP system. They also provided the discontinuous Galerkin scheme for a one-dimensional case in [63]. The implicit methods with the trapezoidal rule and backward differentiation are presented in [33].

Regarding the numerical methods to capture the relation between two regimes, Shi Jin [54] first introduced the numerical scheme that preserves the asymptotic limits from the mesoscopic to the macroscopic models for transport in diffusive regimes – the asymptotic-preserving (AP) scheme. The commutative diagram of Figure 2 (taken from Figure 1 of [55]) illustrates the AP scheme of [55]. As explained in [55],  $\mathcal{F}^\varepsilon$  is a mesoscopic model which depends on parameter  $\varepsilon$  that characterizes the small scale.  $\mathcal{F}_\delta^\varepsilon$  is a discretization of  $\mathcal{F}^\varepsilon$  with parameter  $\delta$  that is related to numerical discretization (such as mesh size and/or time step). As  $\varepsilon$  goes to zero, the mesoscopic

model  $\mathcal{F}^\varepsilon$  is approximated by a macroscopic model  $\mathcal{F}^0$ . Then, the scheme  $\mathcal{F}_\delta^\varepsilon$  is called AP if the asymptotic limit of  $\mathcal{F}_\delta^\varepsilon$  as  $\varepsilon \rightarrow 0$  with  $\delta$  fixed, denoted by  $\mathcal{F}_\delta^0$ , is a good approximation of  $\mathcal{F}^0$ .

The AP schemes are developed for various equations. Especially, there are many studies that deal with the AP schemes for the kinetic equations with the Euler regime. Filbet and Jin [31] developed a penalization method to overcome the Boltzmann integral, which is a fully nonlinear collision operator. Jin and Yan [57] generalized their idea to the nonhomogeneous Fokker–Planck–Landau equation. Dimarco and Pareschi [24] introduce an exponential Runge–Kutta method for kinetic equations. The AP schemes for the high field limit of the VPFP system are considered in [20, 56]. In [20], they also developed the AP scheme based on a micro-macro decomposition for the diffusion limit of the Vlasov–Poisson–BGK model. We refer to the recent surveys by Jin [55], Degond [22] and Pareschi and Russo [67].

Given the existing numerical methods in the literature, the main distinctive of this paper is the use of the neural network approach as a solver for these important problems. We used the neural network method as a function approximator for the VPFP system, PNP system, and the AP scheme as the parameter  $\varepsilon$  goes to 0. The aim of this paper is to complete Figure 1 of neural network version similar to Figure 2 of the numerical analysis version.

### 1.9. Neural network and an approach to solve a PDE

Neural network has also drawn attention in the machine learning community. It has been used for various fields such as natural language process, image recognition, speech recognition, and others. Deep Learning, which uses a deep stack of neural network layer called a Deep Neural Network (DNN), is effectively applied in these areas. The neural network architecture is introduced in [65] for the first time. There are theoretical results to justify the use of neural networks in these applications such as [21, 34, 42, 43]. The key theorem to these results is the universal approximation theorem. The universal approximation theorem states that an arbitrary real-valued function can be well-approximated by a feed-forward neural network. Later, Li [61] showed that the neural network with one hidden layer could approximate not only a target function but also its higher partial derivatives on a compact set.

Then Deep Learning as a PDE solver has also been studied; Lagaris *et al.* [59, 60] suggested the use of neural networks to solve ODEs and PDEs. Recently, Raissi *et al.* [72] introduced physics informed neural networks. They design data-driven algorithms for two main problems: data-driven solutions and data-driven discovery of partial differential equations. The data-driven method to solve the high-dimensional PDEs with a DNN is proposed in [75]. The second problem, called the forward-inverse problem, is also considered in [58] with a theoretical analysis of the convergence of the DNN solutions to the classical solutions. In [2], they present a method for approximating the solution of PDEs using an adaptive collocation strategy. Also, Han *et al.* [39] deal with the uniformly accurate moment system using the kinetic equation as an example.

Hwang *et al.* [52] introduce the Deep Neural Network solutions to the kinetic Fokker–Planck equation in a bounded interval under the varied types of the physical boundary conditions. They observed the asymptotic behaviors of the DNN solutions to verify an agreement with theoretical results. They also provide the theoretical proofs on the relationship between the DNN solutions and the *a priori* analytic solutions. Our paper is motivated by several ideas in [52]. We expand their ideas to a more general VPFP system and its diffusion limit.

### 1.10. Outline of the paper

Each of the four sides of Figure 1 consists of four parts (Part I, II, III, and IV). In Section 2 (Part I), we show that the solutions of the VPFP system converge to the solutions of the PNP system as the Knudsen number  $\varepsilon$  tends to zero, which corresponds to the bottom side of Figure 1. To this end, we use the theoretical result from the paper [83]. In Section 3, we will introduce in detail our Deep Learning method to approximate the solution of the VPFP system and the solution of the PNP system, which is used for the numerical simulations in Part II, III, and IV. Part II will include the detailed descriptions on the DNN architectures for each system (Sect. 3.2), the definition of grid points (Sect. 3.3), and a “Grid Reuse” method that is a newly devised tool in

the paper (Sect. 3.4) to capture the dynamics under a small Knudsen number  $\varepsilon$ . In Section 4 (Part II), we will introduce the DNN approximated solutions to the VPFP system (1.1), which corresponds to the left-hand side of Figure 1. We will provide the suitable loss functions (4.7) to approximate the VPFP system using the Deep Learning in Section 4.1. We will prove the convergence of the DNN solution to an analytic solution of the VPFP system as the loss function vanishes in Section 4.2. We will also provide the numerical simulations that show the asymptotic behaviors of macroscopic quantities and the pointwise values of the DNN solution to the VPFP system in Section 4.3. In Section 5 (Part III), we will introduce the DNN approximated solutions to the PNP system (1.4), which corresponds to the right-hand side of Figure 1. The contents would be analogous to those in Section 4. In Section 6 (Part IV), we will provide several numerical simulations to see the trend of the diffusion limit from the VPFP system to the PNP system, which corresponds to the upper side of Figure 1. We will analyze the convergence (2.4) and (2.5) using the DNN solutions of the VPFP system by varying the Knudsen number from 1 to 0.05 *via* the Asymptotic-Perserving (AP) scheme. Finally, in Section 7, we will summarize our methods and the results.

## 2. PART I. ON CONVERGENCE OF THE VPFP SOLUTION TO THE PNP SOLUTION

In this section, we introduce the convergence of solutions of the VPFP system to a solution of the PNP system from the recent paper ([83], Thm. 2.1). Wu *et al.* [83] prove that the VPFP system (1.1) with the Maxwellian reflection boundary condition converges to the PNP system (1.4) as  $\varepsilon$  tends to zero for the multi-species model case. To be more specific, they consider the renormalized solution  $(f_{\varepsilon,i}, \Phi_\varepsilon)$  of a rescaled  $N$ -species VPFP system ( $i = 1, 2, \dots, N$ ) in a bounded interval  $\Omega \subset \mathbb{R}^d$  using the scaled parameters as

$$\begin{aligned} \partial_t f_{\varepsilon,i} + \frac{1}{\varepsilon} v \cdot \nabla_x f_{\varepsilon,i} - \frac{\kappa_i z_i}{\varepsilon} \nabla_x \Phi_\varepsilon \cdot \nabla_v f_{\varepsilon,i} &= \frac{\zeta_i}{\varepsilon^2} \nabla_v \cdot (v f_{\varepsilon,i} + \kappa_i \nabla_v f_{\varepsilon,i}), \\ -\varpi \Delta_x \Phi_\varepsilon &= \sum_{i=1}^N z_i \int_{\mathbb{R}^d} f_{\varepsilon,i}(t, x, v) dv - h(x), \end{aligned} \quad (2.1)$$

with initial condition, reflection boundary condition (especially, Maxwellian boundary condition) for the distribution function  $f_{\varepsilon,i}$  and zero-outward electric field condition (Neumann boundary condition) for the electric potential  $\Phi_\varepsilon$ . They show that the solution  $(f_{\varepsilon,i}, \Phi_\varepsilon)$  converges to  $(\rho_i(t, x) M_i(v), \Phi(t, x))$ , where  $(\rho_i, \Phi)$  is a weak solution of the PNP system in a bounded interval  $\Omega$  as

$$\begin{aligned} \partial_t \rho_i + \nabla_x \cdot \left( \overbrace{-\frac{1}{\zeta_i} \nabla_x \rho_i - \frac{z_i}{\zeta_i} \rho_i \nabla_x \Phi}^{J_i(t,x) \stackrel{\text{def}}{=}} \right) &= 0, \\ -\varpi \Delta_x \Phi &= \sum_{i=1}^N z_i \rho_i - h(x), \end{aligned} \quad (2.2)$$

with the initial-boundary conditions given as follows

$$\begin{aligned} J_i \cdot n &= 0, \text{ on } \partial\Omega, \\ \nabla_x \Phi \cdot n &= 0, \text{ on } \partial\Omega, \end{aligned} \quad (2.3)$$

as  $\varepsilon$  tends to zero ( $M_i = M_i(v)$ 's are the normalized Maxwellians for each species).

Using this result, we derive our specific system (1.1) with the boundary condition. Firstly, we specify the 1-dimension bounded domain  $\Omega = (-1, 1) \subset \mathbb{R}$  on the spatial domain and  $\mathbb{R}$  on the velocity domain. Also, we consider the single-species case with  $N = 1$ . This case is reasonable in plasma physics when the relatively huge ions are supposed to be static in the background. In this case, we denote the distribution function  $f_{\varepsilon,i}$  as  $f_\varepsilon$  for the VPFP system, since  $i = 1$ . We also choose the classical specular reflection boundary condition for the

$f_\varepsilon(t, x, v)$  instead of the Maxwellian boundary condition used in [83]. We use the Dirichlet boundary condition for the electric force  $E_\varepsilon(t, x)$  which is the same as the Neumann boundary condition for the electric field  $\Phi_\varepsilon(t, x)$  assumed in [83]. The boundary conditions (2.3) imply the Neumann condition ( $\nabla_x \rho_i \cdot n = 0$  on  $\partial\Omega$ ) for the density function and the Dirichlet condition ( $E = \nabla_x \Phi \cdot n = 0$  on  $\partial\Omega$ ) for the electric force  $E(t, x)$ . Additionally, we set all the parameters to be 1 in the systems (2.1) and (2.2) except the Knudsen number  $\varepsilon$  to take the limit.

Then, the solution  $f_\varepsilon$  (corresponding to the  $f_{\varepsilon, i=1}$ ) and the solution  $E_\varepsilon = \partial_x \Phi_\varepsilon$  to the VPFP system (1.1) with the specular boundary condition (1.8) satisfy the following convergence:

$$f_\varepsilon(t, x, v) \rightarrow \rho(t, x)M(v) \quad \text{in } L^1(0, T; L^1(\Omega \times \mathbb{R})), \quad (2.4)$$

where  $M(v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}$  and

$$E_\varepsilon(t, x) \rightarrow E(t, x) \quad \text{in } L^2(0, T; L^p(\Omega)), \quad 1 \leq p < 2 \quad (2.5)$$

as the Knudsen number  $\varepsilon$  tends to zero, where the density  $\rho$  (corresponding to the  $\rho_{i=1}$ ) and the solution  $E$  satisfy the system (1.4) with the no-flux boundary condition (1.10). In Part IV (Sect. 6) of this paper, we provide the corresponding numerical simulations which show the trend of the convergence (2.4) and (2.5).

**Remark 2.1.** In [83], they prove the diffusion limit with two assumption for  $\Phi_\varepsilon$  on the Poisson equation; the global neutrality condition and the zero-mean constraint. The global neutrality condition is the same as the condition (1.2) we assumed. They also assume the zero-mean constraint as follows:

$$\int_{\Omega} \Phi_\varepsilon dx = 0 \quad (2.6)$$

his constraint is necessary to uniquely determine the solution  $\Phi_\varepsilon$ . However, we are interested in the solution  $E_\varepsilon(t, x) = -\partial_x \Phi_\varepsilon(t, x)$  which is the partial of  $\Phi(t, x)$  instead of  $\Phi_\varepsilon(t, x)$  in this paper. Without loss of generality, we can assume (2.6) to apply the diffusion limit theorem from [83].

### 3. SIMULATION METHODOLOGY: THE DEEP LEARNING APPROACH

In this section, we introduce our deep learning method to solve the Cauchy problem to the Vlasov–Poisson–Fokker–Planck system (1.1) and the Poisson–Nernst–Planck (PNP) system (1.4).

#### 3.1. A Deep Learning approach for solving partial differential equation

A Deep Learning algorithm can be described in terms of a non-linear function approximation method using a Deep Neural Network (DNN). A Deep Neural Network consists of a sequence of multiple layers. Each layers has several neurons, which receive the neuron activation from the pre-layer as input. The neurons implement the weighted sum of the input and apply an activation function in order to transform the output to a non-linear one. The output is transmitted to neurons in the post-layer. We assume that a DNN has  $L$  layers; it has an input layer,  $L - 1$  hidden layers and an output layer ( $L$ -th layer). Similarly to the explanation of [52], we denote the relation between the  $l$ -th layer and the  $(l + 1)$ -th layer ( $l = 1, 2, \dots, L - 1$ ) as

$$z_j^{(l+1)} = \sum_{i=1}^{m_l} w_{ji}^{(l+1)} \bar{\sigma}_l(z_i^l) + b_j^{(l+1)},$$

where  $m = (m_0, m_1, m_2, \dots, m_{L-1})$ ,  $w = \left\{ w_{ji}^{(k)} \right\}_{i,j,k=1}^{m_{k-1}, m_k, L}$ ,  $b = \left\{ b_j^{(k)} \right\}_{j=1, k=1}^{m_k, L}$ , and

- $z_i^l$ : the  $i$ -th neuron in the  $l$ -th layer
- $\bar{\sigma}_l$ : the activation function in the  $l$ -th layer

- $w_{ji}^{(l+1)}$ : the weight between the  $i$ -th neuron in the  $l$ -th layer and the  $j$ -th neuron in the  $(l+1)$ -th layer
- $b_j^{(l+1)}$ : the bias of the  $j$ -th neuron in the  $(l+1)$ -th layer
- $m_l$ : the number of neurons in the  $l$ -th layer.

Note that the relation between the input layer and the first hidden layer is expressed as follows:

$$z_j^1 = \sum_{i=1}^3 w_{ji}^1 z_i^0 + b_j^1,$$

where  $(z_1^0, z_2^0, z_3^0) = (t, x, v)$ .

The deep learning algorithm learns the complex nonlinear mapping by adapting these weights  $w_{ji}^{(l+1)}$  and biases  $b_j^{(l+1)}$  to make the output of Deep Neural Network similar to the target function, in our case, the solution of the VPFP and PNP system. The Deep learning uses the back-propagation learning algorithm, which applies the chain rule to calculate the influence of each weight and each bias to reduce a pre-defined cost function, which is called “loss function” in the Deep Learning. Then, the algorithm uses the gradient method to update the weights and biases.

To approximate a solution of PDEs using the deep learning algorithm, we need an appropriate loss function with respect to the PDE system. For example, suppose we consider the following parabolic PDE:

$$\begin{aligned} \frac{\partial u}{\partial t}(t, x) + \mathcal{L}u(t, x) &= 0, & (t, x) \text{ in } [0, T] \times \Omega, \\ u(0, x) &= u_0(x), & x \text{ in } \Omega, \\ \mathcal{B}u(t, x) &= q(x), & (t, x) \text{ in } [0, T] \times \partial\Omega, \end{aligned}$$

where  $\mathcal{L}$  is a differential operator and  $\mathcal{B}$  is the boundary operator with known functions  $u_0(x)$  and  $q(x)$ . In many papers (e.g. [72, 75]), they approximate the solution  $u(t, x)$  using the DNN output  $u^{nn}(t, x)$  with the loss function as

$$\begin{aligned} \text{Loss}(u^{nn}) &= \left\| \frac{\partial u^{nn}}{\partial t}(t, x) + \mathcal{L}u^{nn}(t, x) \right\|_{L^2([0, T] \times \Omega)}^2 \\ &\quad + \|u^{nn}(0, x) - u_0(x)\|_{L^2(\Omega)}^2 + \|\mathcal{B}u^{nn}(t, x) - q(t, x)\|_{L^2([0, T] \times \partial\Omega)}^2. \end{aligned} \quad (3.1)$$

The proposed loss function is an intuitive one to approximate the solution of PDE. In our case, we propose slightly different loss functions for each system. We define the loss function for the VPFP system in Section 4.1 (Part II) and for the PNP system in Section 5.1 (Part III). We propose the loss functions based on our theoretical evidence. In each section, we prove that the DNN output to the VPFP system and PNP system converges to a *priori* classical solution to each system if the proposed loss function goes to zero. The details are precisely described in Part II and Part III.

### 3.2. Our Deep Learning algorithm and the architecture

We take two different neural network structures which share the same inputs to approximate the coupled nonlinear equations. Each DNN has four hidden layers and each layer has 3(or 2)-100-100-100-1 neurons. For the VPFP system, the two Deep Neural Networks are used to approximate the solutions,  $f$  and  $E$ , respectively. The neural network structure is precisely shown in Figure 3.

We denote the approximated solution as  $(f_\varepsilon^{nn}(t, x, v; m, w, b), E_\varepsilon^{nn}(t, x; m, w, b))$ , which consists of the output of each DNN. The two outputs  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  and  $E_\varepsilon^{nn}(t, x; m, w, b)$  are used to calculate the pre-defined loss function. Then, we use a gradient descent algorithm to update the weights and biases of our model’s parameters by iteratively moving in the direction of reducing the loss function. In this work, we use the Adam (Adaptive

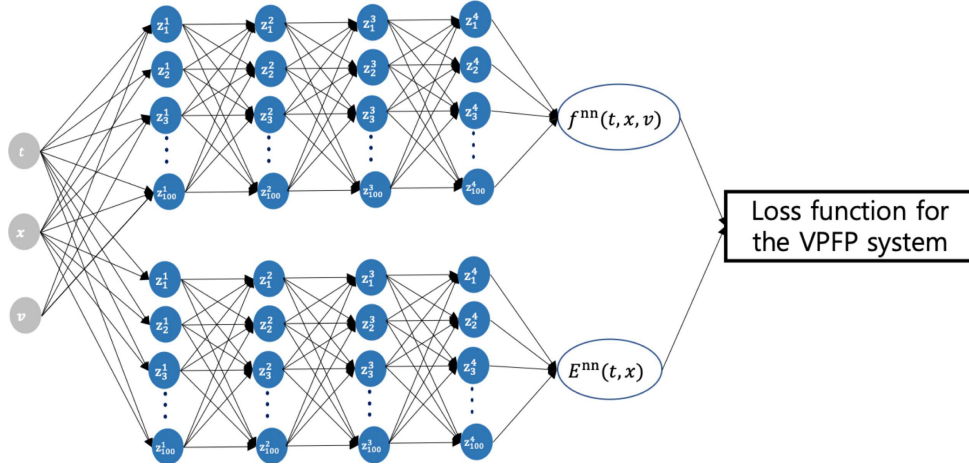


FIGURE 3. The DNN structure for the VPFP system.

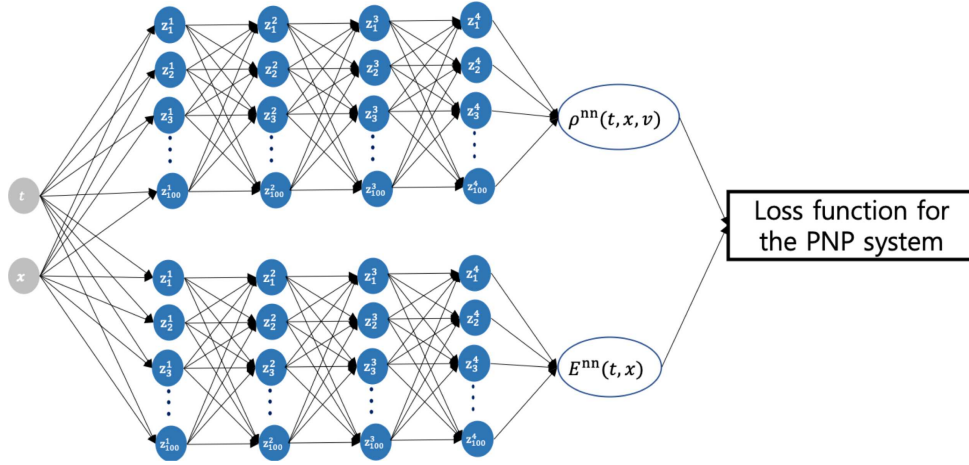


FIGURE 4. The DNN structure for the PNP system.

Moment Estimation) optimizer, an efficient variant of the stochastic gradient descent algorithm which is widely used in deep learning applications due to quick convergence in training.

Similarly, we use the two Deep Neural Networks to approximate the solutions  $(\rho, E)$  for the PNP system as in Figure 4. We denote the approximated solution as  $(\rho^{nn}(t, x; m, w, b), E^{nn}(t, x; m, w, b))$ .

For the four hidden layers in each DNN, we use the hyper-tangent activation function  $(\bar{\sigma}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}})$ , which is the common activation function in Deep Learning literature. While the choice of the activation function for the hidden layers is quite clear, the choice of an activation function for the output layer depends on the purposes. We use the Softplus activation function  $(\bar{\sigma}(x) = \ln(1 + e^x))$  only for the output  $f_\varepsilon^{nn}(t, x, v; m, w, b)$ . It is one of the main issues to preserve the positivity of the output when the numerical scheme is constructed. Since the Softplus function has outputs in scale of  $(0, +\infty)$ , we easily apply the positivity constraint for the output  $f_\varepsilon^{nn}(t, x, v; m, w, b)$ .

We use the PyTorch library for deep learning. It is one of the most standard deep learning frameworks due to its simplicity and ease of use. We also use the Adam optimizer in PyTorch library with the Learning rate



scheduling, which adjusts the learning rate based on the number of epochs. Regarding the loss function, we need the derivation and integration of the output with respect to the variables  $t$ ,  $x$  and  $v$ . To approximate the derivatives of the neural network output with respect to the input variables, we use the Autograd package in PyTorch library. It provides Automatic Differentiation (AD), which is one of the powerful techniques in scientific computing. The AD is different from the usual differentiation methods, such as numerical differentiation or the symbolic differentiation. We refer to the survey papers [5, 69] for more details. Also, we use the trapezoidal rule from the PyTorch library to approximate the integration. The specific loss functions we defined for the VPFP system and the PNP system are explained in Part II (VPFP) and Part III (PNP).

### 3.3. Training data: grid points

To approximate the solutions to the VPFP system and the PNP system *via* the Deep Learning algorithm, we make the grid points for each variable domain as inputs in the neural networks. We need three-dimensional time-space-velocity grid for the probability density  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  in VPFP system and two-dimensional time-space grid for the density  $\rho^{nn}(t, x)$  in PNP system and the force field  $E_\varepsilon^{nn}(t, x; m, w, b)$  to the VPFP system and  $E^{nn}(t, x; m, w, b)$  to the PNP system. We choose the time interval  $[0, T]$  as  $[0, 5]$  only for the VPFP system with  $\varepsilon = 1$  and  $[0, 1]$  with the smaller Knudsen number  $\varepsilon$ , which is enough to see the steady-state of both the VPFP system and the PNP system. Also, we truncate the momentum space for the  $v$  variable as  $V \stackrel{\text{def}}{=} [-10, 10]$  for training and assume that  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  is 0 if  $|v| > 10$ . Note that we sample the grid points for the each variables  $t$ ,  $x$  and  $v$  randomly for each iteration. Compared to the grid created by dividing the domain uniformly, this sampling-based approach has the effect of selecting infinite grids in the each domain. More precisely, the grid points for training  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  are chosen randomly as follows:

$$\{(t_i, x_j, v_k)\}_{i,j,k} \in [0, T] \times \Omega \times V \quad (3.2)$$

for the governing equation,

$$\{(t = 0, x_j, v_k)\}_{j,k} \in \Omega \times V \quad (3.3)$$

for the initial condition and

$$\{(t_i, x = -1 \text{ or } 1, v_k)\}_{i,k} \in [0, T] \times V \quad (3.4)$$

for the boundary condition with  $T = 1$  or  $T = 5$ ,  $\Omega = [-1, 1]$  and  $V = [-10, 10]$ . In every epoch, we sample the grid points for the time  $t$  as  $\{t_i\}_{i=1}^{10}$ , for the position  $x$  as  $\{x_j\}_{j=1}^{10}$ , and for the velocity  $v$  as  $\{v_k\}_{k=1}^{1000}$ . We use a larger number of velocity grids than the time and position grids to approximate the integration with respect to the velocity in the VPFP system (1.1)<sub>3</sub>. We can choose the grid points similarly for  $\rho^{nn}(t, x)$ ,  $E_\varepsilon^{nn}(t, x)$ , and  $E^{nn}(t, x)$ .

### 3.4. “Grid Reuse” method to capture the small Knudsen number

In Part IV, we provide the numerical simulations when the Knudsen number  $\varepsilon$  is small. It is hard to capture the asymptotic limit with the fixed numerical discretization in numerical schemes.

To overcome this challenge, we propose a newly devised technique in this paper; we call it “Grid Reuse” method. The Deep Neural Network is trained to minimize the sum of loss functions at randomly sampled grid points in every epoch, as explained in (3.2)–(3.4). The idea of our “Grid Reuse” method is that we add more top- $k$  grid points of these randomly sampled grid points to use for training in the next epoch. Here the top- $k$  grid points  $\{t_{\alpha_i}, x_{\beta_i}, v_{\gamma_i}\}_{i=1}^k$  are being selected such that the integrand is being the largest before we take the integration of the error with respect to the variables  $t$ ,  $x$  and  $v$ . Namely, we choose the top- $k$  grid points that make the largest values of the integrand in the loss function (4.1). The “Grid Reuse” method helps to solve the time dependency, which is one of the main difficulties on capturing the diffusion limit. Also, we note that we only save grid points of 2-tuple  $(t, x)$  as in Algorithm 1 though the top- $k$  grid points consist of 3-tuple  $(t, x, v)$ . This is because we need to calculate the integration term  $\int_V f_\varepsilon^{nn}(t, x, v; m, w, b) dv$  for the loss function (4.2) when we reuse the top- $k$  grid points. Therefore, we only catch the temporal and spatial grid points  $(t, x)$  where



the integrand in the loss function has the largest value. We then make the three-dimensional time-space-velocity grid with a randomly sampled velocity grid in  $V = [-10, 10]$ .

The “Grid Reuse” method is inspired by the Residual-based adaptive refinement (RAR) method in [64] and the adaptive collocation method in [2]. The technique of these methods and our method are similar to the adaptive mesh refinement method in numerical analysis.

### 3.5. Summary of Deep Learning algorithm

Finally, we summarize our Deep Learning algorithm for the VPFP system as follows:

---

**Algorithm 1** Deep Learning algorithm for the VPFP system

---

```

1: for number of epochs do
2:   Sampling data:
3:     Sample  $m$  samples  $t_1, t_2, \dots, t_m$  from  $[0, 1]$  (or  $[0, 5]$ ).
4:     Sample  $n$  samples  $x_1, x_2, \dots, x_n$  from  $[-1, 1]$ .
5:     Sample  $p$  samples  $v_1, v_2, \dots, v_p$  from  $[-10, 10]$ .
6:     Make a pair the samples to set the training data as (3.2)–(3.4).
7:     Add new top- $k$  training data paired with the velocity samples.
8:   Evaluate the loss function:
9:     Approximate the derivative of the DNN output (Autograd).
10:    Approximate the integration of the DNN output (Trapezoidal rule).
11:    Evaluate the loss function for the VPFP system (4.7).
12:   Updating parameters:
13:     Update neural network parameters using the Adam optimizer:

$$w \leftarrow w^{\text{new}},$$


$$b \leftarrow b^{\text{new}},$$

14:     in the direction of minimizing the pre-defined loss function.
15:   Grid Reuse technique:
16:     Choose top- $k$  grid points  $\{t_{\alpha_i}, x_{\beta_i}, v_{\gamma_i}\}_{i=1}^k$  which make the integrand in the loss function (4.1) the largest.
17:     Save top- $k$  grid points  $\{t_{\alpha_i}, x_{\beta_i}\}_{i=1}^k$ .
18: end for

```

---

We also apply a similar Deep Learning algorithm to the PNP system.

## 4. PART II. ON CONVERGENCE OF DNN SOLUTIONS TO AN ANALYTIC SOLUTION TO THE VPFP SYSTEM AND SIMULATION RESULTS

In this section, we provide a DNN solution to the VPFP system. This section consists of three subsections. First, we propose the loss functions of the VPFP system for deep learning. We also prove the convergence of DNN solutions to an analytic solution of the VPFP system in two steps. Finally, we show that the simulation results on DNN solutions to the VPFP system agree with theoretical results by comparing the time-asymptotic behaviors and the macroscopic physical quantities which are defined in Section 1.6.1.

We will focus on the VPFP system (1.1) when the Knudsen number  $\varepsilon$  is 1 in this section. The fixed Knudsen number can be arbitrarily chosen. For the sake of simplicity, we abuse notations and write  $f_\varepsilon(t, x, v)$  as  $f(t, x, v)$  and  $E_\varepsilon(t, x)$  as  $E(t, x)$  in this section. Later, in Part IV (Sect. 6), we consider the varied Knudsen number regimes.

#### 4.1. Loss functions for the VPFP system

In Algorithm 1, the Adam optimizer finds the optimal parameters  $w^{\text{new}}$  and  $b^{\text{new}}$  in the direction of minimizing a loss function. Thus, we need to define the loss functions for the Vlasov–Poisson–Fokker–Planck system:  $\text{Loss}_{\text{GE}}^{fp}$  for the VPFP system (1.1)<sub>1</sub> and (1.1)<sub>3</sub>,  $\text{Loss}_{\text{IC}}^{fp}$  for the initial condition (1.1)<sub>2</sub> and (1.1)<sub>4</sub>,  $\text{Loss}_{\text{BC}}^{fp}$  for the boundary conditions (1.8) and (1.1)<sub>5</sub>. Note that we use the superscript  $\text{Loss}^{fp}$  for all loss functions to the VPFP system to distinguish it from the superscript  $\text{Loss}^{pnp}$  used for the loss functions to the PNP system in Section 5.1.

First, we define the following loss functions for the governing equation (1.1) as

$$\begin{aligned} \text{Loss}_{\text{GE}(1)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} & \int_{(0,T)} dt \int_{(-1,1)} dx \int_V dv |\partial_t f^{nn}(t, x, v; m, w, b) + v \partial_x f^{nn}(t, x, v; m, w, b) \\ & + E^{nn} \partial_v f^{nn} - (\partial_{vv} f^{nn}(t, x, v; m, w, b) + \partial_v(v f^{nn})(t, x, v; m, w, b))|^2, \end{aligned} \quad (4.1)$$

and

$$\text{Loss}_{\text{GE}(2)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \int_{(0,T)} dt \int_{(-1,1)} dx |\partial_x E^{nn}(t, x; m, w, b) - \int_V dv f^{nn}(t, x, v; m, w, b)|^2, \quad (4.2)$$

where  $V \stackrel{\text{def}}{=} [-10, 10]$ . Then we define  $\text{Loss}_{\text{GE}}^{fp}$  as

$$\text{Loss}_{\text{GE}}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{GE}(1)}^{fp} + \text{Loss}_{\text{GE}(2)}^{fp}.$$

We now define the loss function for the initial condition *via* the use of the initial grid points as

$$\text{Loss}_{\text{IC}(1)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \int_{(-1,1)} dx \int_V dv |f^{nn}(0, x, v) - f_0(x, v)|^2, \quad (4.3)$$

and

$$\text{Loss}_{\text{IC}(2)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \int_{(-1,1)} dx \left| E^{nn}(0, x; m, w, b) - \left( \int_{-1}^x dy \int_{\mathbb{R}} dv f_0(y, v) - (x+1) \right) \right|^2. \quad (4.4)$$

Note that we use the equation (1.3) for the loss function  $\text{Loss}_{\text{IC}}^2$ . Then, we define  $\text{Loss}_{\text{IC}}$  as

$$\text{Loss}_{\text{IC}}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{IC}(1)}^{fp} + \text{Loss}_{\text{IC}(2)}^{fp}.$$

The loss functions for the *specular* boundary condition for  $f$  in Section 1.5.1 and the Dirichlet boundary condition for  $E$  (1.1)<sub>5</sub> are defined as

$$\text{Loss}_{\text{BC}(1)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \int_{(0,T)} dt \int_{\gamma_-} dx dv |f^{nn}(t, x, v; m, w, b) - f^{nn}(t, x, -v; m, w, b)|^2, \quad (4.5)$$

and

$$\text{Loss}_{\text{BC}(2)}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \int_{(0,T)} dt \sum_{x \in \{-1, 1\}} |E^{nn}(t, x; m, w, b)|^2. \quad (4.6)$$

Then we define the total loss for the boundary conditions as

$$\text{Loss}_{\text{BC}}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{BC}(1)}^{fp} + \text{Loss}_{\text{BC}(2)}^{fp}.$$

Finally, we define the total loss as

$$\text{Loss}_{\text{Total}}^{fp}(f^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{GE}}^{fp} + \text{Loss}_{\text{IC}}^{fp} + \text{Loss}_{\text{BC}}^{fp}. \quad (4.7)$$

Note that we compute these loss functions *via* the approximation of the integration by the Riemann sum on the grid points, which is explained in Section 3.3. For example, the loss function  $\text{Loss}_{\text{GE}}^{fp}(f^{nn})$  can be approximated as

$$\begin{aligned} \text{Loss}_{\text{GE}}^{fp} \approx & \frac{1}{N_{i,j,k}} \sum_{i,j,k} \left| \partial_t f^{nn}(t_i, x_j, v_k; m, w, b) + v \partial_x f^{nn}(t_i, x_j, v_k; m, w, b) \right. \\ & \left. + E^{nn} \partial_v f^{nn} - (\sigma \partial_{vv} f^{nn}(t_i, x_j, v_k; m, w, b) + \beta \partial_v (v f^{nn})(t_i, x_j, v_k; m, w, b)) \right|^2 \\ & + \frac{1}{N_{i,j}} \sum_{i,j} \left| \partial_x E^{nn}(t_i, x_j; m, w, b) - \int_V dv f^{nn}(t_i, x_j, v_k; m, w, b) \right|^2, \end{aligned} \quad (4.8)$$

where  $N_{i,j,k}$  and  $N_{i,j}$  are the number of grid points.

## 4.2. On convergence of DNN solutions to analytic solutions to the VPFP system

In this section, we show the convergence of DNN solutions to analytic solutions to the VPFP system (1.1) in two steps. We first prove that there exists a sequence of neural network parameters (neuron numbers  $m$ , weights  $w$  and biases  $b$  as defined in Sect. 3.1) such that the total loss function  $\text{Loss}_{\text{Total}}^{fp}$  converges to 0. Sequentially, we also prove that if we minimize the total loss function  $\text{Loss}_{\text{Total}}^{fp}$ , it implies that the Deep Neural Network solution converges to an analytic solution. Throughout the section, we assume that the existence and the uniqueness of solutions for the VPFP system (1.1) with the specular boundary condition (1.8) are *a priori* given.

We first introduce the following definition and the theorem from [61] on the existence of approximated neural network solutions:

**Definition 4.1** ([61]). For a compact set  $K$  of  $\mathbb{R}^n$ , we say  $f \in \widehat{C}^{\xi(K)}$ ,  $\xi \in \mathbb{Z}_+^n$  if there is an open  $\Omega$  (depending on  $f$ ) such that  $K \subset \Omega$  and  $f \in C^{\xi}(\Omega)$ .

**Theorem 4.2** ([61], Thm. 2.1). Let  $K$  be a compact subset of  $\mathbb{R}^n$ ,  $n \geq 1$ , and  $f \in \widehat{C}^{\xi_1}(K) \cap \widehat{C}^{\xi_2}(K) \cap \dots \cap \widehat{C}^{\xi_q}(K)$ , where  $\xi_i \in \mathbb{Z}_+^n$  for  $1 \leq i \leq q$ . Also, let  $\bar{\sigma}$  be any non-polynomial function in  $C^l(\mathbb{R})$ , where  $l = \max\{|\xi_i| : 1 \leq i \leq q\}$ . Then for any  $\epsilon > 0$ , there is a network

$$f^{nn}(x) = \sum_{i=0}^{\nu} c_i \bar{\sigma}(\langle w_i, x \rangle + b_i), \quad x \in \mathbb{R}^n,$$

where  $c_i \in \mathbb{R}$ ,  $w_i \in \mathbb{R}^n$ , and  $b_i \in \mathbb{R}$ ,  $0 \leq i \leq \nu$  such that

$$\|D^k f - D^k f^{nn}\|_{L^\infty(K)} < \epsilon,$$

for  $k \in \mathbb{Z}_+^n$ ,  $k \leq \xi_i$ , for some  $i$ ,  $1 \leq i \leq q$ .

**Remark 4.3.**  $\nu$ ,  $n$ ,  $q$ , and  $\xi = (\xi_1, \xi_2, \xi_3)$  in Theorem 4.2 correspond to  $m$ , 3, 1, and  $(1, 1, 2)$  in our DNN setting in Section 3.1 for the VPFP system (1.1), respectively.

**Remark 4.4.** We can generalize the result above to the one with several hidden layers (see, [43]). Also, we may assume that the architecture is assumed to have only one hidden layer; *i.e.*,  $L = 2$ .

Now we introduce our first main theorem which states that a sequence of neural network solutions that makes the total loss function converge to zero exists if a  $\widehat{C}^{(1,1,2)}$  solution to the VPFP system exists:

**Theorem 4.5** ([52], Thm. 3.4). Assume that the number of layers  $L = 2$  and that the solution  $f$  to (1.1) with (1.8) which belongs to  $\widehat{C}^{(1,1,2)}([0, T] \times [-1, 1] \times V)$ , and the activation function  $\bar{\sigma}(x) \in C^{(2,2,3)}([0, T] \times [-1, 1] \times V)$  is non-polynomial. Then, there exists  $\{m_{[j]}, w_{[j]}, b_{[j]}\}_{j=1}^{\infty}$  such that a sequence of the DNN solutions  $f^{nn}$  of Theorem 4.2 with  $m_{[j]}$  nodes, denoted by

$$\{f_j(t, x, v) = f^{nn}(t, x, v; m_{[j]}, w_{[j]}, b_{[j]})\}_{j=1}^{\infty}$$

satisfies<sup>1</sup>

$$\text{Loss}_{\text{Total}}^{fp}(f_j) \rightarrow 0 \text{ as } j \rightarrow \infty. \quad (4.9)$$

*Proof.* This is similar to that of Theorem 3.4 of [52].  $\square$

**Remark 4.6.** The assumption  $f \in \widehat{C}^{(1,1,2)}([0, T] \times [-1, 1] \times V)$  can be replaced by a general Sobolev space, since the functions in a Sobolev space can be approximated by continuous functions on a compact set.

The first main Theorem 4.5 provides us that we can find the neural network parameters that reduce the pre-defined total loss function as much as we want. However, it does not imply that the DNN solutions converge to an analytic solution to the VPFP system. Therefore, we introduce our second main theorem, Theorem 4.7, which shows that the DNN solutions converge to an analytic solution in a suitable function space when we minimize the total loss function  $\text{Loss}_{\text{Total}}^{fp}$ . We assume that our compact domain  $V = [-10, 10]$  of the  $v$ -variable is chosen sufficiently large so that we can have

$$\|f\|_{L_x^1([-1, 1]; L_v^1(\mathbb{R} \setminus V))} \leq \epsilon \text{ and } |\partial_v^k f(t, x, v) - \partial_v^k f^{nn}(t, x, v)|_{v \in \partial V} \leq \epsilon, \quad (4.10)$$

for some sufficiently small  $\epsilon > 0$  and  $k = 0, 1$ .

**Theorem 4.7.** Assume that  $f$  is a solution to (1.1) with (1.8) which belongs to  $\widehat{C}^{(1,1,2)}([0, T] \times [-1, 1] \times V)$ . If the solution  $f$  and the Deep Neural Network solution  $f^{nn}(t, x, v; m, w, b)$  satisfy (4.10), then it implies that

$$\|f^{nn}(\cdot, \cdot, \cdot; m, w, b) - f\|_{L_t^\infty([0, T]; L_{x,v}^2([-1, 1] \times V))} \leq C \left( \text{Loss}_{\text{Total}}^{fp}(f^{nn}) + \epsilon \right), \quad (4.11)$$

where  $C$  is a positive constant depending only on  $T$ .

The proof of this theorem is provided in Appendix A.

**Remark 4.8.** Note that we fix the DNN architecture in Figure 3 before we train the DNN. Namely, we first fix the number of neurons for each layer  $m$  before training, and then we update the weights  $w$  and biases  $b$  to minimize the total loss function. Therefore, if we want to approximate the DNN solution  $f^{nn}$  to an analytic solution to the VPFP system, Theorem 4.7 indicates how much the total loss function  $\text{Loss}_{\text{Total}}(f^{nn}(t, x, v; m, w, b))$  has to be reduced. Then, Theorem 4.5 guarantees the existence of a 3-tuple  $(m, w, b)$  where the total loss function is sufficiently reduced as we want. In the DNN simulation, we use Algorithm 1 to find the optimal weights  $w$  and biases  $b$  to reduce the total loss function while the number of neurons for each layer  $m$  is fixed.

### 4.3. Neural Network simulations

In this section, we introduce numerical simulations for the solutions  $f^{nn}(t, x, v; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  to the VPFP system (1.1). We consider the following initial condition:

$$f(0, x, v) = f_0(x, v) = \begin{cases} e^{x-1} (1 - \cos(\frac{\pi}{2}v)), & \text{if } v \in (-4, 4), \\ 0, & \text{otherwise,} \end{cases} \quad (4.12)$$

which has different initial distributions at each position  $x \in [-1, 1]$ . We consider the time interval  $[0, 5]$  which is enough to reach the steady state of the solution to the VPFP system. Also, we set the background charge  $h(x)$

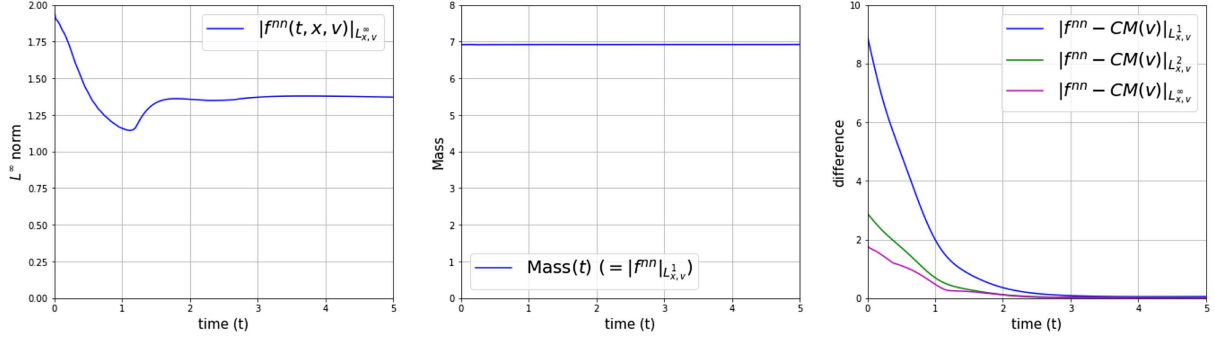


FIGURE 5. The time-asymptotic behaviors of the  $L^\infty$  norm,  $L^1$  norm of  $f^{nn}(t, x, v; m, w, b)$  (the first and the second plot) and the  $L^1$  norm,  $L^2$  norm, and  $L^\infty$  norm of the difference between  $f^{nn}(t, x, v; m, w, b)$  and the global Maxwellian  $\frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|}M(v)$ . It is notable that the total mass  $\text{Mass}(t)$  of the distribution  $f^{nn}(t, x, v; m, w, b)$  is conserved over time in the second plot. Also, note that the third plot shows that the distribution  $f^{nn}(t, x, v; m, w, b)$  converges to the global Maxwellian.

as constant that satisfies the global neutrality condition (1.2). More details about our Deep Learning algorithm are explained in Sections 3.2, 3.3, and the summary of the Deep Learning Algorithm 1.

The first plot in Figure 5 shows the time-asymptotic behaviors of the  $L^\infty$  norm of the distribution  $f^{nn}(t, x, v; m, w, b)$  with respect to position  $x$  and velocity  $v$ . After 3 time grids, the value converges to almost constant. This indicates that the distributions  $f^{nn}(t, x, v; m, w, b)$  converge to the steady state. It can be observed more clearly in the third plot in Figure 5, which shows the difference between the distribution  $f^{nn}(t, x, v; m, w, b)$  and the global equilibrium (1.14). The  $L^1$ ,  $L^2$  and  $L^\infty$  norm of the difference with respect to position  $x$  and velocity  $v$  tend to zero as time increases. This is consistent to our theoretical supports provided in the equation (1.14). Later, the pointwise values of  $f^{nn}(t, x, v; m, w, b)$  show the shape of the convergence to the global Maxwellian in Figure 7.

The second plot in Figure 5 shows the value of  $\text{Mass}(t)$  over time defined in (1.9). The plot shows that the total mass of the system is conserved. It shows an agreement with the theoretical result that the VPFP system with the specular boundary condition (1.8) yields the conservation of the total mass (1.9), which is an important a priori physical law for the VPFP system.

Figure 6 shows the time-asymptotic behaviors of four macroscopic quantities of  $f^{nn}(t, x, v; m, w, b)$ ; the total kinetic energy “KE” (1.18), the entropy “Ent” (1.17), the electric potential energy “EE” (1.19) and the free energy “FE” (1.20). The steady state values of these four macroscopic quantities can be obtained from the macroscopic quantities of the equilibrium in (1.14). Therefore, we expect the steady state values of the four macroscopic quantities as follows:

$$\text{KE}_\infty = \frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|}, \quad (4.13)$$

$$\text{Ent}_\infty = -\|f_0(\cdot, \cdot)\|_{L^1_{x,v}} \log \left( \frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|(2\pi)^{0.5}} \right) + \frac{1}{2}\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}, \quad (4.14)$$

$$\text{EE}_\infty = 0, \quad (4.15)$$

$$\text{FE}_\infty = \text{KE}_\infty - \text{Ent}_\infty + \text{EE}_\infty, \quad (4.16)$$

<sup>1</sup>Each of  $m_{[j]}, w_{[j]}, b_{[j]}$  represents the matrix of the numbers corresponding to  $f_j$  for each  $j = 1, 2, \dots, \infty$ . The matrices  $m_{[j]}, w_{[j]}, b_{[j]}$  consist of the element represented as  $m_{[j],ik}^{(l)}, w_{[j],ik}^{(l)}, b_{[j],ik}^{(l)}$ , respectively.

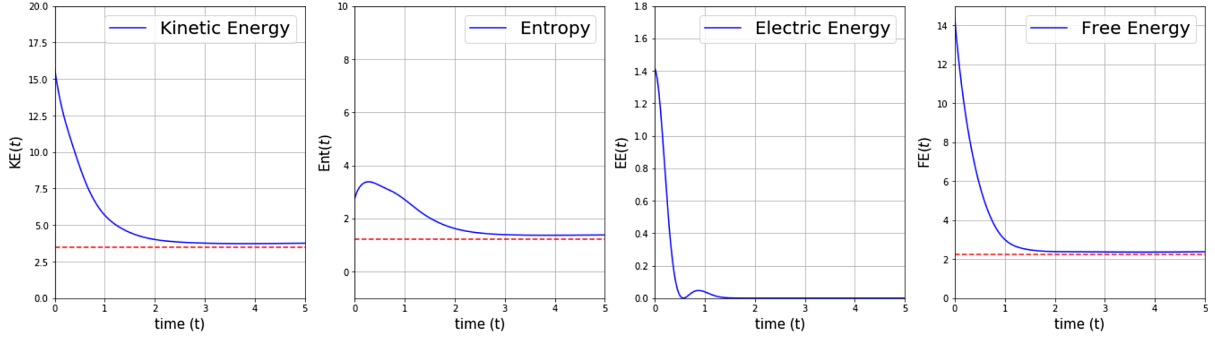


FIGURE 6. The time-asymptotic behaviors of the macroscopic quantities of  $f^{nn}(t, x, v; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$ . The steady-state values of the kinetic energy (4.13), the entropy (4.14), the free energy (4.16) are indicated in the red-dotted lines. Note that the free energy is monotonically decreasing.

where  $|\Omega| = 2$  and  $\|f_0(\cdot, \cdot)\|_{L^1_{x,v}} \approx 6.917$  in our case. We denote the steady state values *via* the red-dotted lines in Figure 6. The four plots show that each physical quantity converges to each steady state. Also, the fourth plot in Figure 6 shows a non-increasing trend of the free energy. This is also consistent to our theoretical supports of (1.26).

Figure 7 shows the pointwise values of the approximated neural network solution  $f^{nn}_\varepsilon(t, x, v; m, w, b)$  as time  $t$  varies at each position  $x = -1, -0.5, 0, 0.5$  and  $x = 1$ . Also, Figure 8 shows the pointwise values of the neural network solution  $E^{nn}_\varepsilon(t, x; m, w, b)$  as time  $t$  varies at some positions  $x = -1, -0.5, 0, 0.5$  and  $x = 1$  in different colors as shown in the legend. The two plots show that the  $f^{nn}(t, x, v; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  converge pointwisely to each equilibrium

$$f_{\varepsilon,\infty}(x, v) = \frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|} M(v) \text{ and } E_{\varepsilon,\infty}(x) = 0,$$

which is precisely explained in (1.14). We expect that the steady-state of the distribution  $f^{nn}(t, x, v; m, w, b)$  to the VPFP system has the same global Maxwellian at each position  $x \in [-1, 1]$  although the initial condition (4.12) has the different distributions at each position. To confirm this, we denote the global Maxwellian function  $f_{\varepsilon,\infty}(x, v)$  *via* the red-dotted lines in Figure 7. As we expect, Figure 7 shows that the distribution functions  $f^{nn}(t, x, v; m, w, b)$  converge to the same Maxwellian shape at time  $t = 5$ . The relative  $L^2_{x,v}$  error between the global Maxwellian  $f_{\varepsilon,\infty}(x, v)$  and the equilibrium of the neural network solution at  $t = 5$  is  $4.7 \times 10^{-3}$ . Also, the pointwise values of  $E^{nn}(t, x; m, w, b)$  for all positions  $x \in [-1, 1]$  converge to zero as shown in Figure 8. This result also shows an agreement with the theoretical steady-state (1.14).

## 5. PART III. ON CONVERGENCE OF DNN SOLUTIONS TO AN ANALYTIC SOLUTION TO THE PNP SYSTEM AND SIMULATION RESULTS

In this section, we provide a DNN solution to the PNP system (1.4). This section also consists of three subsections, similarly to Part II (Sect. 4). First, we propose the loss functions for the PNP system. Second, we prove the convergence of a DNN solution to an analytic solution to the PNP system in two steps. Finally, we show the simulation results of the DNN solutions to the PNP system by comparing the time-asymptotic behaviors, the macroscopic quantities, and the steady-state of the PNP system which is defined in Section 1.6.1.

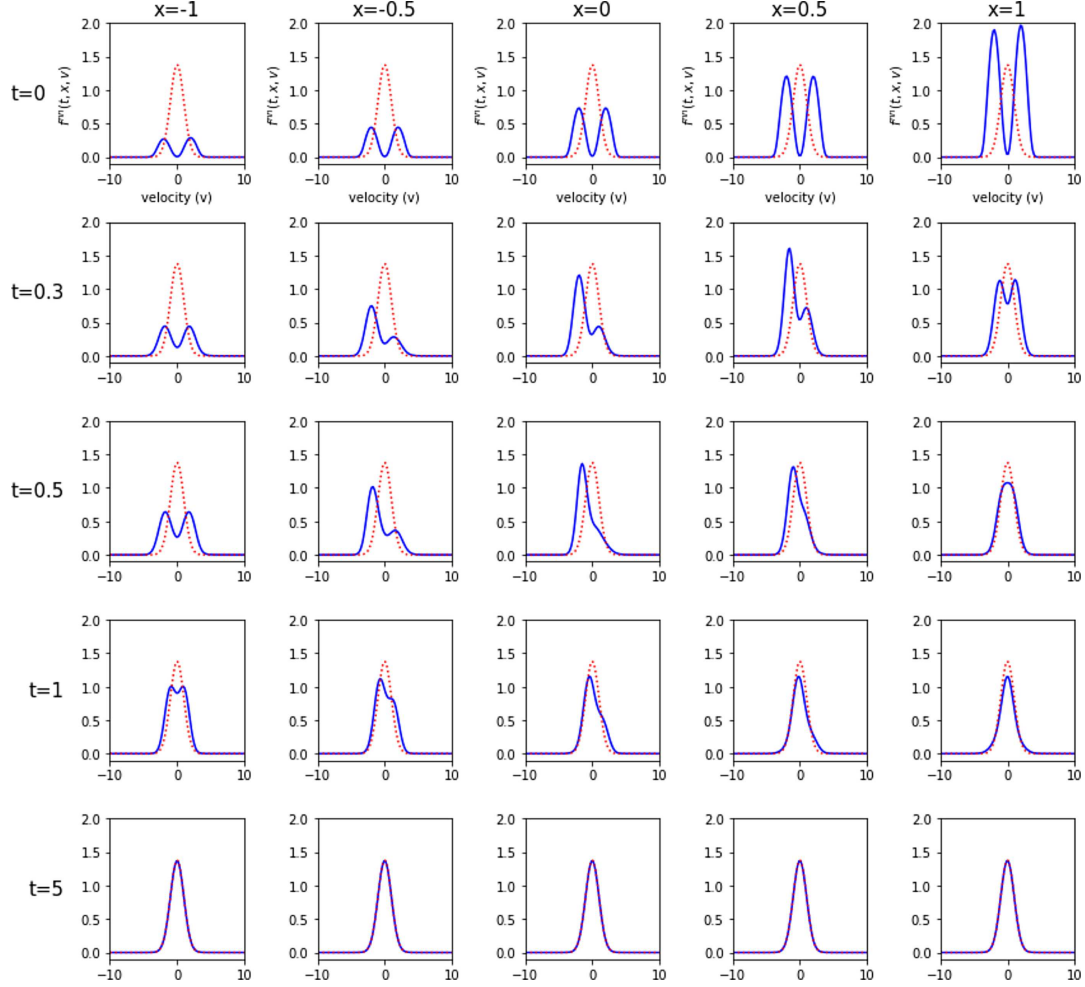


FIGURE 7. The pointwise values of  $f^{nn}(t, x, v; m, w, b)$  as time  $t$  varies at each position  $x$ 's.  $x = -1, -0.5, 0, 0.5, 1$  are the points to explain the convergence to the global Maxwellian  $\frac{\|f_0(\cdot, \cdot)\|_{L^1_{x,v}}}{|\Omega|} M(v)$ . The steady-state (global Maxwellian) is given *via* the red-dotted lines.

### 5.1. Loss functions for the PNP system

We need to define loss functions for the PNP system:  $\text{Loss}_{\text{GE}}^{np}$  for the PNP system (1.4)<sub>1</sub> and (1.4)<sub>3</sub>,  $\text{Loss}_{\text{IC}}^{np}$  for the initial condition (1.4)<sub>2</sub> and (1.4)<sub>4</sub> and  $\text{Loss}_{\text{BC}}^{np}$  for the boundary condition (1.10) and (1.4)<sub>5</sub>. Note that we use the superscript  $\text{Loss}^{np}$  for all loss functions to the PNP system to distinguish it from the superscript  $\text{Loss}^{vpfp}$  used for the loss functions to the VPFP system in Section 4.1.

First, we define loss functions for the governing equation as

$$\begin{aligned} \text{Loss}_{\text{GE}(1)}^{np}(\rho^{nn}) &\stackrel{\text{def}}{=} \int_{(0,T)} dt \int_{(-1,1)} dx |\partial_t \rho^{nn}(t, x; m, w, b) - \partial_{xx} \rho^{nn}(t, x; m, w, b) \\ &\quad + \partial_x(\rho^{nn}(t, x; m, w, b) E^{nn}(t, x; m, w, b))|^2, \end{aligned} \quad (5.1)$$

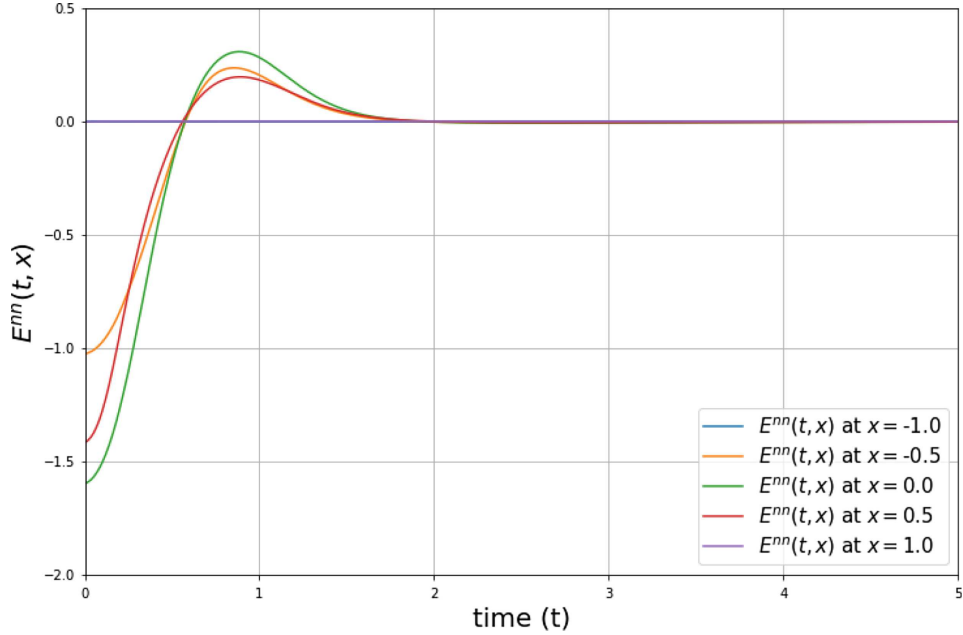


FIGURE 8. The pointwise values of  $E^{nn}(t, x; m, w, b)$  at each position  $x$ 's over time  $t$ . The values at each position  $x = -1, -0.5, 0, 0.5, 1$  are drawn in different colors as shown in the legend.

and

$$\begin{aligned} \text{Loss}_{\text{GE}(2)}^{pnp}(\rho^{nn}) &\stackrel{\text{def}}{=} \int_{(0,T)} dt \int_{(-1,1)} dx |\partial_x E^{nn}(t, x; m, w, b) - \rho^{nn}(t, x; m, w, b) + 1|^2 \\ &\quad + \int_{(0,T)} dt \int_{(-1,1)} dx |\partial_t (\partial_x E^{nn}(t, x; m, w, b) - \rho^{nn}(t, x; m, w, b) + 1)|^2 \\ &\quad + \int_{(0,T)} dt \int_{(-1,1)} dx |\partial_x (\partial_x E^{nn}(t, x; m, w, b) - \rho^{nn}(t, x; m, w, b) + 1)|^2. \end{aligned} \quad (5.2)$$

Note that this loss function  $\text{Loss}_{\text{GE}(2)}^{pnp}(\rho^{nn})$  is not just the  $L^2$  error with respect to  $t$  and  $x$ . We add the  $t$ -derivative and the  $x$ -derivative of the error to the original  $L^2$  error as shown in the definition (5.2). We need these two terms to prove the convergence of the neural network solution to the analytic solution in Theorem 5.2 in the following section. Then we define  $\text{Loss}_{\text{GE}}^{pnp}$  as

$$\text{Loss}_{\text{GE}}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{GE}(1)}^{pnp} + \text{Loss}_{\text{GE}(2)}^{pnp}.$$

We now define the loss function for the initial condition

$$\text{Loss}_{\text{IC}(1)}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \int_{(-1,1)} dx |\rho^{nn}(0, x; m, w, b) - \rho_0(x)|^2 \quad (5.3)$$

and

$$\text{Loss}_{\text{IC}(2)}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \int_{(-1,1)} dx \left| E^{nn}(0, x; m, w, b) - \left( \int_{-1}^x \rho_0(y) dy - (x+1) \right) \right|^2. \quad (5.4)$$

Then, we define  $\text{Loss}_{\text{IC}}^{pnp}$  as

$$\text{Loss}_{\text{IC}}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{IC}(1)}^{pnp} + \text{Loss}_{\text{IC}(2)}^{pnp}.$$



The loss function for the *Neumann* boundary condition for  $\rho(t, x)$  is defined as follows:

$$\text{Loss}_{\text{BC}(1)}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \int_{(0,T)} dt \int_{x \in \partial[-1,1]} dx |\partial_x \rho^{nn}(t, x; m, w, b)|^2. \quad (5.5)$$

We defined the loss function for the *Dirichlet* boundary condition for  $E(t, x)$

$$\begin{aligned} \text{Loss}_{\text{BC}(2)}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} & \int_{(0,T)} dt \int_{x \in \partial[-1,1]} dx (|E^{nn}(t, x; m, w, b)|^2) \\ & + \int_{(0,T)} dt \int_{x \in \partial[-1,1]} dx (|\partial_t E^{nn}(t, x; m, w, b)|^2). \end{aligned} \quad (5.6)$$

Note that we add the error of  $\partial_t E^{nn}(t, x; m, w, b)$  to the original  $L^2$  error as shown in the definition (5.6). This is also for the proof in Theorem 5.2 in the following section. Then, we define the total loss for the boundary conditions as

$$\text{Loss}_{\text{BC}}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{BC}(1)}^{pnp} + \text{Loss}_{\text{BC}(2)}^{pnp}.$$

Finally, we define the total loss as

$$\text{Loss}_{\text{Total}}^{pnp}(\rho^{nn}) \stackrel{\text{def}}{=} \text{Loss}_{\text{GE}}^{pnp} + \text{Loss}_{\text{IC}}^{pnp} + \text{Loss}_{\text{BC}}^{pnp}. \quad (5.7)$$

Note that we compute these loss functions *via* approximating the integration by the Riemann sum on the grid points similarly to Section 4.1.

## 5.2. On convergence of DNN solutions to an analytic solution to the PNP system

This section shows the convergence of the DNN solutions to an analytic solution to the PNP system (1.4) in two steps, similarly to Section 5.2. First, we prove that there exists a sequence of neural network parameters such that the total loss function  $\text{Loss}_{\text{Total}}^{pnp}$  converges to 0. We then show that the corresponding sequence of DNN solutions converges to an analytic solution if we minimize the total loss function  $\text{Loss}_{\text{Total}}^{pnp}$ . Throughout the section, we assume that the existence and the uniqueness of solutions for the PNP system (1.4) with the no-flux boundary condition (1.10) are *a priori* given.

We introduce our first main theorem similarly to that of Theorem 4.5 which shows the existence of a sequence of neural network parameters that makes the total loss function converge to zero if the  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$  solution to the PNP system exists:

**Theorem 5.1** ([52], Thm. 3.4). *Assume that the solution  $\rho$  to (1.4) with (1.10) which belongs to  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$ , and the activation function  $\bar{\sigma}(x) \in C^{(1,2)}([0, T] \times [-1, 1])$  is non-polynomial. Then, there exists  $\{m_{[j]}, w_{[j]}, b_{[j]}\}_{j=1}^{\infty}$  such that a sequence of the DNN solutions  $\rho^{nn}$  of Theorem 4.2 with  $m_{[j]}$  nodes, denoted by*

$$\{\rho_j(t, x, v) = \rho^{nn}(t, x, v; m_{[j]}, w_{[j]}, b_{[j]})\}_{j=1}^{\infty}$$

*satisfies<sup>2</sup>*

$$\text{Loss}_{\text{Total}}^{pnp}(\rho_j) \rightarrow 0 \text{ as } j \rightarrow \infty. \quad (5.8)$$

Now we introduce our second main theorem, which shows that the sequence of DNN solutions converges to an analytic solution to the PNP system in a suitable function space when we minimize the total loss function  $\text{Loss}_{\text{Total}}^{pnp}$ . We also refer to Remark 4.8, which explains how these main theorems are related to our Deep Learning algorithm.

---

<sup>2</sup>Each of  $m_{[j]}, w_{[j]}, b_{[j]}$  represents the matrix of the numbers corresponding to  $\rho_j$  for each  $j = 1, 2, \dots, \infty$ . The matrices  $m_{[j]}, w_{[j]}, b_{[j]}$  consist of the element represented as  $m_{[j], ik}^{(l)}, w_{[j], ik}^{(l)}, b_{[j], ik}^{(l)}$ , respectively.

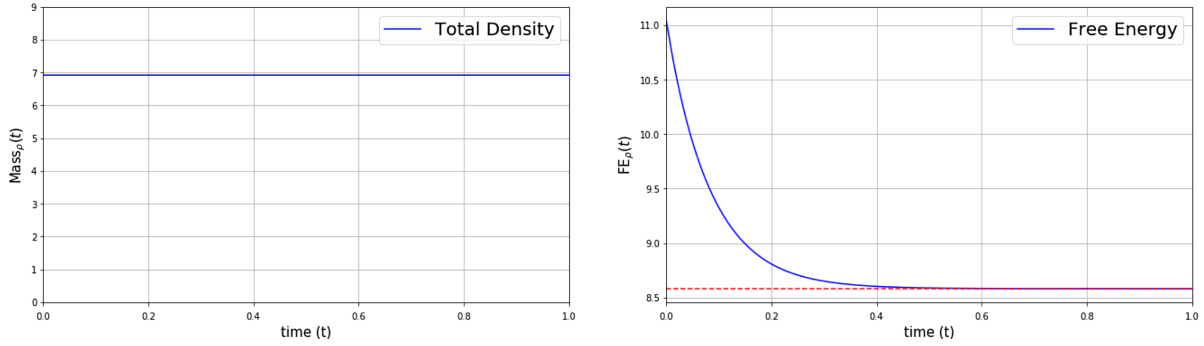


FIGURE 9. The time-asymptotic behaviors of the total density  $\text{Mass}_\rho(t)$  and the free energy  $\text{FE}_\rho(t)$  of the PNP system. In the second plot, the steady-state value of the free energy is indicated in the red-dotted line. Note that the free energy is monotonically decreasing.

**Theorem 5.2.** Assume that  $\rho$  is a solution to (1.4) with (1.10) which belongs to  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$ . Then, the Deep Neural Network solution  $\rho^{nn}(t, x; m, w, b)$  satisfies that

$$\|\rho^{nn}(\cdot, \cdot; m, w, b) - \rho\|_{L_t^\infty([0, T]; L_x^2([-1, 1]))} \leq C \text{Loss}_{\text{Total}}^{pnp}(\rho^{nn}), \quad (5.9)$$

where  $C$  is a positive constant depending only on  $T$ .

The proof of this theorem is provided in Appendix B.

### 5.3. Neural Network simulations

In this section, we provide numerical simulations for the solutions  $\rho^{nn}(t, x; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  to the PNP system (1.4). We set the initial condition of  $\rho^{nn}(t, x; m, w, b)$  as follows:

$$\rho_0(0, x) = \int_{\mathbb{R}} dv f_0(x, v) = 8e^{x-1}. \quad (5.10)$$

Note that we set the initial condition (5.10) which satisfies  $\rho(0, x) = \rho_0(x) = \int_{\mathbb{R}} dv f_0(x, v)$  to compare the convergence on the solutions of the VPFP system with the solutions of the PNP system in Part IV. We also set the background charge  $h(x)$  as constant to satisfy  $\int_{\Omega} \rho(0, x) - h(x) = 0$ . The details of our Deep Learning algorithm are explained in Sections 3.2, 3.3, and the summary of Algorithm 1.

Figure 9 shows the total density (1.11) and the free energy (1.25) of the PNP system. As shown in the left plot in Figure 9, the total density  $\text{Mass}_\rho(t)$  of the neural network solution  $\rho^{nn}(t, x; m, w, b)$  is conserved. It is well-matched to the theoretical result as in (1.11), which is an important property in the PNP system with the no-flux boundary condition (1.10). The right plot in Figure 9 shows the free energy of the neural network solution with the steady-state value *via* the red-dotted line. We compute the steady-state value of the free energy using the steady state of the PNP solution,  $\rho_\infty(x)$  and  $E_\infty(x)$ , in (1.24). We observe that the free energy is non-increasing as shown in the right plot in Figure 9. It verifies that the neural network solutions,  $\rho^{nn}(t, x; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$ , of the PNP system satisfy the dissipation law of the free energy as explained in (1.26).

Also, we expect that the free energy decreases exponentially to the steady-state based on Theorem 1.2 in [7]. In Figure 10, the plot shows the time evolution of the free energy of the neural network solution in a log-linear scale. We compute the decreasing rate of the free energy with the difference between the free energy  $\text{FE}_\rho(t)$  and the steady-state of the free energy  $\text{FE}_{\rho, \infty}$  at  $t = 1$ . We also denote the algebraic rates and the geometric rates in log-linear scale. We can observe that the decreasing rate of the free energy is almost similar to the geometric rate  $Ce^{-11.7t}$ , which is a linear function in Figure 10, except for a small error near the final time.

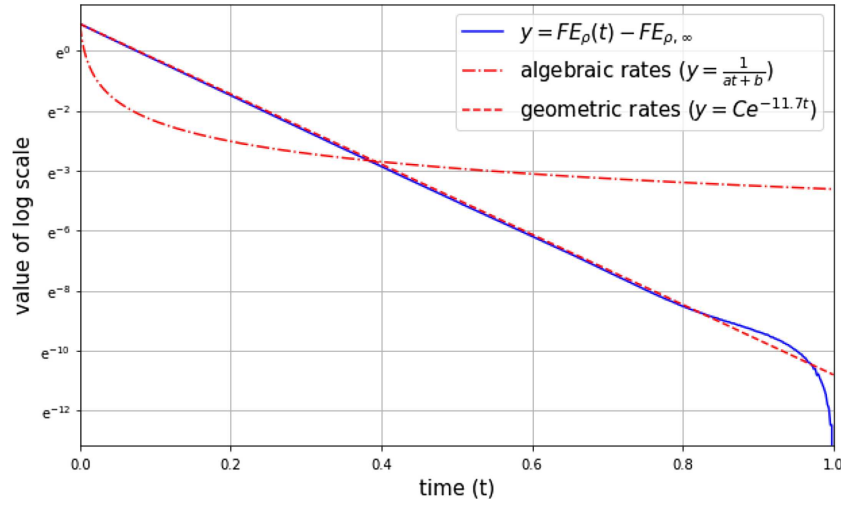


FIGURE 10. The time-asymptotic behaviors of the difference between the free energy  $FE_\rho(t)$  and the steady-state of the free energy  $FE_{\rho,\infty}$  in log-linear scale. We consider the numerical solution at  $t = 1$  as the steady state of the free energy. Note that this plot verifies the exponential decay of the approximated free energy.

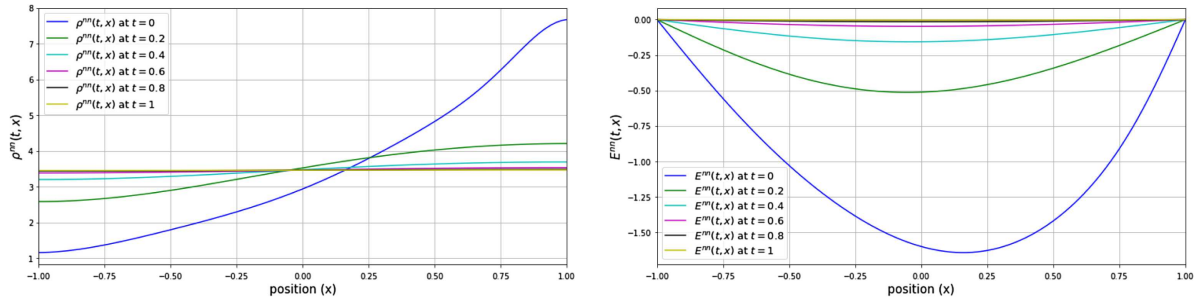


FIGURE 11. The pointwise values of  $\rho^{nn}(t, x; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  for each position  $x$  as time  $t$  varies. The values at each time  $t = 0, 0.2, 0.4, 0.6, 0.8, 1$  are drawn in different colors as shown in the legend.

Figure 11 shows the pointwise values of each approximated neural network solution  $\rho^{nn}(t, x; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  in different colors as time  $t$  varies at each position  $x$  from  $-1$  to  $1$ . We expect that the neural network solutions  $\rho^{nn}(t, x; m, w, b)$  and  $E^{nn}(t, x; m, w, b)$  converge pointwisely to each equilibrium

$$\rho_\infty(x) = C_{pnp} \text{ and } E_\infty(x) = 0,$$

which is precisely explained in (1.24). As shown in the first plot in Figure 11, the neural network solution  $\rho^{nn}(t, x; m, w, b)$  converges to constant for all  $x$ . It is well consistent to the theoretical supports provided in (1.24). Also, the second plot in Figure 11 shows that the neural network solution  $E^{nn}(t, x; m, w, b)$  converges to zero for all  $x \in [-1, 1]$  as  $t$  increases. This simulation result also well matches the expected steady state of the PNP system as in (1.24).

## 6. PART IV. ON THE SIMULATION RESULTS OF THE DIFFUSION LIMIT FROM THE VPFP SYSTEM TO THE PNP SYSTEM

In this section, we provide the trend of the diffusion limit from the VPFP system to the PNP system using the simulation results of our Deep Neural Network approach. We consider the convergence of the VPFP solutions to the PNP solution, as summarized in Part I (Sect. 2). We expect that the neural network solutions of the VPFP system and the PNP system have the trend of diffusion limit as explained in the equations (2.4) and (2.5). To observe the trend of the convergence, we compare the neural network solutions to the VPFP system with the Knudsen numbers  $\varepsilon = 1, 0.5, 0.2, 0.1, 0.05$  and the corresponding neural network solutions to the PNP system. The methods of how to train the neural network solutions to the VPFP system and the PNP system are precisely described in Section 3. Also, the results of the numerical simulations are given in Part II (Sect. 4) for the VPFP system and Part III (Sect. 5) for the PNP system, respectively.

As we have introduced in Section 3.4 on the simulation methodology, we use the “Grid Reuse” method to capture the VPFP with the small Knudsen number  $\varepsilon$ . When the “Grid Reuse” strategy is not used, the neural network solutions to the VPFP system could not approximate well at the early part of the time grid (about  $0.0 \sim 0.2$  time grids) as the Knudsen number  $\varepsilon$  is smaller. This means that the VPFP system with the small Knudsen number is hard to be approximated at the early time grid using Deep Learning. Therefore, the “Grid Reuse” method is essential to observe the diffusion limit from the neural network solution of the VPFP system to the neural network solution of the PNP system.

We define the total loss function (4.7) in the sense of the Mean Square Error (MSE). In this section, we use the Root Mean Square Error (RMSE) as the loss function for the VPFP system. These two cases show almost similar results, but we choose the RMSE loss function that offers better results. We use 50 reused grid points ( $k = 50$ ) in our work. Here we mention that it was not successful to approximate the solution when we chose  $k$  between 0 and 10 for the small Knudsen number  $\varepsilon$ . When  $k$  was selected around 50, the DNN solution well approximated the solution of the VPFP system with the Knudsen number  $\varepsilon$  being as small as 0.05. The most accurate DNN solution was obtained when we selected the value of  $k$  as 50.

The details of our Deep Learning algorithm are explained in Section 3.2 and the summary of the Deep Learning Algorithm 1.

### 6.1. Neural Network simulations

In this section, we present the results of the numerical simulations for the diffusion limit from the VPFP system to the PNP system. We set the initial condition (4.12) for the VPFP system and the initial condition (5.10) for the PNP system. It is worth noting that we do not change the number of the grid points for the VPFP system with any Knudsen numbers, *i.e.*, we analyze the diffusion limit in the sense of the Asymptotic-Preserving (AP) scheme. Instead, we use the “Grid Reuse” method for all neural network solutions.

Figure 12 indicates the total mass  $\int_{\Omega \times V} f_\varepsilon^{nn} dv dx$  of the VPFP system with the different Knudsen numbers  $\varepsilon = 1, 0.5, 0.2, 0.1$  and 0.05 in different colors as shown in the legend. As shown in Figure 12, all five graphs overlap so that they appear as one graph. This is because that all the five cases conserve the total mass over time. This plot implies that the neural network solutions for all 5 cases well approximate the solution of the VPFP system.

Figure 13 shows the  $L^\infty$  norm of the solution  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  to the VPFP system with the different Knudsen numbers  $\varepsilon = 1, 0.5, 0.2, 0.1$  and 0.05 in different colors as shown in the legend. Also, we plot the  $L^\infty$  norm of the neural network solution  $\rho^{nn}(t, x; m, w, b)M(v)$  via the red-dotted-line in Figure 13. We can observe that the  $L^\infty$  norm of the solution  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  converges pointwisely to the  $L^\infty$  norm of the  $\rho^{nn}(t, x; m, w, b)M(v)$  as the Knudsen number  $\varepsilon$  becomes close to zero. This gives more information than the theoretical result of convergence as explained in (2.4).

The graphs in Figure 14 and in Figure 15 show the pointwise values of the solutions as time  $t$  varies at each  $x$ 's. Figure 14 shows the pointwise values of  $\int f_\varepsilon^{nn}(t, x, v; m, w, b) dv$  as the Knudsen number  $\varepsilon$  varies in different colors as shown in the legend. We also plot the pointwise values of  $\rho^{nn}(t, x; m, w, b)$  via the red-dotted lines.

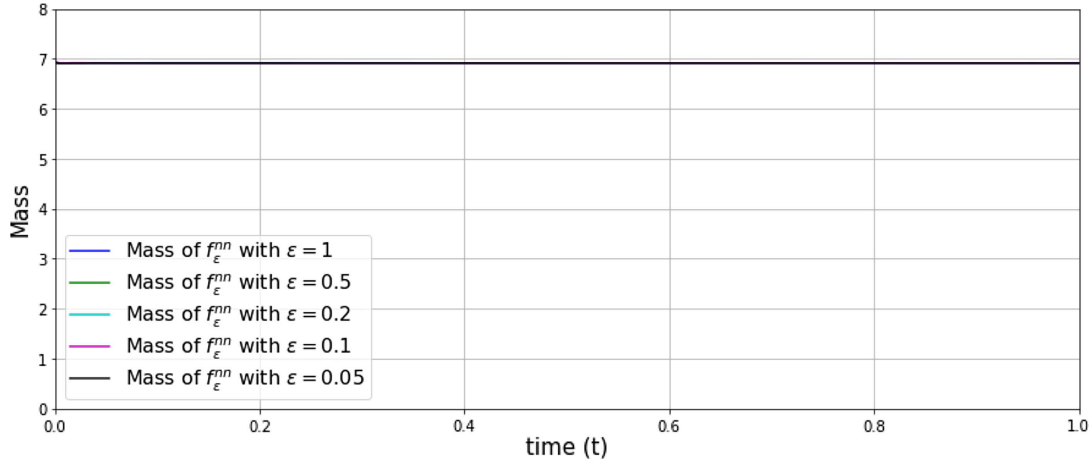


FIGURE 12. The time-asymptotic behaviors of the total mass of  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  ( $L_{x,v}^1$  norm of  $f_\varepsilon^{nn}$ ) as  $\varepsilon$  varies. The values with each Knudsen number  $\varepsilon$  are drawn in different colors as shown in the legend. It is notable that the total mass of the distribution  $f_\varepsilon^{nn}$  is conserved over time although the Knudsen number  $\varepsilon$  varies.

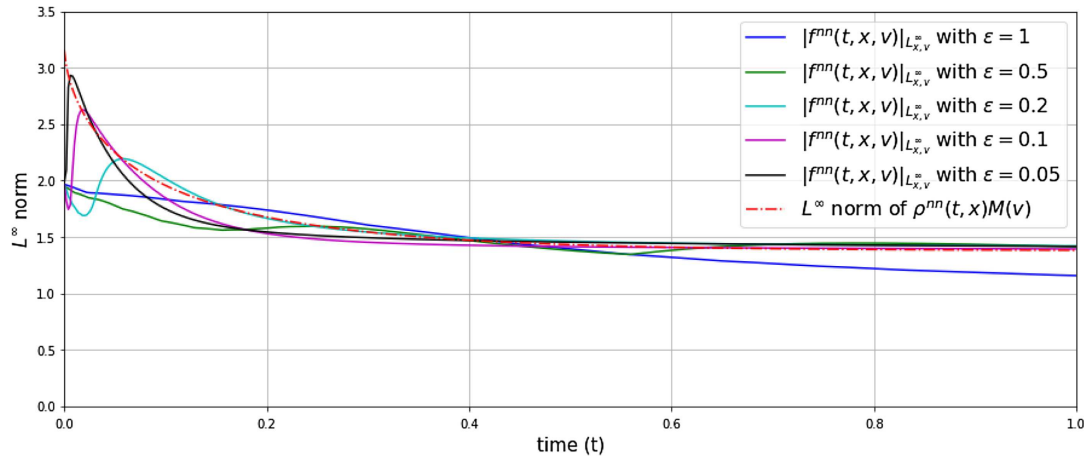


FIGURE 13. The time-asymptotic behavior of the  $L_{x,v}^\infty$  norm of  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  and  $\rho^{nn}(t, x; m, w, b)M(v)$  over time  $t$  as the Knudsen number  $\varepsilon$  varies. Each value is drawn in different colors as shown in the legend.

The first plot in Figure 14 shows that the initial condition (4.12) for the neural network solution of the VPFP system is consistent with the initial condition (5.10) for the neural network solution of the PNP system, since we set the initial conditions to satisfy the relation  $\int_{\mathbb{R}} f_0(t, x, v) dv = \rho_0(t, x)$ . It is remarkable that the neural network solutions to the VPFP system with the different Knudsen numbers well approximate the initial condition and the same for the solution to the PNP system. Also, we expect that the integration of neural network solution  $\int f_\varepsilon(t, x, v) dv$  converges to the density  $\rho(t, x; m, w, b)$  which is consistent to the convergence of  $f_\varepsilon$  to  $\rho(t, x)M(v)$  as explained in (2.4). As shown in the six plots in Figure 14, the pointwise values of the  $\int f_\varepsilon(t, x, v) dv$  to the VPFP system converge to the solution  $\rho(t, x; m, w, b)$  for all time  $t \in [0, 1]$  as the small Knudsen number  $\varepsilon$  becomes small. It is consistent to the theoretical supports in (2.4).

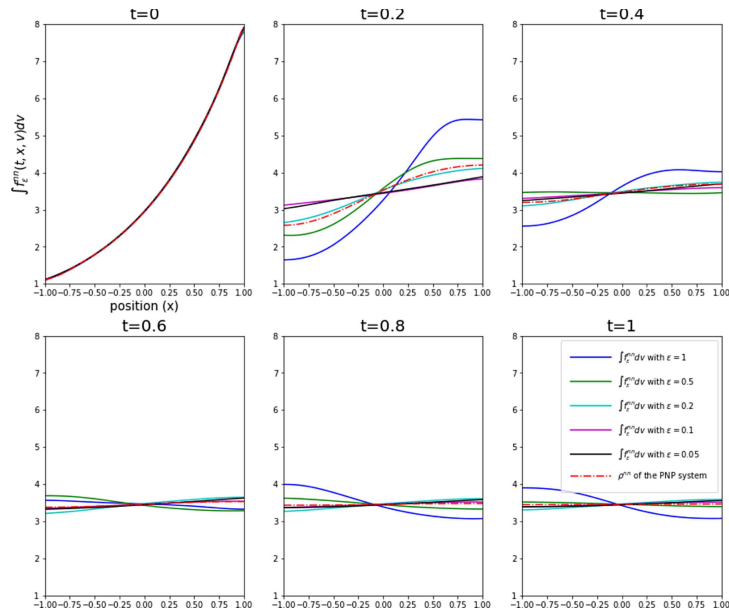


FIGURE 14. The pointwise values of  $\int f_{\varepsilon}^{nn}(t, x, v; m, w, b) dv$  to the VPFP system and  $\rho^{nn}(t, x; m, w, b)$  to the PNP system for each position  $x$  at time  $t = 0, 0.2, 0.4, 0.6, 0.8, 1$  as  $\varepsilon$  varies. The values with each Knudsen number  $\varepsilon$  are drawn in different colors as shown in the legend.

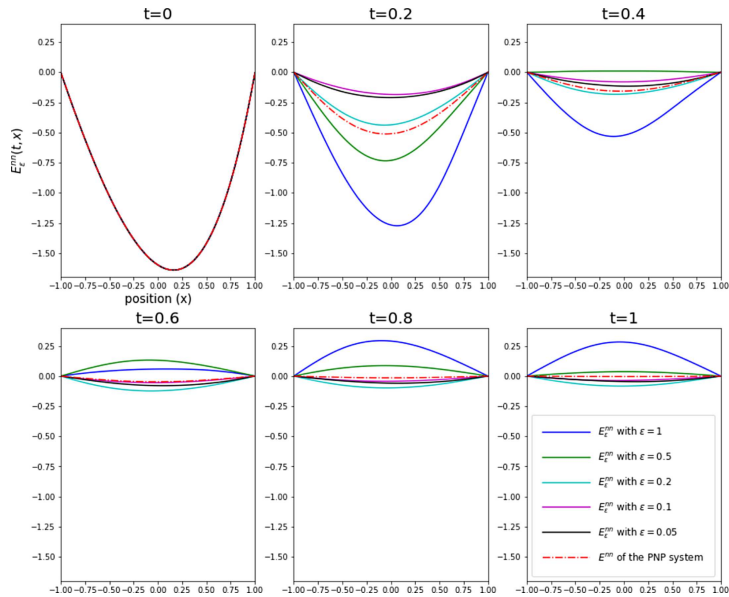


FIGURE 15. The pointwise values of  $E_{\varepsilon}^{nn}(t, x; m, w, b)$  to the VPFP system and  $E^{nn}(t, x; m, w, b)$  to the PNP system for each position  $x$  at time  $t = 0, 0.2, 0.4, 0.6, 0.8, 1$  as  $\varepsilon$  varies. The values with each Knudsen number  $\varepsilon$  are drawn in different colors as shown in the legend.

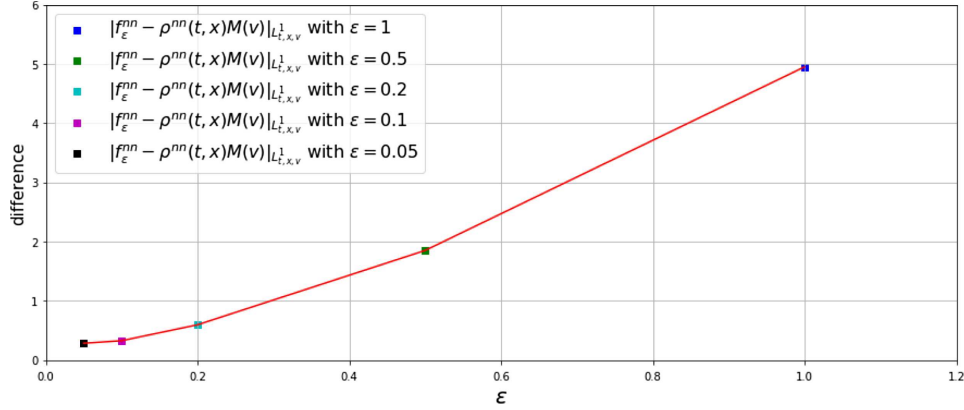


FIGURE 16. The values of  $L^1_{t,x,v}$  norm of the difference between  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  and  $\rho^{nn}(t, x; m, w, b)M(v)$  as  $\varepsilon$  varies.

Figure 15 shows the pointwise values of the electric force  $E_\varepsilon^{nn}(t, x; m, w, b)$  with the different Knudsen numbers  $\varepsilon = 1, 0.5, 0.2, 0.1$  and  $\varepsilon = 0.05$  in different colors as time  $t$  varies at each  $x$ 's. Also, we plot the electric force of the PNP system  $E^{nn}(t, x; m, w, b)$  in the red-dotted lines. We remark that the first plot in Figure 15 shows the same pointwise values of the electric force to the VPFP system with different Knudsen numbers and to the PNP system. It means that the initial conditions of the neural network solution of the electric force  $E^{nn}$  to the VPFP system and the PNP system are well approximated. Also, we observe that the six plots in Figure 15 show the solution  $E_\varepsilon^{nn}(t, x; m, w, b)$  of the VPFP system converges to the solution  $E^{nn}(t, x; m, w, b)$  of the PNP system as the Knudsen number  $\varepsilon$  goes to zero. It agrees with the theoretical result as explained in (2.5).

Finally, Figure 16 shows that the  $L^1_{t,x,v}$  norm of the difference between the distribution  $f_\varepsilon^{nn}(t, x, v; m, w, b)$  and the solution  $\rho^{nn}(t, x; m, w, b)M(v)$  as  $\varepsilon$  varies. We note that Figure 16 shows the convergence of  $f_\varepsilon$  to  $\rho(t, x)M(v)$  more quantitatively than the plots in the previous figures. As we expected in (2.4), the graph shows that the  $L^1_{t,x,v}$  norm of the difference between  $f_\varepsilon^{nn}$  and  $\rho^{nn}M$  becomes smaller as the Knudsen number  $\varepsilon$  tends to zero.

## 7. CONCLUSION

In this paper, we establish the commutation of the diagram of diffusion limit in Figure 1. This also implies the reduction of the VPFP system with the specular boundary condition to the PNP system with the no-flux boundary condition as the Knudsen number  $\varepsilon$  tends to zero. To this end, we have introduced the Deep Neural Network (DNN) solutions to the VPFP system and the PNP system using the Deep Learning algorithm. We use the two neural networks to approximate the VPFP system and the PNP system, coupled with the Poisson equation. Also, we propose appropriate loss functions for training, including the loss function for the initial conditions and the boundary conditions to each system: the VPFP system in Part II and the PNP system in Part III. We also provide the theoretical supports on which the approximated DNN solutions converge to analytic solutions of each system as the proposed total loss function tends to zero. We also provide the numerical simulations on the DNN solutions of each system, which support the theoretical predictions on the asymptotic behaviors of each system. These include the steady-states for the solutions and the physical quantities such as the total mass, the kinetic energy, the entropy, the electric energy, and the free energy.

Finally, using these DNN solutions of the two systems, we observe the trend of the diffusion limit in Part IV. We analyze our DNN solutions based on the theory shown in Part I. We use the newly devised technique ‘‘Grid Reuse’’ method adapted to the Deep Learning algorithm. This technique makes it possible to approximate the solution of the VPFP system with the Knudsen number  $\varepsilon$  in the range between 0.05 and 1.

We have provided the numerical simulation for the trend of the diffusion limit without a theoretical support and have seen that the DNN solution of the VPFP system converges to the DNN solution of the PNP system at the deep learning level. It is an interesting problem to theoretically prove the diffusion limit between the DNN solutions of those systems in the deep learning stage. Also, an improved method to approximate the VPFP system will be needed to make it work with a smaller value of the Knudsen number  $\varepsilon$  than 0.05. We leave these questions for future works.

One of the difficulties that the Deep Learning approach experiences as a PDE solver is on its rate of convergence and stability. Compared to the traditional numerical schemes which have a great amount of well-known studies on each method's performance, the Deep Learning method still has some difficulties in dealing with such things due to the optimization issues. However, it is also true that the Deep Learning is a new approach with many advantages as the Deep Learning method is a mesh-free method. The Deep Learning algorithm that we introduce in this paper does not require itself to have the mesh generation and instead it re-samples the grid points for each domain in every epoch. We expect that our work can be applied to arbitrary domains in higher dimensional kinetic equations.

## APPENDIX A. PROOF OF THEOREM 4.7

In this section, we provide the proof of Theorem 4.7.

*Proof.* Motivated by [17], we define a transform  $\bar{u}(t, x, v)$  of a function  $u(t, x, v)$  as follows:

$$\bar{u}(t, x, v) = e^{-t}u(t, x, e^{-t}v).$$

Then the transformed function  $\bar{f}$  satisfies

$$\partial_t \bar{f} + e^{-t}(v \cdot \partial_x) \bar{f} + e^t E \partial_v \bar{f} - e^{2t} \partial_v^2 \bar{f} = 0.$$

Also, we define the error values of the functions  $\bar{f}^{nn}$  and  $E^{nn}$  as the following equations:

$$d_{ge,j}^{(1)}(t, x, v) \stackrel{\text{def}}{=} [\partial_t + e^{-t}(v \cdot \partial_x) + e^t E^{nn} \partial_v - e^{2t} \partial_v^2] \bar{f}^{nn}$$

for  $(t, x, v) \in [0, T] \times [-1, 1] \times e^t V$ ,

$$d_{ge,j}^{(2)}(t, x) \stackrel{\text{def}}{=} -(\partial_x E^{nn} - \int dv \bar{f}^{nn}(t, x, v) + h(x))$$

for  $(t, x) \in [0, T] \times [-1, 1]$ ,

$$d_{bc,j}^{(1)}(t, x, v) \stackrel{\text{def}}{=} -(\bar{f}^{nn}(t, x, v) - \bar{f}^{nn}(t, x, -v))$$

for  $(t, x, v) \in \gamma_{T,e^t V}^-$ , and

$$d_{bc,j}^{(2)}(t, x) \stackrel{\text{def}}{=} -E^{nn}(t, x)$$

for  $(t, x) \in [0, T] \times \partial[-1, 1]$ . Note that the interval  $e^t V$  is defined as

$$e^t V \stackrel{\text{def}}{=} [-10e^t, 10e^t],$$

and  $\gamma_{T,e^t V}^\pm$  is defined as  $[0, T] \times \gamma_{e^t V}^\pm$ , where  $\gamma_{e^t V}^\pm$  is equal to  $\gamma^\pm$  with the velocity domain  $\mathbb{R}$  is replaced by  $e^t V$ .

We now consider the following equation on the difference between  $\bar{f}$  and  $\bar{f}^{nn}$  for each fixed  $j$  on the compact set of  $t, x, v$  only as

$$[\partial_t + e^{-t}(v \cdot \partial_x) - e^{2t} \partial_v^2] \{\bar{f} - \bar{f}^{nn}\} + e^t (E \partial_v \bar{f} - E^{nn} \partial_v \bar{f}^{nn}) = d_{ge,j}^{(1)}(t, x, v). \quad (\text{A.1})$$



Then we derive the inequality below by multiplying  $2(\bar{f} - \bar{f}^{nn})$  onto (A.1) and integrating it over  $[-1, 1] \times e^t V$  as

$$\begin{aligned} & \int_{-1}^1 \int_{-10e^t}^{10e^t} \frac{\partial}{\partial t} (\bar{f} - \bar{f}^{nn})^2(t, x, v) dv dx + \left( \int_{\gamma_{e^t V}} (\bar{f} - \bar{f}^{nn})^2 d\gamma \right) - 2e^{2t} \langle \partial_v^2 (\bar{f} - \bar{f}^{nn}), (\bar{f} - \bar{f}^{nn}) \rangle \\ &= - \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t (E \partial_v \bar{f} - E^{nn} \partial_v \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn}) dv dx + 2 \langle d_{ge,j}^{(1)}, (\bar{f} - \bar{f}^{nn}) \rangle, \quad (\text{A.2}) \end{aligned}$$

where  $\langle \cdot, \cdot \rangle$  denotes the standard inner product on  $L^2([-1, 1] \times e^t V)$ . On the left-hand side of (A.2), we note that

$$\begin{aligned} & \int_{\gamma_{e^t V}} (\bar{f} - \bar{f}^{nn})^2 d\gamma = \int_{\gamma_{e^t V}^+} (\bar{f} - \bar{f}^{nn})^2 d\gamma - \int_{\gamma_{e^t V}^-} (\bar{f} - \bar{f}^{nn})^2 d\gamma \\ &= \int_{\gamma_{e^t V}^+} (\bar{f} - \bar{f}^{nn})^2 d\gamma - \int_{\gamma_{e^t V}^-} \left( d_{bc,j}^{(1)}(t, x, v) + (\bar{f} - \bar{f}^{nn})(t, x, -v) \right)^2 d\gamma \\ &\geq \int_{\gamma_{e^t V}^+} (\bar{f} - \bar{f}^{nn})^2 d\gamma - 2 \int_{\gamma_{e^t V}^-} |d_{bc,j}^{(1)}(t, x, v)|^2 d\gamma - 2 \int_{\gamma_{e^t V}^-} (\bar{f} - \bar{f}^{nn})^2(t, x, -v) d\gamma \\ &= 3 \int_{\gamma_{e^t V}^+} (\bar{f} - \bar{f}^{nn})^2 d\gamma - 2 \int_{\gamma_{e^t V}^-} |d_{bc,j}^{(1)}(t, x, v)|^2 d\gamma \geq -2 \int_{\gamma_{e^t V}^-} |d_{bc,j}^{(1)}(t, x, v)|^2 d\gamma. \quad (\text{A.3}) \end{aligned}$$

Also, note that

$$\begin{aligned} & \frac{d}{dt} \|(\bar{f} - \bar{f}^{nn})(t, \cdot, \cdot)\|_{L_{x,v}^2([-1,1] \times e^t V)}^2 \\ &= \int_{-1}^1 \int_{-10e^t}^{10e^t} \frac{\partial}{\partial t} (\bar{f} - \bar{f}^{nn})^2(t, x, v) dv dx \\ &\quad + \underbrace{10e^t \left( \|(\bar{f} - \bar{f}^{nn})(t, \cdot, 10e^t)\|_{L_x^2([-1,1])}^2 + \|(\bar{f} - \bar{f}^{nn})(t, \cdot, -10e^t)\|_{L_x^2([-1,1])}^2 \right)}_{\stackrel{\text{def}}{=} B_1(t)}, \end{aligned}$$

by the Leibniz rule and

$$\begin{aligned} & 2e^{2t} \langle \partial_v^2 (\bar{f} - \bar{f}^{nn}), (\bar{f} - \bar{f}^{nn}) \rangle \\ &= -2e^{2t} \|\partial_v (\bar{f} - \bar{f}^{nn})\|_{L_{x,v}^2([-1,1] \times e^t V)}^2 \\ &\quad + \underbrace{2e^{2t} \int_{-1}^1 \partial_v (\bar{f} - \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn})(t, \cdot, 10e^t) - \partial_v (\bar{f} - \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn})(t, \cdot, -10e^t) dx}_{\stackrel{\text{def}}{=} B_2(t)}. \end{aligned}$$

So, it yields that

$$2e^{2t} \langle \partial_v^2 (\bar{f} - \bar{f}^{nn}), (\bar{f} - \bar{f}^{nn}) \rangle \leq B_2(t).$$

Also, note that

$$(E \partial_v \bar{f} - E^{nn} \partial_v \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn}) = (E - E^{nn}) \partial_v \bar{f} (\bar{f} - \bar{f}^{nn}) + \frac{1}{2} E^{nn} \partial_v (\bar{f} - \bar{f}^{nn})^2,$$

and Poisson equation implies the difference between  $E$  and  $E^{nn}$  as

$$\partial_x (E - E^{nn}) = \int_{e^t V} (\bar{f} - \bar{f}^{nn}) dv + \int_{\mathbb{R} \setminus e^t V} \bar{f} dv + d_{ge,j}^{(2)}(t, x). \quad (\text{A.4})$$

So, it yields that

$$\begin{aligned}
|E - E^{nn}|(t, x) &\leq \int_{-1}^x dx' \int_{e^t V} dv |\bar{f} - \bar{f}^{nn}|(t, x', v) + \int_{-1}^x dx' \left( \int_{\mathbb{R} \setminus e^t V} |\bar{f}(t, x', v)| dv \right) \\
&\quad + \int_{-1}^x dx' |d_{ge,j}^{(2)}(t, x')| + \underbrace{|(E - E^{nn})(t, x = -1)|}_{=|E^{nn}(t, x = -1)|} \\
&\leq \int_{-1}^x dx' \int_{e^t V} dv |\bar{f} - \bar{f}^{nn}|(t, x', v) + \int_{-1}^x dx' \left( \int_{\mathbb{R} \setminus V} dv |f(t, x', v)| \right) \\
&\quad + \sqrt{2} \|d_{ge,j}^{(2)}(t, x')\|_{L_x^2([-1, 1])} + \underbrace{\|d_{bc,j}^{(2)}(t, x)\|_{L_x^1(\partial[-1, 1])}}_{\leq \sqrt{2} \|d_{bc,j}^{(2)}(t, x)\|_{L_x^2(\partial[-1, 1])}},
\end{aligned}$$

by Hölder's inequality. Thus, we have

$$\|E(t) - E^{nn}(t)\|_{L_x^\infty} \leq \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} + \epsilon + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2} \|d_{bc,j}^{(2)}\|_{L_x^2}, \quad (\text{A.5})$$

by (4.10). Then the integration by parts in  $v$  variable yields that

$$\begin{aligned}
&\left| \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t (E \partial_v \bar{f} - E^{nn} \partial_v \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn}) dv dx \right| \\
&\leq \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t |E - E^{nn}| |\partial_v \bar{f}| |\bar{f} - \bar{f}^{nn}| dv dx + \frac{1}{2} \left| \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t E^{nn} \partial_v (\bar{f} - \bar{f}^{nn})^2 dv dx \right| \\
&\leq 2e^t \|\bar{f}\|_{C_{x,v}^1([-1, 1] \times e^t V)} \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} \\
&\quad \times \left( \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} + \epsilon + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2} \|d_{bc,j}^{(2)}\|_{L_x^2} \right) \\
&\quad + \frac{1}{2} \left| \int_{-1}^1 2e^t E^{nn} \left( (\bar{f} - \bar{f}^{nn})^2(t, x, 10e^t) - (\bar{f} - \bar{f}^{nn})^2(t, x, -10e^t) \right) dx \right| \\
&\leq 2e^t \|\bar{f}\|_{C_{x,v}^1([-1, 1] \times e^t V)} \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} \\
&\quad \times \left( \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} + \epsilon + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2} \|d_{bc,j}^{(2)}\|_{L_x^2} \right) \\
&\quad + e^t \|E^{nn}\|_{L_x^1} \left( \|(\bar{f} - \bar{f}^{nn})(t, \cdot, 10e^t)\|_{L_\infty^2}^2 + \|(\bar{f} - \bar{f}^{nn})(t, \cdot, -10e^t)\|_{L_\infty^2}^2 \right). \quad (\text{A.6})
\end{aligned}$$

Here, by Hölder's inequality, we have

$$\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^1([-1, 1] \times e^t V)} \leq \sqrt{40e^t} \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2}.$$

Also,  $f \in \hat{C}^{(1,1,2)}$  implies that

$$\|\bar{f}\|_{C_{x,v}^1([-1, 1] \times e^t V)} = \|e^{-t} f(t, x, e^{-t} v)\|_{C_{x,v}^1([-1, 1] \times e^t V)} \leq e^{-t} (C_0 + e^{-t} C_0),$$

for some positive constant  $C_0$ . Thus, by (4.10), we have

$$\begin{aligned}
&\left| \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t (E \partial_v \bar{f} - E^{nn} \partial_v \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn}) dv dx \right| \\
&\leq 2 \left( \sqrt{40e^t} \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2} + \epsilon + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2} \|d_{bc,j}^{(2)}\|_{L_x^2} \right) \\
&\quad \times (C_0 + e^{-t} C_0) \sqrt{40e^t} \|\bar{f} - \bar{f}^{nn}\|_{L^2} + 2 \|E^{nn}\|_{L_x^1} e^t \epsilon. \quad (\text{A.7})
\end{aligned}$$

Also, by (A.5), we have

$$\begin{aligned} \|E^{nn}\|_{L_x^1} &= \|E^{nn} - E + E\|_{L_x^1} \leq \|E - E^{nn}\|_{L_x^1} + \|E\|_{L_x^1} \\ &\leq 2 \left( \|f - f_j\|_{L_{x,v}^1([-1,1] \times V)} + \epsilon + \sqrt{2}\|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2}\|d_{bc,j}^{(2)}\|_{L_x^2} \right) + \|E\|_{L_x^1} \\ &\leq 2 \left( \sqrt{40e^t}\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2} + \epsilon + \sqrt{2}\|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2}\|d_{bc,j}^{(2)}\|_{L_x^2} \right) + \|E\|_{L_x^1}, \end{aligned}$$

by (4.10). Therefore, (A.7) yields that if  $\epsilon < 1$ , then

$$\begin{aligned} &\left| \int_{-1}^1 \int_{-10e^t}^{10e^t} 2e^t (E\partial_v \bar{f} - E^{nn}\partial_v \bar{f}^{nn}) (\bar{f} - \bar{f}^{nn}) \, dv dx \right| \\ &\leq 2 (C_0 + e^{-t}C_0) \sqrt{40e^t}\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2} \times \left( \sqrt{40e^t}\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2} + \epsilon + \sqrt{2}\|d_{ge,j}^{(2)}\|_{L_x^2} + \sqrt{2}\|d_{bc,j}^{(2)}\|_{L_x^2} \right) \\ &\quad + 2e^t\epsilon \left( \left( \sqrt{40e^t}\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2} + \epsilon + \mathcal{L}(t) \right) + \|E\|_{L_x^1} \right) \\ &\leq C_1\|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2}^2 + C_2\epsilon + C_3\|d_{ge,j}^{(2)}\|_{L_x^2}^2 + C_4\|d_{bc,j}^{(2)}\|_{L_x^2}^2, \end{aligned} \quad (\text{A.8})$$

for some positive constant  $C_1, C_2, C_3$ , and  $C_4$  by the use of Young's inequality. Note that  $\|E\|_{L_x^1}$  is also bounded as we have  $f \in \hat{C}^{(1,1,2)}$  on the compact domain and we also have  $\|f\|_{L_x^1([-1,1]; L_v^1(\mathbb{R} \setminus V))} \leq \epsilon$  by (4.10). Then, we can reduce (A.2) to

$$\begin{aligned} \frac{d}{dt} \overbrace{\|\bar{f} - \bar{f}^{nn}\|_{L^2([-1,1] \times e^t V)}^2}^{Y(t) \stackrel{\text{def}}{=}} &\leq (C_1 + 1) \|\bar{f} - \bar{f}^{nn}\|_{L_{x,v}^2([-1,1] \times e^t V)}^2 + B_1(t) + B_2(t) + C_4\epsilon \\ &\quad + \|d_{ge,j}^{(1)}(t, x, v)\|_{L_{x,v}^2([-1,1] \times e^t V)}^2 + C_3\|d_{ge,j}^{(2)}(t, x)\|_{L_x^2}^2 \\ &\quad + 2 \int_{\gamma_{T,e^t V}^-} |d_{bc,j}^{(1)}(t, x, v)|^2 d\gamma + C_4\|d_{bc,j}^{(2)}(t, x)\|_{L_x^2}^2. \end{aligned} \quad (\text{A.9})$$

If we use  $C_1 + 1 \leq C_5$  with some positive constant  $C_5$ , we can rewrite (A.9) as follows:

$$Y'(t) - C_5 Y(t) \leq B_1(t) + B_2(t) + C_4\epsilon + C_6 \underbrace{\left( \|d_{ge,j}^{(1)}\|_{L_{x,v}^2}^2 + \|d_{ge,j}^{(2)}\|_{L_x^2}^2 + \int_{\gamma_{T,e^t V}^-} |d_{bc,j}^{(1)}|^2 d\gamma + \|d_{bc,j}^{(2)}\|_{L_x^2}^2 \right)}_{\stackrel{\text{def}}{=} L(t)}, \quad (\text{A.10})$$

with some positive constant  $C_6$ . By Grönwall's inequality, we have

$$Y(t) \leq e^{C_5 t} \left( Y(0) + \int_0^t e^{-C_5 s} C_6 L(s) ds + \int_0^t e^{-C_5 s} (B_1(s) + B_2(s) + C_4\epsilon) ds \right). \quad (\text{A.11})$$

Finally, we recall that  $Y(t) = e^{-t}\|f - f_j\|_{L_{x,v}^2([-1,1] \times V)}^2$ ,  $Y(0) = \text{Loss}_{\text{IC}}^{fp}$ , and

$$Y(0) + \int_0^t e^{-C_5 s} L(s) ds \leq Y(0) + \int_0^T L(s) ds \leq C_7 \left( \text{Loss}_{\text{IC}}^{fp} + \text{Loss}_{ge}^{fp} + \text{Loss}_{\text{BC}}^{fp} \right) = C_7 \text{Loss}_{\text{Total}}^{fp}(f_j).$$

for some positive constant  $C_7$ . Moreover, under the assumption on (4.10), we have

$$\int_0^t e^{-C_5 s} B_1(s) ds \leq 40\epsilon^2 \int_0^t e^{-C_5 s} e^{-s} ds \leq C_8\epsilon^2 \text{ and } \int_0^t e^{-C_5 s} B_2(s) ds \leq 8\epsilon^2 \int_0^t e^{-C_5 s} e^{-s} ds \leq C_9\epsilon^2,$$

for some positive constant  $C_8$  and  $C_9$ . Therefore, (A.11) and the inverse transform from  $\bar{f}$  to  $f$  imply that

$$\|f - f^{nn}\|_{L_t^\infty([0,T]; L_{x,v}^2([-1,1] \times V))} \leq C (\text{Loss}_{\text{Total}}(f^{nn}) + \epsilon),$$

for some positive constant  $C$  which depends only on  $T$ . This completes the proof of Theorem 4.7.  $\square$

## APPENDIX B. PROOF OF THEOREM 5.2

In this section, we provide the proof of Theorem 5.2.

*Proof.* We define the error values of the neural network outputs  $\rho^{nn}$  and  $E^{nn}$  as the following equations:

$$d_{ge,j}^{(1)}(t, x) \stackrel{\text{def}}{=} -[\partial_t - \partial_{xx}] \rho^{nn} + \partial_x (\rho^{nn} E^{nn}) \quad \text{and} \quad d_{ge,j}^{(2)}(t, x) \stackrel{\text{def}}{=} -(\partial_x E^{nn} - \rho^{nn}(t, x) + h(x))$$

for  $(t, x) \in [0, T] \times [-1, 1]$ ,

$$d_{bc,j}^{(1)}(t, x) \stackrel{\text{def}}{=} -\partial_x \rho^{nn}(t, x) \quad \text{and} \quad d_{bc,j}^{(2)}(t, x) \stackrel{\text{def}}{=} -E^{nn}(t, x)$$

for  $x = \pm 1$ . Then, we consider the following equation on the difference between  $\rho$  and  $\rho^{nn}$  for each fixed  $j$  on the compact set of  $t, x$  only as

$$[\partial_t - \partial_{xx}] \{\rho - \rho^{nn}\} + \partial_x (\rho E - \rho^{nn} E^{nn}) = d_{ge,j}^{(1)}(t, x). \quad (\text{B.1})$$

Then we derive the inequality below by multiplying  $2(\rho - \rho^{nn})$  onto (B.1) and integrating it over  $[-1, 1]$  as

$$\int_{-1}^1 \frac{\partial}{\partial t} (\rho - \rho^{nn})^2(t, x) dx - 2 \int_{-1}^1 (\rho - \rho^{nn}) \partial_{xx} (\rho - \rho^{nn}) dx \quad (\text{B.2})$$

$$= -2 \int_{-1}^1 (\rho - \rho^{nn}) \partial_x (\rho E - \rho^{nn} E^{nn}) dx + 2 \langle d_{ge,j}^{(1)}, (\rho - \rho^{nn}) \rangle, \quad (\text{B.3})$$

where  $\langle \cdot, \cdot \rangle$  denotes the standard inner product on  $L_x^2([-1, 1])$ . On the left-hand side of (B.2), we note that

$$\int_{-1}^1 \frac{\partial}{\partial t} (\rho - \rho^{nn})^2(t, x) dx = \frac{\partial}{\partial t} \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2$$

and

$$\begin{aligned} \int_{-1}^1 (\rho - \rho^{nn}) \partial_{xx} (\rho - \rho^{nn}) dx &= \int_{\partial[-1, 1]} (\rho - \rho^{nn}) d_{bc,j}^{(1)} n_x dS_x - \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 \\ &\leq \frac{1}{2} \|\rho - \rho^{nn}\|_{L_x^2(\partial[-1, 1])}^2 + \frac{1}{2} \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1, 1])}^2 - \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 \\ &\leq \frac{1}{2} \left( \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2 + \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 \right) + \frac{1}{2} \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1, 1])}^2 - \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 \\ &\leq \frac{1}{2} \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2 - \frac{1}{2} \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 + \frac{1}{2} \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1, 1])}^2, \end{aligned} \quad (\text{B.4})$$

by the trace theorem for  $\rho - \rho^{nn}$ . Since  $\rho$  belongs to  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$  and  $\rho^{nn}$  is a continuous function with respect to  $x$ , it implies that  $|(\rho - \rho^{nn})(t, x)| < M_1$  on the compact domain  $x \in \partial[-1, 1]$  for some positive constant  $M_1$ . Also for the second term on the right-hand side of (B.2), we note that

$$\langle d_{ge,j}^{(1)}, (\rho - \rho^{nn}) \rangle \leq \frac{1}{2} \left( \|d_{ge,j}^{(1)}\|_{L_x^2([-1, 1])}^2 + \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2 \right).$$

Therefore, we reduce (B.2) to

$$\begin{aligned} \frac{\partial}{\partial t} \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2 - \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2 + \|\partial_x (\rho - \rho^{nn})\|_{L_x^2([-1, 1])}^2 - \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1, 1])}^2 \\ \leq -2 \int_{-1}^1 (\rho - \rho^{nn}) \partial_x (\rho E - \rho^{nn} E^{nn}) dx + \|d_{ge,j}^{(1)}\|_{L_x^2([-1, 1])}^2 + \|\rho - \rho^{nn}\|_{L_x^2([-1, 1])}^2. \end{aligned} \quad (\text{B.5})$$

Also, we reduce the absolute value of the first term on the right hand side of (B.5) to

$$\begin{aligned} & \left| -2 \int_{-1}^1 (\rho - \rho^{nn}) \partial_x (\rho E - \rho^{nn} E^{nn}) dx \right| \\ & \leq 2 \underbrace{\left| \int_{\partial[-1,1]} (\rho - \rho^{nn}) (\rho E - \rho^{nn} E^{nn}) n_x dS_x \right|}_{\stackrel{\text{def}}{=} B_1(t)} + 2 \underbrace{\left| \int_{-1}^1 \partial_x (\rho - \rho^{nn}) (\rho E - \rho^{nn} E^{nn}) dx \right|}_{\stackrel{\text{def}}{=} B_2(t)}, \end{aligned} \quad (\text{B.6})$$

by the integration by parts with respect to  $x$ . Note that we have

$$\begin{aligned} B_1(t) &= 2 \left| \int_{\partial[-1,1]} (\rho - \rho^{nn}) (-\rho^{nn} E^{nn}) n_x dS_x \right| = 2 \left| \int_{\partial[-1,1]} (\rho - \rho^{nn}) ((\rho - \rho^{nn}) - \rho) E^{nn} n_x dS_x \right| \\ &\leq 2 \left| \int_{\partial[-1,1]} (\rho - \rho^{nn})^2 d_{bc,j}^{(2)} n_x dS_x \right| + 2 \left| \int_{\partial[-1,1]} (\rho - \rho^{nn}) \rho d_{bc,j}^{(2)} n_x dS_x \right|. \end{aligned} \quad (\text{B.7})$$

Therefore, by Holder's inequality, trace theorem and the Cauchy–Schwarz inequality with  $\epsilon_0$ , we can bound  $B_1(t)$  as

$$\begin{aligned} B_1(t) &\leq 2 \left\| d_{bc,j}^{(2)} \right\|_{L_x^\infty(\partial[-1,1])} \left\| \rho - \rho^{nn} \right\|_{L_x^2(\partial[-1,1])}^2 + 2M_1 \left\| \rho - \rho^{nn} \right\|_{L_x^2(\partial[-1,1])} \left\| d_{bc,j}^{(2)} \right\|_{L_x^2(\partial[-1,1])} \\ &\leq \frac{1}{3} \left\| \rho - \rho^{nn} \right\|_{L_x^2(\partial[-1,1])}^2 + M_1 \left( \epsilon_0 \left\| \rho - \rho^{nn} \right\|_{L_x^2(\partial[-1,1])}^2 + \frac{1}{\epsilon_0} \left\| d_{bc,j}^{(2)} \right\|_{L_x^2(\partial[-1,1])}^2 \right) \\ &\leq \left( \frac{1}{3} + M_1 \epsilon_0 \right) \left\| \rho - \rho^{nn} \right\|_{L_x^2(\partial[-1,1])}^2 + \frac{M_1}{\epsilon_0} \left\| d_{bc,j}^{(2)} \right\|_{L_x^2(\partial[-1,1])}^2 \\ &\leq \left( \frac{1}{3} + M_1 \epsilon_0 \right) \left( \left\| \rho - \rho^{nn} \right\|_{L_x^2([-1,1])}^2 + \left\| \partial_x (\rho - \rho^{nn}) \right\|_{L_x^2([-1,1])}^2 \right) + \frac{M_1}{\epsilon_0} \left\| d_{bc,j}^{(2)} \right\|_{L_x^2(\partial[-1,1])}^2, \end{aligned} \quad (\text{B.8})$$

since we can reduce  $\text{Loss}_{\text{BC}^{(2)}}^{pnp}$  defined in (5.6) sufficiently small so we can bound  $\left\| d_{bc,j}^{(2)} \right\|_{L_x^\infty(\partial[-1,1])}$  for all time  $t \in [0, T]$  using the Sobolev embedding theorem as follows:

$$\begin{aligned} \left\| \left\| d_{bc,j}^{(2)} \right\|_{L_x^\infty(\partial[-1,1])} \right\|_{L_t^\infty([0,T])} &\leq \left\| d_{bc,j}^{(2)}(t, x = -1) \right\|_{L_t^\infty([0,T])} + \left\| d_{bc,j}^{(2)}(t, x = 1) \right\|_{L_t^\infty([0,T])} \\ &\leq C_0 \left( \left\| d_{bc,j}^{(2)}(t, x = -1) \right\|_{H_t^1([0,T])} + \left\| d_{bc,j}^{(2)}(t, x = 1) \right\|_{H_t^1([0,T])} \right) \\ &\leq \sqrt{2} C_0 \left( \left\| d_{bc,j}^{(2)}(t, x = -1) \right\|_{H_t^1([0,T])}^2 + \left\| d_{bc,j}^{(2)}(t, x = 1) \right\|_{H_t^1([0,T])}^2 \right) \\ &= \sqrt{2} C_0 \text{Loss}_{\text{BC}^{(2)}}^{pnp} < \frac{1}{6}, \end{aligned} \quad (\text{B.9})$$

for some positive constant  $C_0$ . We have some positive constant  $M_1$  satisfying  $|\rho(t, x)| < M_1$  on  $x \in \partial[-1, 1]$ , since  $\rho$  belongs to  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$ . By choosing a sufficiently small  $\epsilon_0$  so that  $\frac{1}{3} + M_1 \epsilon_0 < \frac{1}{2}$ , then we can bound  $B_1(t)$  as

$$B_1(t) \leq \frac{1}{2} \left\| \rho - \rho^{nn} \right\|_{L_x^2([-1,1])}^2 + \frac{1}{2} \left\| \partial_x (\rho - \rho^{nn}) \right\|_{L_x^2([-1,1])}^2 + C_1 \left\| d_{bc,j}^{(2)} \right\|_{L_x^2(\partial[-1,1])}^2, \quad (\text{B.10})$$

for some positive constant  $C_1$ .

To bound the second term  $B_2(t)$  on the right-hand side of the inequality (B.6), we have

$$\begin{aligned}
 B_2(t) &\leq 2 \left| \int_{-1}^1 \partial_x(\rho - \rho^{nn}) \rho (E - E^{nn}) dx \right| + 2 \left| \int_{-1}^1 \partial_x(\rho - \rho^{nn}) E^{nn} (\rho - \rho^{nn}) dx \right| \\
 &\leq \underbrace{\|\rho\|_{L_x^\infty([-1,1])} \left( \frac{1}{\epsilon_1} \|E - E^{nn}\|_{L_x^\infty([-1,1])}^2 + \epsilon_1 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \right)}_{\stackrel{\text{def}}{=} B_{2,1}(t)} \\
 &\quad + 2 \underbrace{\|E^{nn}\|_{L_x^\infty([-1,1])} \int_{-1}^1 |(\rho - \rho^{nn}) \partial_x(\rho - \rho^{nn})| dx}_{\stackrel{\text{def}}{=} B_{2,2}(t)}, \tag{B.11}
 \end{aligned}$$

by the Cauchy–Schwarz inequality with  $\epsilon_1$ . We consider the equation on the difference between  $E$  and  $E^{nn}$  for each fixed  $j$  on the compact set of  $t, x$  only as

$$\begin{aligned}
 |(E - E^{nn})(t, x)| &= \left| \int_{-1}^x \left( (\rho - \rho^{nn})(t, x') + d_{ge,j}^{(2)}(t, x') \right) dx' + (E - E^{nn})(t, x = -1) \right| \\
 &\leq \int_{-1}^x |\rho - \rho^{nn}|(t, x') dx' + \int_{-1}^x |d_{ge,j}^{(2)}(t, x')| dx' + |E^{nn}(t, x = -1)|. \tag{B.12}
 \end{aligned}$$

Thus, we have

$$\begin{aligned}
 \|E - E^{nn}\|_{L_x^\infty} &\leq \|\rho - \rho^{nn}\|_{L^1([-1,1])} + \|d_{ge,j}^{(2)}\|_{L^1([-1,1])} + |E^{nn}(t, x = -1)| \\
 &\leq \sqrt{2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])} + (|E^{nn}(t, x = -1)| + |E^{nn}(t, x = 1)|) \\
 &\leq \sqrt{2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])} + \|d_{bc,j}^{(2)}\|_{L^1(\partial[-1,1])} \\
 &\leq \sqrt{2} \left( \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])} + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])} \right), \tag{B.13}
 \end{aligned}$$

by Holder's inequality. Then the first term  $B_{2,1}(t)$  on the right-hand side of the inequality (B.11) is bounded as

$$\begin{aligned}
 B_{2,1}(t) &\leq M_2 \left( \frac{2}{\epsilon_1} \left( \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])} + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])} \right)^2 + \epsilon_1 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \right) \\
 &\leq M_2 \left( \frac{6}{\epsilon_1} \left( \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}^2 + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 \right) + \epsilon_1 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \right), \tag{B.14}
 \end{aligned}$$

with  $M_2$  satisfies  $|\rho(t, x)| < M_2$  on  $x \in [-1, 1]$  since the solution  $\rho$  belongs to  $\widehat{C}^{(1,2)}([0, T] \times [-1, 1])$ .

Now, we consider the second term  $B_{2,2}(t)$  on the right-hand side of the inequality (B.11). Since we reduce  $\text{Loss}_{\text{GE}(2)}^{np}$  sufficiently small, we can bound the  $\|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}$  for all time  $t \in [0, T]$  as

$$\begin{aligned}
 \left\| \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])} \right\|_{L_x^\infty([0,T])} &\leq \left\| \|d_{ge,j}^{(2)}\|_{L_x^\infty([-1,1])} \right\|_{L_x^\infty([0,T])} \\
 &\leq \|d_{ge,j}^{(2)}\|_{L_x^\infty([0,T] \times [-1,1])} \leq C_2 \|d_{ge,j}^{(2)}\|_{H_{t,x}^1([0,T] \times [-1,1])} = C_2 \text{Loss}_{\text{GE}(2)}^{np}{}^{\frac{1}{2}} \leq M_4, \tag{B.15}
 \end{aligned}$$

for some positive constant  $M_4$  by the Sobolev embedding theorem. Therefore, we can bound  $\|E - E^{nn}\|_{L_x^\infty}$  in the inequality (B.13) as

$$\|E - E^{nn}\|_{L_x^\infty([-1,1])} \leq \sqrt{2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \underbrace{\sqrt{2} \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}}_{\leq M_4} + \underbrace{\sqrt{2} \|d_{bc,j}^{(2)}\|_{L_x^\infty(\partial[-1,1])}}_{\leq \frac{1}{6}} \quad (\text{B.16})$$

using the inequalities (B.9) and (B.15). Using this bound, we can bound the second term  $B_{2,2}(t)$  on the right-hand side of the inequality (B.11) as

$$\begin{aligned} B_{2,2}(t) &\leq 2 \|\rho - \rho^{nn}\|_{L_x^2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2} \left( \|E - E^{nn}\|_{L_x^\infty([-1,1])} + \|-E\|_{L_x^\infty([-1,1])} \right) \\ &\leq 2 \|\rho - \rho^{nn}\|_{L_x^2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2} \left( \sqrt{2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])} + \sqrt{2} M_4 + \frac{\sqrt{2}}{6} + M_3 \right). \end{aligned}$$

Defining  $M_5 \stackrel{\text{def}}{=} \sqrt{2} M_4 + \frac{\sqrt{2}}{6} + M_3$  yields that

$$\begin{aligned} B_{2,2}(t) &\leq 2\sqrt{2} \|\rho - \rho^{nn}\|_{L_x^2}^2 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2} + 2M_5 \|\rho - \rho^{nn}\|_{L_x^2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2} \\ &\leq \frac{\sqrt{2}}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2}^4 + \sqrt{2} \epsilon_2 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 + M_5 \left( \frac{1}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2}^2 + \epsilon_2 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 \right) \quad (\text{B.17}) \end{aligned}$$

for any positive constant  $\epsilon_2$  and some positive constant  $M_3$  satisfying  $|E(t, x)| < M_3$  on  $x \in [-1, 1]$  since  $E$  is the continuous function with respect to  $x$ . Therefore, we can bound  $B_{2,2}(t)$  as

$$B_{2,2}(t) \leq \frac{M_5}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + \frac{\sqrt{2}}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^4 + \left( M_5 + \sqrt{2} \right) \epsilon_2 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2. \quad (\text{B.18})$$

Therefore, we can bound  $B_2(t) \leq B_{2,1}(t) + B_{2,2}(t)$  in the inequality (B.17) as

$$\begin{aligned} B_2(t) &\leq M_2 \left( \frac{6}{\epsilon_1} \left( \|\rho - \rho^{nn}\|_{L_x^2}^2 + \|d_{ge,j}^{(2)}\|_{L_x^2}^2 + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 \right) + \epsilon_1 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 \right) \\ &\quad + \frac{M_5}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2}^2 + \frac{\sqrt{2}}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2}^4 + \left( M_5 + \sqrt{2} \right) \epsilon_2 \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 \\ &\leq \left( \frac{6M_2}{\epsilon_1} + \frac{M_5}{\epsilon_2} \right) \|\rho - \rho^{nn}\|_{L_x^2}^2 + \frac{\sqrt{2}}{\epsilon_2} \|\rho - \rho^{nn}\|_{L_x^2}^4 + \frac{6M_2}{\epsilon_1} \|d_{ge,j}^{(2)}\|_{L_x^2}^2 \\ &\quad + \frac{6M_2}{\epsilon_1} \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 + \left( M_2 \epsilon_1 + \left( M_5 + \sqrt{2} \right) \epsilon_2 \right) \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 \quad (\text{B.19}) \end{aligned}$$

by Holder's inequality. By choosing sufficiently small  $\epsilon_1$  and  $\epsilon_2$  so that  $M_2 \epsilon_1 + (M_5 + \sqrt{2}) \epsilon_2 < \frac{1}{2}$ , then  $B_2(t)$  is bounded as

$$\begin{aligned} B_2(t) &\leq C_3 \left( \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^4 \right) \\ &\quad + C_4 \left( \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}^2 + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 \right) + \frac{1}{2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \quad (\text{B.20}) \end{aligned}$$

for some positive constant  $C_3$  and  $C_4$ . Therefore, by using the boundedness of  $B_1(t)$  in (B.10) and  $B_2(t)$  in (B.20), the inequality (B.6) yields that

$$\begin{aligned}
& \left| -2 \int_{-1}^1 (\rho - \rho^{nn}) \partial_x (\rho E - \rho^{nn} E^{nn}) dx \right| \leq B_1(t) + B_2(t) \\
& \leq \frac{1}{2} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + \frac{1}{2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 + C_1 \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 + C_3 \left( \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 \right. \\
& \quad \left. + \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^4 \right) + C_4 \left( \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}^2 + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 \right) + \frac{1}{2} \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \\
& \leq \left( C_3 + \frac{1}{2} \right) \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + C_3 \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^4 + \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 \\
& \quad + (C_1 + C_4) \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 + C_4 \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}^2. \tag{B.21}
\end{aligned}$$

Therefore, we reduce (B.5) to

$$\begin{aligned}
& \frac{\partial}{\partial t} \overbrace{\|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2}^{Y(t) \stackrel{\text{def}}{=}} - \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 + \|\partial_x(\rho - \rho^{nn})\|_{L_x^2([-1,1])}^2 - \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1,1])}^2 \\
& \leq \left( C_3 + \frac{1}{2} \right) \|\rho - \rho^{nn}\|_{L_x^2}^2 + C_3 \|\rho - \rho^{nn}\|_{L_x^2}^4 + \|\partial_x(\rho - \rho^{nn})\|_{L_x^2}^2 \\
& \quad + (C_1 + C_4) \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 + C_4 \|d_{ge,j}^{(2)}\|_{L_x^2}^2 + \|d_{ge,j}^{(1)}\|_{L_x^2}^2 + \|\rho - \rho^{nn}\|_{L_x^2}^2. \tag{B.22}
\end{aligned}$$

If we define the constant  $C_5 \stackrel{\text{def}}{=} C_2 + \frac{5}{2}$ , then we can rewrite inequality (B.22) as follows

$$Y'(t) - C_5 Y(t) \leq C_3 Y(t)^2 + C_6 \underbrace{\left( \|d_{ge,j}^{(1)}\|_{L_x^2([-1,1])}^2 + \|d_{ge,j}^{(2)}\|_{L_x^2([-1,1])}^2 + \|d_{bc,j}^{(1)}\|_{L_x^2(\partial[-1,1])}^2 + \|d_{bc,j}^{(2)}\|_{L_x^2(\partial[-1,1])}^2 \right)}_{\stackrel{\text{def}}{=} L(t)}, \tag{B.23}$$

with some positive constant  $C_6$ . Multiplying (B.23) by  $e^{-C_5 t}$  and integrating it over  $[0, t]$  for  $t < T$ , we have

$$e^{-C_5 t} Y(t) - Y(0) \leq C_3 \int_0^t e^{-C_5 s} Y(s)^2 ds + C_6 \int_0^t e^{-C_5 s} L(s) ds \leq C_3 \int_0^t Y(s)^2 ds + C_6 \int_0^t L(s) ds. \tag{B.24}$$

Therefore, we have the inequality

$$Y(t) \leq e^{C_5 t} \left( Y(0) + C_6 \int_0^t L(s) ds \right) + C_3 e^{C_5 t} \int_0^t Y(s)^2 ds. \tag{B.25}$$

By Theorem 25 of [28], we have

$$Y(t) \leq e^{C_5 t} \left( Y(0) + C_6 \int_0^t L(s) ds \right) \left[ 1 - C_3 \int_0^t e^{2C_5 s'} \left( Y(0) + C_6 \int_0^{s'} L(s) ds \right) ds' \right]^{-1} \tag{B.26}$$

for  $0 \leq t \leq \beta$ , where

$$\beta = \sup \left\{ t \in [0, T] \mid C_3 \underbrace{\int_0^t e^{2C_5 s'} \left( Y(0) + C_6 \int_0^{s'} L(s) ds \right) ds'}_{\stackrel{\text{def}}{=} h(s')} < 1 \right\}. \tag{B.27}$$



Note that  $Y(0) = \text{Loss}_{\text{IC}}^{\text{pnp}}(\rho^{nn})$  and

$$Y(0) + C_6 \int_0^t L(s) ds \leq Y(0) + C_6 \int_0^T L(s) ds = \text{Loss}_{\text{Total}}^{\text{pnp}}(\rho^{nn}).$$

Therefore, the value of  $h(s')$  in (B.27) is bounded as  $\text{Loss}_{\text{Total}}^{\text{pnp}}(\rho^{nn})$  which can be sufficiently small. Therefore, we can choose  $t = T$  in (B.26) to obtain that

$$\begin{aligned} \|\rho - \rho^{nn}\|_{L_x^2([-1,1])}^2 &\leq e^{C_5 T} \text{Loss}_{\text{Total}}^{\text{pnp}}(\rho^{nn}) \left[ 1 - C_3 \int_0^T e^{2C_5 s'} \left( Y(0) + C_6 \int_0^{s'} L(s) ds \right) ds' \right]^{-1} \\ &\leq C \text{Loss}_{\text{Total}}^{\text{pnp}}(\rho^{nn}) \end{aligned} \quad (\text{B.28})$$

for some positive constant  $C$ . Therefore, this completes the proof of theorem.  $\square$

*Acknowledgements.* J.Y. Lee and H.J. Hwang were supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF-2017R1E1A1A03070105 and NRF-2019R1A5A1028324). J.W. Jang was supported by the CRC 1060 *The Mathematics of Emergent Effects* of the University of Bonn funded through the German Science Foundation (DFG) and by Basic Science Research Institute Fund (NRF-2021R1A6A1A10042944).

## REFERENCES

- [1] E.J. Allen and H.D. Victory Jr, A computational investigation of the random particle method for numerical solution of the kinetic Vlasov–Poisson–Fokker–Planck equations. *Phys. A: Stat. Mech. App.* **209** (1994) 318–346.
- [2] C. Anitescu, E. Atroshchenko, N. Alajlan and T. Rabczuk, Artificial neural network methods for the solution of second order boundary value problems. *Comput. Mater. Continua* **59** (2019) 345–359.
- [3] A. Arnold, J.A. Carrillo, I. Gamba and C.-W. Shu, Low and high field scaling limits for the Vlasov- and Wigner–Poisson–Fokker–Planck systems. In: Vol. 30 of *The Sixteenth International Conference on Transport Theory, Part I* (Atlanta, GA, 1999) (2001) 121–153.
- [4] A. Arnold, P. Markowich and G. Toscani, On large time asymptotics for drift-diffusion-Poisson systems. In: Vol. 29 of *Proceedings of the Fifth International Workshop on Mathematical Aspects of Fluid and Plasma Dynamics* (Maui, HI, 1998) (2000) 571–581.
- [5] A.G. Baydin, B.A. Pearlmutter and A.A. Radul and J.M. Siskind, Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18** (2017) 43.
- [6] Y.A. Berezin, V.N. Khudick and M.S. Pekker, Conservative finite-difference schemes for the Fokker–Planck equation not violating the law of an increasing entropy. *J. Comput. Phys.* **69** (1987) 163–174.
- [7] P. Biler and J. Dolbeault, Long time behavior of solutions to Nernst–Planck and Debye–Hückel drift-diffusion systems. In: Vol. 1 of *Annales Henri Poincaré*. Springer (2000) 461–472.
- [8] P. Biler, W. Hebisch and T. Nadzieja, The Debye system: existence and large time behavior of solutions. *Nonlinear Anal.* **23** (1994) 1189–1209.
- [9] L.L. Bonilla and J. Soler, High-field limit of the Vlasov–Poisson–Fokker–Planck system: a comparison of differential perturbation methods. *Math. Models Methods Appl. Sci.* **11** (2001) 1457–1468.
- [10] L.L. Bonilla, J.A. Carrillo and J. Soler, Asymptotic behavior of an initial-boundary value problem for the Vlasov–Poisson–Fokker–Planck system. *SIAM J. Appl. Math.* **57** (1997) 1343–1372.
- [11] F. Bouchut, Existence and uniqueness of a global smooth solution for the Vlasov–Poisson–Fokker–Planck system in three dimensions. *J. Funct. Anal.* **111** (1993) 239–258.
- [12] F. Bouchut and J. Dolbeault, On long time asymptotics of the Vlasov–Fokker–Planck equation and of the Vlasov–Poisson–Fokker–Planck system with Coulombic and Newtonian potentials. *Differ. Integral Equ.* **8** (1995) 487–514.
- [13] C. Buet and S. Cordier, Conservative and entropy decaying numerical scheme for the isotropic Fokker–Planck–Landau equation. *J. Comput. Phys.* **145** (1998) 228–245.
- [14] C. Buet and S. Cordier, Numerical analysis of conservative and entropy schemes for the Fokker–Planck–Landau equation. *SIAM J. Numer. Anal.* **36** (1999) 953–973.
- [15] C. Buet, S. Cordier, P. Degond and M. Lemou, Fast algorithms for numerical, conservative, and entropy approximations of the Fokker–Planck–Landau equation. *J. Comput. Phys.* **133** (1997) 310–322.
- [16] L. Caffarelli, J. Dolbeault, P.A. Markowich and C. Schmeiser, On Maxwellian equilibria of insulated semiconductors. *Interfaces Free Bound.* **2** (2000) 331–339.
- [17] J.A. Carrillo, Global weak solutions for the initial-boundary-value problems to the Vlasov–Poisson–Fokker–Planck system. *Math. Methods Appl. Sci.* **21** (1998) 907–938.

- [18] J.A. Carrillo, J. Soler and J.L. Vázquez, Asymptotic behaviour and self-similarity for the three-dimensional Vlasov–Poisson–Fokker–Planck system. *J. Funct. Anal.* **141** (1996) 99–132.
- [19] L. Chacón, D.C. Barnes, D.A. Knoll and G.H. Miley, An implicit energy-conservative 2D Fokker–Planck algorithm. I. Difference scheme. *J. Comput. Phys.* **157** (2000) 618–653.
- [20] N. Crouseilles and M. Lemou, An asymptotic preserving scheme based on a micro-macro decomposition for collisional Vlasov equations: diffusion and high-field scaling limits. *Kinet. Relat. Models* **4** (2011) 441–477.
- [21] G. Cybenko, Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2** (1989) 303–314.
- [22] P. Degond, Asymptotic-preserving schemes for fluid models of plasmas. In: Numerical Models for Fusion. Vol. 39/40 of *Panor. Synthèses*. Soc. Math. France, Paris (2013) 1–90.
- [23] P. Degond and B. Lucquin-Desreux, An entropy scheme for the Fokker–Planck collision operator of plasma kinetic theory. *Numer. Math.* **68** (1994) 239–262.
- [24] G. Dimarco and L. Pareschi, Exponential Runge–Kutta methods for stiff kinetic equations. *SIAM J. Numer. Anal.* **49** (2011) 2057–2077.
- [25] G. Dimarco and L. Pareschi, Numerical methods for kinetic equations. *Acta Numer.* **23** (2014) 369–520.
- [26] J. Dolbeault, Stationary states in plasma physics: Maxwellian solutions of the Vlasov–Poisson system. *Math. Models Methods Appl. Sci.* **1** (1991) 183–208.
- [27] J. Dolbeault, Free energy and solutions of the Vlasov–Poisson–Fokker–Planck system: external potential and confinement (large time behavior and steady states). *J. Math. Pures Appl.* **78** (1999) 121–157.
- [28] S.S. Dragomir, Some Grönwall Type Inequalities and Applications. Nova Science Publishers, Inc., Hauppauge, NY (2003).
- [29] K. Dressler, Stationary solutions of the Vlasov–Fokker–Planck equation. *Math. Methods Appl. Sci.* **9** (1987) 169–176.
- [30] N. El Ghani and N. Masmoudi, Diffusion limit of the Vlasov–Poisson–Fokker–Planck system. *Commun. Math. Sci.* **8** (2010) 463–479.
- [31] F. Filbet and S. Jin, A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources. *J. Comput. Phys.* **229** (2010) 7625–7648.
- [32] F. Filbet and L. Pareschi, A numerical method for the accurate solution of the Fokker–Planck–Landau equation in the nonhomogeneous case. *J. Comput. Phys.* **179** (2002) 1–26.
- [33] A. Flavell, M. Machen, B. Eisenberg, J. Kabre, C. Liu and X. Li, A conservative finite difference scheme for Poisson–Nernst–Planck equations. *J. Comput. Electron.* **13** (2014) 235–249.
- [34] K.-I. Funahashi, On the approximate realization of continuous mappings by neural networks. *Neural Networks* **2** (1989) 183–192.
- [35] H. Gajewski and K. Gröger, On the basic equations for carrier transport in semiconductors. *J. Math. Anal. Appl.* **113** (1986) 12–35.
- [36] T. Goudon, Hydrodynamic limit for the Vlasov–Poisson–Fokker–Planck system: analysis of the two-dimensional case. *Math. Models Methods Appl. Sci.* **15** (2005) 737–752.
- [37] T. Goudon, J. Nieto, F. Poupaud and J. Soler, Multidimensional high-field limit of the electrostatic Vlasov–Poisson–Fokker–Planck system. *J. Differ. Equ.* **213** (2005) 418–442.
- [38] Y. Guo, H.J. Hwang, J.W. Jang and Z. Ouyang, The Landau equation with the specular reflection boundary condition. *Arch. Ration. Mech. Anal.* **236** (2020) 1389–1454.
- [39] J. Han, C. Ma, Z. Ma and E. Weinan, Uniformly accurate machine learning-based hydrodynamic models for kinetic equations. *Proc. Natl. Acad. Sci. USA* **116** (2019) 21983–21991.
- [40] K.J. Havlak and H.D. Victory, Jr, The numerical analysis of random particle methods applied to Vlasov–Poisson–Fokker–Planck kinetic equations. *SIAM J. Numer. Anal.* **33** (1996) 291–317.
- [41] M. Herda, On massless electron limit for a multispecies kinetic system with external magnetic field. *J. Differ. Equ.* **260** (2016) 7861–7891.
- [42] K. Hornik, Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4** (1991) 251–257.
- [43] K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (1989) 359–366.
- [44] H.J. Hwang and J. Jang, On the Vlasov–Poisson–Fokker–Planck equation near Maxwellian. *Discrete Contin. Dyn. Syst. Ser. B* **18** (2013) 681–691.
- [45] H.J. Hwang and J. Kim, The Vlasov–Poisson–Fokker–Planck equation in an interval with kinetic absorbing boundary conditions. *Stochastic Process. Appl.* **129** (2019) 240–282.
- [46] H.J. Hwang and D. Phan, On the Fokker–Planck equations with inflow boundary conditions. *Quart. Appl. Math.* **75** (2017) 287–308.
- [47] H.J. Hwang, J. Jang and J.J.L. Velázquez, The Fokker–Planck equation with absorbing boundary conditions. *Arch. Ration. Mech. Anal.* **214** (2014) 183–233.
- [48] H.J. Hwang, J. Jang and J. Jung, On the kinetic Fokker–Planck equation in a half-space with absorbing barriers. *Indiana Univ. Math. J.* **64** (2015) 1767–1804.
- [49] H.J. Hwang, J. Jang and J. Jung, The Fokker–Planck equation with absorbing boundary conditions in bounded domains. *SIAM J. Math. Anal.* **50** (2018) 2194–2232.
- [50] H.J. Hwang, J. Jang and J.J.L. Velázquez, Nonuniqueness for the kinetic Fokker–Planck equation with inelastic boundary conditions. *Arch. Ration. Mech. Anal.* **231** (2019) 1309–1400.

- [51] H.J. Hwang, J. Jang and J.J.L. Velázquez, On the structure of the singular set for the kinetic Fokker–Planck equations in domains with boundaries. *Quart. Appl. Math.* **77** (2019) 19–70.
- [52] H.J. Hwang, J.W. Jang, H. Jo, and J.Y. Lee, Trend to equilibrium for the kinetic Fokker–Planck equation *via* the neural network approach. *J. Comput. Phys.* **419** (2020) 109665.
- [53] Y. Hyon, B. Eisenberg and C. Liu, A mathematical model for the hard sphere repulsion in ionic solutions. *Commun. Math. Sci.* **9** (2011) 459–475.
- [54] S. Jin, Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations. *SIAM J. Sci. Comput.* **21** (1999) 441–454.
- [55] S. Jin, Asymptotic preserving (AP) schemes for multiscale kinetic and hyperbolic equations: a review. *Riv. Math. Univ. Parma (N.S.)* **3** (2012) 177–216.
- [56] S. Jin and L. Wang, An asymptotic preserving scheme for the Vlasov–Poisson–Fokker–Planck system in the high field regime. *Acta Math. Sci. Ser. B (Engl. Ed.)* **31** (2011) 2219–2232.
- [57] S. Jin and B. Yan, A class of asymptotic-preserving schemes for the Fokker–Planck–Landau equation. *J. Comput. Phys.* **230** (2011) 6420–6437.
- [58] H. Jo, H. Son, H.J. Hwang and E.H. Kim, Deep neural network approach to forward-inverse problems. *Netw. Heterog. Media* **15** (2020) 247–259.
- [59] I.E. Lagaris, A. Likas and D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Networks* **9** (1998) 987–1000.
- [60] I.E. Lagaris, A. Likas and D.G. Papageorgiou, Neural-network methods for boundary value problems with irregular boundaries. *IEEE Trans. Neural Networks* **11** (2000) 1041–1049.
- [61] X. Li, Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer. *Neurocomputing* **12** (1996) 327–343.
- [62] H. Liu and Z. Wang, A free energy satisfying finite difference method for Poisson–Nernst–Planck equations. *J. Comput. Phys.* **268** (2014) 363–376.
- [63] H. Liu and Z. Wang, A free energy satisfying discontinuous Galerkin method for one-dimensional Poisson–Nernst–Planck systems. *J. Comput. Phys.* **328** (2017) 413–437.
- [64] L. Lu, X. Meng, Z. Mao and G.E. Karniadakis, Deepxde: a deep learning library for solving differential equations. Preprint [arXiv:1907.04502](https://arxiv.org/abs/1907.04502) (2019).
- [65] W.S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5** (1943) 115–133.
- [66] J. Nieto, F. Poupaud and J. Soler, High-field limit for the Vlasov–Poisson–Fokker–Planck system. *Arch. Ration. Mech. Anal.* **158** (2001) 29–59.
- [67] L. Pareschi and G. Russo, Efficient asymptotic preserving deterministic methods for the Boltzmann equation, AVT-194 RTO AVT/VKI. In: *Models and Computational Methods for Rarefied Flows*. Lecture Series held at the von Karman Institute, Rhode St, Genese, Belgium (2011) 24–28.
- [68] L. Pareschi, G. Russo and G. Toscani, Fast spectral methods for the Fokker–Planck–Landau collision operator. *J. Comput. Phys.* **165** (2000) 216–236.
- [69] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, Automatic differentiation in Pytorch (2017).
- [70] F. Poupaud, Diffusion approximation of the linear semiconductor Boltzmann equation: analysis of boundary layers. *Asymptotic Anal.* **4** (1991) 293–317.
- [71] F. Poupaud and J. Soler, Parabolic limit and stability of the Vlasov–Fokker–Planck system. *Math. Models Methods Appl. Sci.* **10** (2000) 1027–1045.
- [72] M. Raissi, P. Perdikaris and G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** (2019) 686–707.
- [73] G. Rein and J. Weckler, Generic global classical solutions of the Vlasov–Fokker–Planck–Poisson system in three dimensions. *J. Differ. Equ.* **99** (1992) 59–77.
- [74] J. Schaeffer, Convergence of a difference scheme for the Vlasov–Poisson–Fokker–Planck system in one dimension. *SIAM J. Numer. Anal.* **35** (1998) 1149–1175.
- [75] J. Sirignano and K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* **375** (2018) 1339–1364.
- [76] T. Sokalski, P. Lingenfelter and A. Lewenstam, Numerical solution of the coupled Nernst–Planck and Poisson equations for liquid junction and ion selective membrane potentials. *J. Phys. Chem. B* **107** (2003) 2443–2452.
- [77] J. Soler, Asymptotic behaviour for the Vlasov–Poisson–Fokker–Planck system. In: Vol. 30 of *Proceedings of the Second World Congress of Nonlinear Analysts*, Part 8 (Athens, 1996) (1997) 5217–5228.
- [78] H.D. Victory, Jr and B.P. O’Dwyer, On classical solutions of Vlasov–Poisson Fokker–Planck systems. *Indiana Univ. Math. J.* **39** (1990) 105–156.
- [79] G.-W. Wei, Q. Zheng, Z. Chen and K. Xia, Variational multiscale models for charge transport. *SIAM Rev.* **54** (2012) 699–754.
- [80] S. Wollman and E. Ozizmir, Numerical approximation of the Vlasov–Poisson–Fokker–Planck system in one dimension. *J. Comput. Phys.* **202** (2005) 602–644.
- [81] S. Wollman and E. Ozizmir, A deterministic particle method for the Vlasov–Fokker–Planck equation in one dimension. *J. Comput. Appl. Math.* **213** (2008) 316–365.

- [82] S. Wollman and E. Ozizmir, Numerical approximation of the Vlasov–Poisson–Fokker–Planck system in two dimensions. *J. Comput. Phys.* **228** (2009) 6629–6669.
- [83] H. Wu, T.-C. Lin and C. Liu, Diffusion limit of kinetic equations for multiple species charged particles. *Arch. Ration. Mech. Anal.* **215** (2015) 419–441.

## Subscribe to Open (S2O)

### A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

#### **Please help to maintain this journal in open access!**

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting [subscribers@edpsciences.org](mailto:subscribers@edpsciences.org)

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/maths-s2o-programme>