

Learning collision operators from plasma phase space data using differentiable simulators

Diogo D. Carvalho^{1,2} , Pablo J. Bilbao^{1,3} , Warren B. Mori⁴, Luis O. Silva¹
and E. Paulo Alves^{2,4}

¹GoLP/Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade de Lisboa,
Lisbon 1049-001, Portugal

²Mani L. Bhaumik Institute for Theoretical Physics, University of California, Los Angeles, CA, USA

³The Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Oxford OX1 3NP, UK

⁴Department of Physics and Astronomy, University of California, Los Angeles, CA, USA

Corresponding author: Diogo D. Carvalho, diogo.d.carvalho@tecnico.ulisboa.pt

(Received 16 January 2026 UTC; revision received 23 March 2026 UTC; accepted 21 April 2026 UTC)

We propose a methodology to infer collision operators from phase space data of plasma dynamics. Our approach combines a differentiable kinetic simulator, whose core component in this work is a differentiable Fokker–Planck solver, with a gradient-based optimisation method to learn the collisional operators that best describe the phase space dynamics. We test our method using data from two-dimensional particle-in-cell simulations of spatially uniform thermal plasmas, and learn the collision operator that captures the self-consistent electromagnetic interaction between finite-size charged particles over a wide variety of simulation parameters. We demonstrate that the learned operators are more accurate than alternative estimates based on particle tracks, while making no prior assumptions about the relevant time scales of the processes and significantly reducing memory requirements. We find that the retrieved operators, obtained in the non-relativistic regime, are in excellent agreement with theoretical predictions derived for electrostatic scenarios. Our results show that differentiable simulators offer a powerful and computational efficient approach to infer novel operators for a wide range of problems, such as electromagnetically dominated collisional dynamics and stochastic wave–particle interactions.

Key words: plasma dynamics, plasma simulation

1. Introduction

Plasma physics has seen significant advances in the ability to simulate nonlinear plasma dynamics from first principles. However, capturing the complex multi-scale dynamics of plasmas, especially in non-equilibrium and strongly coupled regimes, remains challenging (Kushner 2020). One such challenge is the development of collisional operators that capture the macroscopic dynamics in regimes where analytical theory does not exist or is expected to fail, e.g. regimes where the collisional dynamics are mediated by electromagnetic interactions (relativistic temperatures)

(Braams & Karney 1987; Pike & Rose 2016) or when assumptions about the predominance of small-angle scattering events break down (strongly coupled regimes) (Baalrud 2012; Baalrud & Daligault 2014). These regimes occur for example in inertial confinement fusion experiments (Atzeni & Meyer-ter Vehn 2004), dense astrophysical objects (e.g. white dwarfs, neutron stars, core of gas giants) (Chabrier, Saumon & Potekhin 2006) and in dusty (Merlino & Goree 2004) and ultracold (Killian *et al.* 2007) astrophysical and laboratory plasmas. It is therefore important to design first-principles numerical simulations to probe deviations from the theory and to extract improved models which can serve as the building blocks for future theoretical developments.

For weakly coupled plasmas, where small-angle Coulomb collisions dominate the dynamics, the Fokker–Planck (FP) collision operator is expected to provide a good description of the collisional dynamics (Landau 1936; Rosenbluth, MacDonald & Judd 1957). The operator can be written as (Rosenbluth *et al.* 1957)

$$\left(\frac{\partial f}{\partial t}\right)_{FP} = -\nabla_v \cdot (\mathbf{A}f) + \frac{1}{2} \nabla_v \cdot [\nabla_v \cdot (\mathbf{D}f)], \quad (1.1)$$

where $\mathbf{v} \in \mathbb{R}^N$ is the velocity vector in the N -dimensional phase space, $f(\mathbf{v}, t) : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}$ is the underlying spatially distribution function (for simplicity here we assume that f does not depend on the position $\mathbf{r}$), $\mathbf{A}(\mathbf{v}, t) : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}^N$ is the advection (drift) vector and $\mathbf{D}(\mathbf{v}, t) : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}^{N \times N}$ is the diffusion matrix.

The FP operator is often coupled with the Vlasov and Maxwell systems of equations to self-consistently model collisional transport and the kinetic evolution of particle distribution functions in phase space (Bell *et al.* 2006; Tzoufras *et al.* 2011; Thomas *et al.* 2012; Tzoufras *et al.* 2013; Bell & Sherlock 2024). Furthermore, it has been shown that other stochastic acceleration processes mediated by turbulence (and not collisional dynamics) can also be studied using this description (Wong *et al.* 2020, 2025; Camporeale *et al.* 2022).

Significant theoretical work has been made to determine the mathematical form of the advection and diffusion coefficients that capture the dynamics of interest (Rosenbluth *et al.* 1957; Braams & Karney 1987; Reynolds, Fried & Morales 1997). Usually, these efforts result in collisional integrals, which can be costly to compute and to treat analytically. When theory is not available, both numerical and experimental tools have been used to estimate such coefficients, e.g. particle track information from numerical simulations is commonly used to infer the coefficients for a set-up of interest (Hockney 1971; Matsuda & Okuda 1975; Wong *et al.* 2020, 2025). While particle tracks provide all the required information to estimate the coefficients, it is often unfeasible to store and post-process all the necessary data for large-scale simulations and broad parameter scans. Furthermore, to accurately estimate the coefficients (at run time) and, therefore, avoid storing the full tracks, one needs to a priori have an estimate of all the acceleration (forces) time scales involved (not only collisions). In particular, one must guarantee that the time step over which velocity change statistics are computed fulfils the Markovian assumption underpinning the FP equation.

In this work, we propose a general-purpose framework to learn collision operators from phase space data (in this paper, obtained from kinetic numerical simulations) by rewriting the problem as an optimisation task, which we tackle using a differentiable kinetic simulator and gradient-based minimisation methods. The simulator evolves phase space distributions using a differentiable FP solver, where the

advection and diffusion tensors are learned from data. By making use solely of phase space data, we greatly reduce the memory requirements associated with storing particle tracks. Furthermore, the differentiable simulator allows us to extract operators that accurately capture the long-term dynamics without prior knowledge of the involved time scales. The technique is general and can be used with other sets of phase space information, e.g. from experimental (Bergeson *et al.* 2025) or observational (Camporeale *et al.* 2022) data, and the simulator can be extended to include different collision operator forms, external fields or other options deemed relevant for the problem under study.

As a first test case, we use these tools to extract the collision operator for an electromagnetic particle-in-cell (PIC) code (Birdsall & Langdon 2018), which evolves the self-consistent fields of finite-size particles, and compare the results against theory originally proposed by Langdon (1970) and Langdon & Birdsall (1970), and against coefficients derived from particle tracks. This is a good test bench since theoretical predictions assume that a FP-type collision operator correctly captures the self-consistent collisional dynamics of the PIC algorithm. We note that the collision operator we aim to extract is not explicitly solved by the PIC algorithm, e.g. via a Monte Carlo module (Takizuka & Abe 1977) or a grid-based Langevin operator (Manheimer, Lampe & Joyce 1997). The learned operator is instead a consequence of the self-consistent interactions between the finite-size particles and the generated fields in a regime where small-angle scattering events are expected to dominate.

This paper is structured as follows. In §§ 1.1 and 1.2 we review the underpinning theoretical and numerical results. In § 2 we describe the methods deployed in this paper to learn collision operators. In § 3 we present the main results for the collision operator learned for plasmas of finite-size particles. Finally, in § 4 we state the conclusions and describe directions for future work.

1.1. The PIC collision operator

The PIC algorithm (Birdsall & Langdon 2018) solves the Klimontovich equation (Klimontovich 1967) for finite-size particles,¹ coupled with the Maxwell system of equations (or a subset) to self-consistently model the fields generated by the movement of the particles and advance their position over time. Since particles have a finite size, the electrostatic force between two particles is reduced at short distances (comparable to the particle radius), which significantly reduces the short-range collisional interactions compared with point-like particles; we note that the long-range interactions are correctly captured. Hence, the effective Coulomb collision operator for PIC-simulated plasmas is modified relative to that of real plasmas.

Theoretical work on the collision operator for plasmas composed of finite-size particles was originally made by Langdon (1970), Langdon & Birdsall (1970) and Okuda & Birdsall (1970). Langdon (1970) derived the first description of a FP collision operator for electrostatic PIC simulations, which includes the effect of aliasing due to the finite grid size, discrete time step and shape function, among others. More recently, Touati *et al.* (2022) thoroughly re-derived these results and extended the description to include the effect of varying macroparticle weights.

Several numerical experiments were conducted to complement and verify some predictions using electrostatic PIC codes. Hockney (1971) empirically studied collisional relaxation rates and inferred a scaling law for two-dimensional (2-D) thermalisation times for the nearest grid point (zeroth-order interpolation) and

¹Also referred to as macroparticles or superparticles in the literature.

cloud-in-cell (CIC, first-order interpolation) shape functions. Matsuda & Okuda (1975) performed direct comparisons between the predicted advection for Gaussian-shaped particles moving at $v = v_{th} = \sqrt{k_B T/m}$ (where k_B is the Boltzmann constant, T the species temperature, and m the species mass) and diffusion for particles at rest $v = 0$ for both 2-D and 3-D simulations. They found that for ions the theoretical predictions were accurate within a 20 % error, while for electrons the predictions could deviate from the result up to a factor of 200 %. They also demonstrated that, when imposing a strong external magnetic field, the collisionality of the simulated plasma decreases significantly if the gyroradius is smaller than the particle size. More recently, Gatsonis & Spirkin (2009) and Averkin & Gatsonis (2018) studied deflection and slow-down times in 3-D PIC codes with unstructured Delaunay–Voronoi grids, and Jubin *et al.* (2024) studied numerical relaxation rates for 2-D low-temperature plasma set-ups using CIC shape functions while proposing mitigation strategies to reduce undesired thermalisation effects due to finite-size particle collisions. Finally, Park, Moore & Baalrud (2025) complemented the work of Jubin *et al.* (2024) by thoroughly exploring 1-D, 2-D and 3-D scenarios.

In this work, we extend these previous results by providing a thorough comparison between predicted and measured advection and diffusion coefficients for a broad range of particle velocities ($v_{x,y}/v_{th} \in [0, 5]$) and simulation parameters (shape functions, grid resolution, particles per cell, thermal velocity). Furthermore, we conduct these tests for the first time using an electromagnetic PIC code (Fonseca *et al.* 2002). We note that the theory of collisions in plasmas is based purely on electrostatic interactions, so this work provides an important first step on possible modifications of the collisional operators when including self-consistent electromagnetic interactions. For all scenarios, we consider the 2-D version of OSIRIS (Fonseca *et al.* 2002) that utilises a finite-difference time-domain (FDTD) Yee solver (Yee 1966), and a plasma consisting of a single mobile electron species moving over a uniform neutralising ion background.

Under these conditions, and when neglecting the effect of temporal and spatial aliasing, the theoretical advection $\mathbf{A}(\mathbf{v})$ and diffusion $\mathbf{D}(\mathbf{v})$ operators for the finite-size e–e collisions are given (in S.I. units) by (cf. derivation in Supplemental Material S1)

$$\mathbf{A}(\mathbf{v}) = \mathbf{A}^{(1)}(\mathbf{v}) + \mathbf{A}^{(2)}(\mathbf{v}), \quad (1.2)$$

$$\mathbf{A}^{(1)}(\mathbf{v}) = -\frac{\omega_p v_{th}}{N_{Dmac} 2^{5/2} \pi^{3/2}} \int_{\mathbb{R}^2} d\tilde{\mathbf{k}} \frac{S_\rho^m(\tilde{\mathbf{k}})^4}{\left| \epsilon(\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}}, \tilde{\mathbf{k}}) \right|^2 \tilde{k}_s^4} \frac{\tilde{\mathbf{k}}_s (\tilde{\mathbf{k}}_s \cdot \tilde{\mathbf{k}}) (\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}})}{\tilde{k}^3} e^{-(\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}})^2 / 2\tilde{k}^2}, \quad (1.3)$$

$$\mathbf{A}^{(2)}(\mathbf{v}) = \frac{1}{2v_{th}} \nabla_{\tilde{\mathbf{v}}} \cdot \mathbf{D}(\mathbf{v}), \quad (1.4)$$

$$\mathbf{D}(\mathbf{v}) = \frac{\omega_p v_{th}^2}{N_{Dmac} 2^{3/2} \pi^{3/2}} \int_{\mathbb{R}^2} d\tilde{\mathbf{k}} \frac{S_\rho^m(\tilde{\mathbf{k}})^4}{\left| \epsilon(\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}}, \tilde{\mathbf{k}}) \right|^2 \tilde{k}_s^4} \frac{(\tilde{\mathbf{k}}_s \otimes \tilde{\mathbf{k}}_s)}{\tilde{k}} e^{-(\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}})^2 / 2\tilde{k}^2}, \quad (1.5)$$

where the pre-factor constants correspond to the electron plasma frequency $\omega_p = \sqrt{n_e e^2 / \epsilon_0 m_e}$ (with n_e being the electron number density, e the electron charge, ϵ_0 the vacuum permittivity and m_e electron mass), and the number of macroparticles per Debye square $N_{Dmac} = n_{mac} \lambda_D^2$ where $\lambda_D = v_{th} / \omega_p$ is the electron Debye length and $n_{mac} = N_{ppc} / (\Delta_x \Delta_y)$ is the macroparticle number density (with N_{ppc} being the number of particles per cell and $\Delta_{x,y}$ the grid resolution). The integral

is defined over $\mathbb{R}^2$ and all vector quantities have two directions (e.g. $\tilde{\mathbf{v}} = (\tilde{v}_x, \tilde{v}_y)$, $d\tilde{\mathbf{k}} = d\tilde{k}_x d\tilde{k}_y$); $\tilde{\mathbf{v}} = \mathbf{v}/v_{th}$ is the particle velocity normalised to the thermal velocity, $\tilde{\mathbf{k}} = \mathbf{k}\lambda_D$ is the wave vector normalised by the inverse of the electron Debye length and $\tilde{\mathbf{k}}_s = \mathbf{k}_s\lambda_D$ is a dimensionless vector associated with the FDTD field solver used where $\mathbf{k}_s = (k_x \text{sinc}(k_x \Delta_x/2), k_y \text{sinc}(k_y \Delta_y/2))$ with $\text{sinc}(x) = \sin(x)/x$. Finally, $S_\rho^m(\tilde{\mathbf{k}}) = \text{sinc}(\tilde{k}_x \Delta_x/2\lambda_D)^{m+1} \text{sinc}(\tilde{k}_y \Delta_y/2\lambda_D)^{m+1}$ is the charge deposition shape function of order m , and ϵ the dielectric function defined as (cf. derivation in Supplemental Material S1)

$$\epsilon(\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}}, \tilde{\mathbf{k}}) = 1 - \frac{S_\rho^m(\tilde{\mathbf{k}})^2 (\tilde{\mathbf{k}}_s \cdot \tilde{\mathbf{k}})}{2\tilde{k}_s^2 \tilde{k}^2} Z' \left(\frac{\tilde{\mathbf{k}} \cdot \tilde{\mathbf{v}}}{\sqrt{2}\tilde{k}} \right), \quad (1.6)$$

where $Z'(\xi) = \pi^{-1/2} \int_{\mathbb{R}} dt e^{-t^2}/(\xi - t)^2$ is the first derivative of the plasma dispersion function (Fried & Conte 1961). A more in-depth discussion regarding the derivation of the advection and diffusion coefficients (based on the more general form of the PIC collision operator derived by Touati *et al.* (2022)) is provided in Supplementary Material S1.

1.2. Related machine learning work

The machine learning community has proposed several approaches to learning operators that recover observed dynamical data (a type of inverse problem). Common neural network (NN) based approaches include neural operators such as deep operator networks (Lu, Jin & Karniadakis 2019) and Fourier neural operators (Li *et al.* 2020), among many others (Kovachki *et al.* 2023) designed to learn mappings between function spaces. While these approaches are extremely flexible, since the operator form is not hard constrained, they require substantial amounts of training data to obtain an accurate general operator. Furthermore, the learned operator is, by construction, not interpretable. On the other hand, physics-informed NNs (PINNs) (Raissi, Perdikaris & Karniadakis 2019) or discrete grid-based alternatives (Karnakov, Litvinov & Koumoutsakos 2022), can also be used to extract operators at a significantly reduced data cost. However, they require the underlying form of the operator to be defined *a priori*, and the full solution of the differential equation must also be learned during the fitting process. A middle ground between traditional numerical solvers and data-driven methods is found in universal differentiable equations (UDEs) (Rackauckas *et al.* 2020), where learnable models (e.g. NNs) are integrated within a prescribed differentiable simulator. Therefore, UDEs combine the benefits of previous extensive research on traditional numerical solvers with the flexibility of data-driven approaches. This is an important combination for problems where we want to strongly enforce known physical biases and numerical constraints, and are not interested in creating a surrogate model for the differential equation solution (since the cost of integrating the dynamics with the simulator is not significant). Our work belongs to this family of algorithms: we combine a differentiable FP solver with learnable advection/diffusion operators. Interpretable symbolic regression methods can also be used for this task (Brunton, Proctor & Kutz 2016; Rudy *et al.* 2017; Gurevich *et al.* 2024). However, to correctly recover the operator, its expression must be a linear combination of the library of pre-defined terms. This is a problem in situations where we do not have an intuition on what the underlying form of the operator is, or how its coefficients should be parameterised, since there exist infinite possible combinations of input variables that can be added to the library.

The majority of recent work at the intersection of machine learning and collision operators for plasma physics simulations has focused on reducing the associated computational cost of evaluating the collision operator using surrogate models. For example, surrogate models were trained for the Landau–Fokker–Planck (Miller *et al.* 2021; Lee *et al.* 2023; Noh, Lee & Yoon 2025), Rosenbluth–Fokker–Planck (Chung *et al.* 2023) and Boltzmann (Holloway, Wood & Alekseenko 2021; Xiao & Frank 2021, 2023; Miller *et al.* 2022; Lee *et al.* 2024, 2025) collision operators. Unlike our work, the underlying simulation data used for training all these models were explicitly solving the prescribed collision operator. The goal of these approaches was to reduce the computational cost of evaluating the operator, not to inform the discovery of a new model (the goal of the present work).

PINNs have also been extensively used to model equations that contain FP-type operators. This includes works on the Vlasov–Fokker–Planck (Hwang *et al.* 2020), the Vlasov–Poisson–Fokker–Planck (Lee, Jang & Hwang 2021), the Landau–Fokker–Planck (Chung *et al.* 2023) and the FP (Xu *et al.* 2020; Chen *et al.* 2021; Zhai, Dobson & Li 2022; Zhang & Yuen 2022; Wang *et al.* 2025) equations. Similarly, PINNs have been used to solve the Boltzmann equation for weakly ionised plasmas (Kawaguchi & Murakami 2022; Zhong, Wu & Wang 2022), the simplified Boltzmann–Bhatnagar–Gross–Krook equation (Lou, Meng & Karniadakis 2021; Li *et al.* 2024; Oh *et al.* 2025) and relativistic FP equations describing the runaway electron dynamics in magnetic confinement fusion devices (McDevitt 2023; McDevitt *et al.* 2025a,b).

We emphasise that, with the exception of Chen *et al.* (2021), all previous examples focus on solving the equation of interest (a forward problem) using a prescribed collision operator (with fixed, non-learnable coefficients) rather than on extracting a new operator from the observed dynamics (an inverse problem), as we aim in this paper. While Chen *et al.* (2021) proposed using PINNs to simultaneously address the forward and inverse problems, the simulation data in that work were generated by explicitly solving the equation with the learned collision operator, whereas in our work the data are obtained from first-principles simulations, that capture the electromagnetic interactions between particles self-consistently.

An exception to this line of research is the work of Camporeale *et al.* (2022), who used PINNs to learn FP operators that capture the observed electron dynamics in the Earth’s radiation belts. The underlying motivation and method are the same as ours, with the exception that we use a differentiable simulator to integrate the dynamics. We believe our approach to be more advantageous for our case study since we can generate as much kinetic simulation data as required, and the differentiable simulator allows us to quickly extract the collision operator without having to train a PINN, as in Camporeale *et al.* (2022). Additionally, in this work, we highlight the reason why the PINNs in Camporeale *et al.* (2022) do not find a unique solution to the advection/diffusion operators and propose an approach to mitigate this problem.

Another very promising related work is that of Zhao & Lei (2025) and Zhao *et al.* (2025), which recently proposed an approach to learn collision operators from molecular dynamics (MD) data. Similarly to our work, they infer a collision operator from kinetic data, with the operator written in its integral formulation. They demonstrate the capability of this formulation to accurately capture the results of MD simulations for both strongly coupled and weakly correlated plasmas initially outside thermal equilibrium. Unlike our approach here, they still use particle tracks during training while we learn directly from phase space data. Our approach, as

previously explained, is more efficient in terms of storage requirements and can be more easily generalisable to other datasets. Furthermore, while their approach minimises the prediction error over one time step, ours optimises for long-term stability since we use a differentiable simulator to integrate the dynamics over multiple time steps. Lastly, while the formulation of the operator proposed in Zhao *et al.* (2025) is tuned specifically for collisional dynamics, the FP description proposed in this work is applicable to other processes.

Differentiable simulators have been previously used in plasma physics to tackle distinct problems. Examples include the discovery of long-lived plasma wave packets (Joglekar & Thomas 2022), learning a fluid closure for nonlinear Landau damping (Joglekar & Thomas 2023), mitigating laser–plasma instabilities (Joglekar 2024) and simulating the kinetic dynamics of a 1-D plasma with arbitrary degrees of collisionality (Carvalho, Ferreira & Silva 2024). While it can be argued that a collisional operator was being learned in Carvalho *et al.* (2024), the main motivation of the work was reducing the run time of the simulator. Furthermore, the graph NN model used only captures collisions between a reduced number of neighbouring electrons and does not provide an interpretable description as presented in this work.

2. Methods

Our aim is to extract advection and diffusion operators that are capable of reproducing the collisional dynamics of a plasma composed of finite-size particles. Based on the generated simulated data, we will use two different methods to estimate the coefficients. The first method makes use of particle tracks and is the standard approach from previous literature (Hockney 1971; Matsuda & Okuda 1975). The second is the novel contribution of the work, and consists of using only the distribution function (phase space) evolution of subpopulations to estimate the operator using a differentiable simulator framework. We will examine the benefits/tradeoffs between the two approaches later in this work.

2.1. Particle-in-cell simulations

Throughout this work, we perform 2-D simulations using the electromagnetic PIC code OSIRIS (Fonseca *et al.* 2002). We consider a single mobile electron species moving over a fixed uniform ion background inside a periodic box. In these simulations, only electron–electron collisions are captured. The initial electron velocities are sampled from a thermal distribution with zero net velocity and standard deviation v_{th} in all directions, and the simulation window is set to $10^3 \lambda_D \times 10^3 \lambda_D$. For all scenarios, we use a charge-conserving current deposition scheme (Esirkepov 2001), a second-order FDTD electromagnetic field solver (Yee 1966) and the standard Boris pusher (Boris & Shanny 1972) to advance particle momenta and position.

To generate a dataset of simulations with different advection and diffusion operators, we vary four parameters: the number of particles per cell (N_{ppc}), the current deposition shape function (S_j^m , where m is the interpolation order), the ratio of the grid-resolution over the electron Debye length ($\Delta_x/\lambda_D = \Delta_y/\lambda_D$) and the electron thermal velocity (v_{th}/c). In total, we performed approximately 100 simulations corresponding to different parameter combinations and preserved a subset of approximately 75 simulations for which no significant PIC heating was observed.²

²We preserved all simulations for which the total energy variation ratio threshold $\Delta E/E < 10^{-2}$ was not exceeded. We discarded simulations above this threshold because in this work we are only focusing on learning FP operators that do not change over time.

The detailed list of simulation parameters used and the heating tests performed is provided in Supplementary Material S2.

For each simulation, we stored the raw data of all macroparticles (approximately 10^6 – 10^7 macroparticles depending on the numerical parameters, we store both their position and momenta) at equally spaced time intervals to collect a minimum of 100 snapshots over the full simulation. This information is used to construct the phase space information of different subpopulations in post-processing, as described in Supplementary Material S3. We note that we only store the data of all macroparticles to be able to compare the operators recovered from the two methods: from macroparticle particle tracks (i.e. macroparticle position and momenta history), and from the phase space evolution of subpopulations. In future works, this memory-intensive step will not be required.

2.2. Advection–diffusion coefficients from particle tracks

The advection and diffusion coefficients for particles moving at a specific velocity $\mathbf{v}$ can be estimated by tracking the rate of the average velocity drift and spread over time

$$A_i(\mathbf{v}) = \frac{\langle \Delta v_i \rangle_{\mathbf{v}}}{\Delta t}, \quad (2.1)$$

$$D_{ij}(\mathbf{v}) = \frac{\langle \Delta v_i \Delta v_j \rangle_{\mathbf{v}}}{\Delta t}, \quad (2.2)$$

where $\Delta v_i = v_i(t_0 + \Delta t) - v_i(t_0)$ corresponds to the change in the velocity along the i -axis of a macroparticle with $\mathbf{v}(t_0) = \mathbf{v}$ over a time interval Δt , and $\langle \cdot \rangle_{\mathbf{v}}$ corresponds to an average over all macroparticles with the selected initial velocity.

In practice, we relax the above condition to consider a finite region of phase space over which the average is computed (figure 1). In this work, we divide the $\mathcal{V} : [-5, 5]v_{th} \times [-5, 5]v_{th}$ phase space into equal-sized bins resulting in a $N_v \times N_v$ grid with $N_v = 51$. To increase the statistics, we divide the simulation into contiguous, equally sized time intervals so that the average is computed across particles trajectories starting at the desired region of phase space at the beginning of any of the time intervals ($t_0 = \{k\Delta t : k \in \{0, 1, \dots, t_{max}/\Delta t\}\}$). Equations (2.1) and (2.2) are then rewritten as

$$A_i(\mathbf{v}_{bin}) = \frac{\langle \Delta v_i \rangle_{\mathbf{v}(t_0) \in \mathbf{v}_{bin}}}{\Delta t}, \quad (2.3)$$

$$D_{ij}(\mathbf{v}_{bin}) = \frac{\langle \Delta v_i \Delta v_j \rangle_{\mathbf{v}(t_0) \in \mathbf{v}_{bin}}}{\Delta t}, \quad (2.4)$$

where $\mathbf{v}_{bin}$ represents all velocities within the phase space bin and $\langle \cdot \rangle_{\mathbf{v}(t_0) \in \mathbf{v}_{bin}}$ corresponds to an average over all macroparticle trajectories starting inside the bin at the beginning of any time interval.

The time interval over which statistics are computed affects the retrieved values of the advection–diffusion coefficients. If Δt is too small, the particle trajectories are correlated, and the Markovian assumption underlying the FP derivation breaks down. If Δt is too large, the evolution of $\langle \Delta v_i \rangle$ and $\langle \Delta v_i \Delta v_j \rangle$ stops being linear (see figure 1). Note that these time scales are simulation dependent since they are defined by the collisional time scales (which we want to infer) and collective plasma processes. To tackle this problem, we then devised and automatised process where,

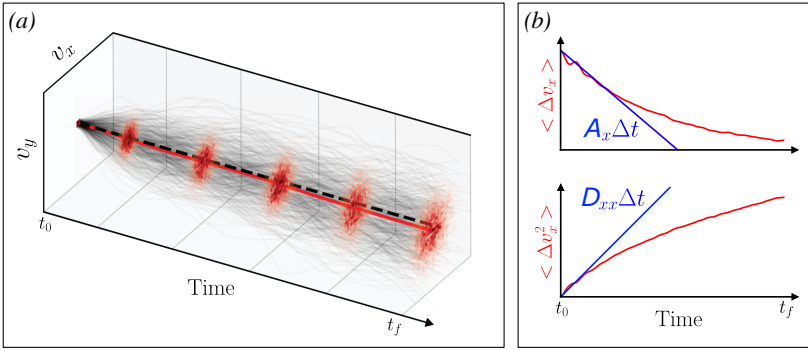


FIGURE 1. Illustration of how advection and diffusion coefficients can be inferred from 2-D particle tracks. (a) By following a group of particles with similar initial velocities over time, we observe that advection leads to an average velocity drift ($\langle \Delta v_i \rangle$), while diffusion leads to an increased spread of the distribution ($\langle \Delta v_i \Delta v_j \rangle$). Drift is visible by noting that the average particle velocity (red line) is changing when compared with the initial velocity (dashed black line). Diffusion is clearly illustrated by the increased distribution spread (highlighted via the phase space histograms). (b) In practice, the advection–diffusion values are estimated by measuring the rate of change of $\langle \Delta v_i \rangle$ and $\langle \Delta v_i \Delta v_j \rangle$ with respect to a period of time where the evolution is linear. To obtain an accurate estimate of the coefficients it is crucial that statistics are computed during the linear evolution phase.

for each simulation: (i) we compute the coefficients using different values of Δt ; (ii) for each value of Δt , we evaluate the corresponding advection–diffusion coefficients accuracy in reproducing the phase space dynamics of different subpopulations of particles; and (iii) we pick the best performing coefficients to compare against the machine learning based approaches. A broader discussion on the consequences of estimating the advection–diffusion coefficients with this technique, is provided in Supplementary Material S4.

2.3. Advection–diffusion coefficients from phase space evolution

Determining the advection and diffusion operators that describe the phase space dynamics can be written as an optimisation task

$$\min_{\mathbf{A}, \mathbf{D}} \sum_u \mathcal{L}(\hat{f}^{(t+u)}(\mathbf{A}, \mathbf{D}, f^{(t)}), f^{(t+u)}), \quad (2.5)$$

where $f^{(t+u)} \in \mathbb{R}^{N_v} \times \mathbb{R}^{N_v}$ corresponds to the true distribution function at time step $t + u$, $\hat{f}^{(t+u)} \in \mathbb{R}^{N_v} \times \mathbb{R}^{N_v}$ the predicted distribution function when evolving the phase space dynamics of $f^{(t)}$ over u time steps using the estimated $\mathbf{A}$, $\mathbf{D}$ operators and $\mathcal{L}$ is an error (loss) function computed over the full phase space (e.g. mean absolute error).

For a given single ‘trajectory’ of f , this is an ill-posed inverse problem, since there is a family of solutions that results in equivalent dynamics for a given initial distribution function. This is caused by the presence of velocity space gradients in (1.1), which allows perturbations to the advection/diffusion coefficients to cancel each other (cf. discussion in Supplementary Material S5). This non-uniqueness issue has been observed, e.g. in previous work by Camporeale *et al.* (2022), where PINNs initialised with different random seeds recover significantly different advection–diffusion operators that describe the phase space dynamics with similar levels of

accuracy. Furthermore, to determine the coefficients in a particular region of phase space, it is necessary to have access to data in that region, i.e. $\partial f/\partial t \neq 0$ at some time step.³

To address these two challenges, we propose to employ the phase space evolution of different subpopulations. Ideally, these subpopulations are sampled at initialisation and cover different regions of the phase space. Furthermore, since we want operators that are stable in the long-term prediction of the dynamics, we optimise for the evolution of the distribution function over multiple time steps; this is a common strategy known as temporal unrolling (see e.g. Brandstetter, Worrall & Welling 2022; List *et al.* 2025). Both these strategies serve the same underlying purpose: to generate data that can constrain the space of possible solutions. The optimisation task is then rewritten as

$$\min_{\mathbf{A}, \mathbf{D}} \sum_s^{N_s} \sum_t^{N_t - N_u} \sum_u^{N_u} \mathcal{L}(\hat{f}_s^{(t+u)}(\mathbf{A}, \mathbf{D}, f_s^{(t)}) - f_s^{(t+u)}), \quad (2.6)$$

where f_s corresponds to the distribution function of subpopulation s , and we make explicit the sum over N_s subpopulations, N_t simulation time steps and N_u time steps into the future for which we generate a prediction $\hat{f}^{(t+u)}$; we are softly enforcing the physics via the loss function, similarly to PINNs or equivalent algorithms, with the difference that these methods minimise the theoretical partial differential equation (PDE) residual evaluated on the predicted solution at predefined spatio-temporal positions while here we are minimising the error between PDE trajectories.

We solve the optimisation task using a differentiable simulator coupled with a gradient-based optimisation method (figure 2). We implement the full differentiable simulator and its core 2-D FP solver in PyTorch (Ansel *et al.* 2024), which we use to compute the forward dynamics of different subpopulations, and we use the Adam optimiser (Kingma 2014) to update the advection and diffusion operators such that the difference between the predicted and real distribution function evolutions is minimised. To facilitate experiment tracking and performance comparisons over a large range of training and test set-ups we make use of MLflow (Zaharia *et al.* 2018).

Using this methodology, the advection and diffusion operators can be parameterised using a discrete or continuous function. The discrete case is equivalent to computing coefficients from particle tracks (2.2), i.e. we discretise the operators over a fixed grid $\mathbf{A}_i \in \mathbb{R}^{N_v \times N_v}$ and $\mathbf{D}_{ij} \in \mathbb{R}^{N_v \times N_v}$. A continuous operator can be obtained by instead using a smooth function approximation. In our case, we use NN models, i.e. $\mathbf{A}_i(\mathbf{v}) = \text{NN}_{\mathbf{A}_i}(\mathbf{v})$ and $\mathbf{D}_{ij}(\mathbf{v}) = \text{NN}_{\mathbf{D}_{ij}}(\mathbf{v})$. Furthermore, a continuous description allows us to extract more general operators that might depend on other system parameters, since these can be provided as inputs to the NN. In our case study, these will correspond to numerical simulation parameters, e.g. $\mathbf{A}_i(\mathbf{v}) = \text{NN}_{\mathbf{A}_i}(\mathbf{v}, N_{ppc}, \Delta x/\lambda_D, m, v_{th}/c)$. For other cases, it might be more relevant to include meaningful physical parameters or even the simulation time itself if the coefficients are expected to change over time.

It is possible to guarantee known constraints in the advection and diffusion operators, for both the discrete and continuous cases, by enforcing known symmetries. We will elaborate further on this topic when comparing the performance of the different approaches in later sections of the manuscript. More details about the symmetries and the overall FP solver implementation are also provided in Supplementary Material S6.

³For a thermal distribution, $\partial f/\partial t \approx 0$ everywhere, so no dynamics are observed in phase space.

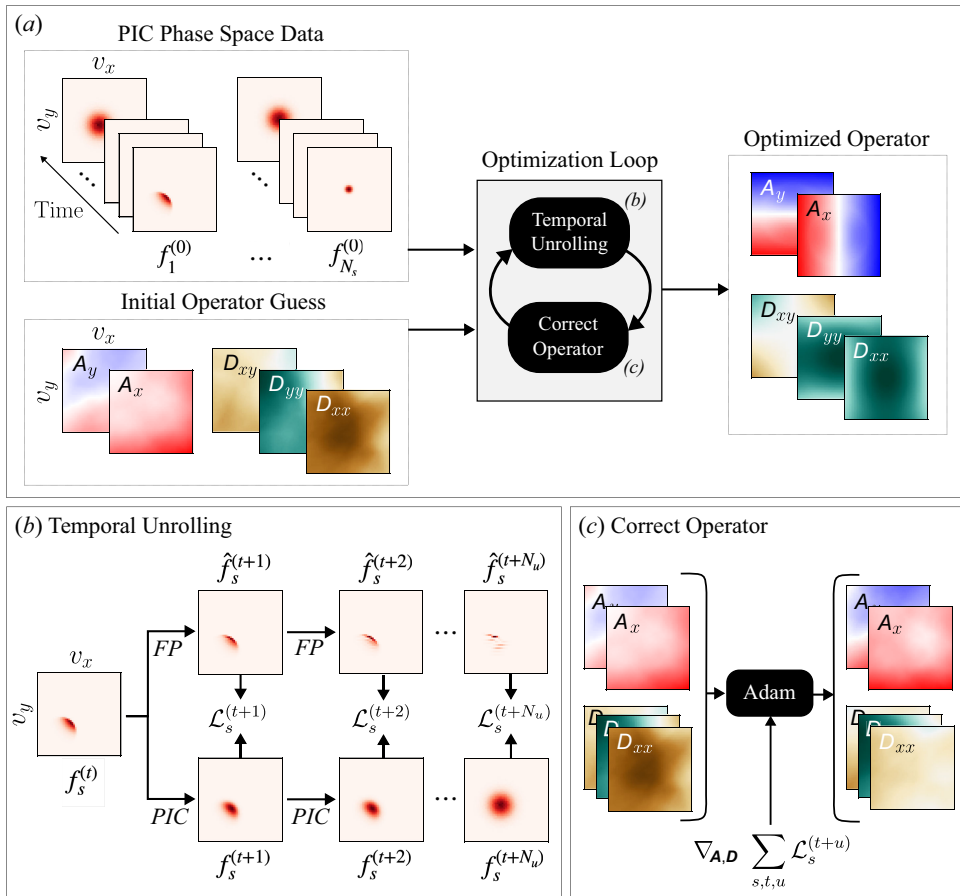


FIGURE 2. (a) Illustration of how advection and diffusion coefficients are inferred from the 2-D phase space evolution of N_s subpopulations using a differentiable FP solver. The evolution of the phase space and the changes in the operator are exaggerated for visualisation purposes. (b) Using the FP solver and the current operator state, we advance the phase space over N_u time steps and compare against the PIC results via a scalar error metric $\mathcal{L}$. This operation is performed over all training subpopulations $s \in [1, N_s]$ and initial time steps $t \in [0, N_t - N_u]$. (c) We then update the operator using the gradient-based optimiser Adam (Kingma 2014), to minimise the unrolled error across all subpopulations and time steps. The two operations are performed sequentially in a loop until the results have converged.

3. Results

We now evaluate the performance of the advection and diffusion operators extracted using different methods. They are named accordingly: Tracks – computed using particle tracks; PS-Tensor – computed using the phase space evolution and a discrete operator description; PS-NN – computed using phase space evolution and a continuous operator parameterised by a NN; PS-NN-Multi – similar to PS-NN but in this case we trained a single NN model on multiple simulations at once (i.e. predicted coefficients are conditioned on simulation parameters, $A_i(\mathbf{v}) = \text{NN}_{A_i}(\mathbf{v}, N_{ppc}, \Delta_x/\lambda_D, m, v_{th}/c)$ and $D_{ij}(\mathbf{v}) = \text{NN}_{D_{ij}}(\mathbf{v}, N_{ppc}, \Delta_x/\lambda_D, m, v_{th}/c)$ with simulation parameters pre-processed as detailed in Supplementary Material S6).

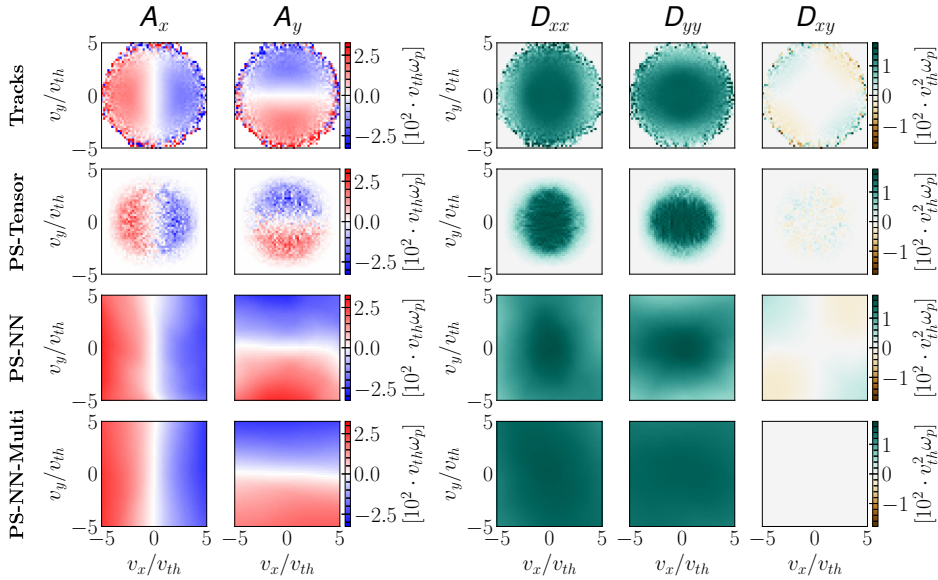


FIGURE 3. Advection and diffusion coefficients retrieved with different approaches for simulation $index = 0$ ($N_{ppc} = 4$, $m = 1$, $\Delta_x/\lambda_D = 1$, $v_{th} = 0.01c$). Track operator is computed from statistics over all macroparticles, PS operators are computed from the phase space evolution of 9 subpopulations using a differentiable FP solver and a discrete (PS-Tensor) or continuous (PS-NN, PS-NN-Multi) approximator. Note that all these operators perform similarly when predicting the phase space evolution of most tested subpopulations. More examples for other simulation parameters are provided in Supplementary Material S8.

The NN models used to parameterise the advection and diffusion operators ($NN_{A_i}(\cdot)$, $NN_{D_{ij}}(\cdot)$) are all multi-layer perceptrons with leaky rectified linear unit (Leaky ReLU) activation functions after each hidden layer. For the PS-NN method, we use 2 hidden layers, while for PS-NN-Multi we use 3 hidden layers. All hidden layers have 128 neurons. Further details on the optimisation loop (e.g. curriculum, learning rates, duration) can be found in Supplementary Material S7. For reference, extracting the operator from one simulation takes 5–10 min on a single Nvidia Titan X GPU using the default optimisation strategy without resorting to PyTorch Just-in-Time compilation capabilities.

To assess the generalisation potential of operators recovered from the phase space dynamics, we will make use of two distinct sets of populations: Train – 9 subpopulations (sampled from a normal distribution at different regions in phase space); Test – 19 subpopulations (distinct normal distributions, rings, quadrants, etc.). Illustrations of the different subpopulations are provided in figures S1 to S4 in Supplementary Material S3.

3.1. Example for a single simulation

To illustrate the overall aspect of the PIC collision operators recovered by the different methods, we start by analysing the results for a single simulation, simulation $index = 0$ ($N_{ppc} = 4$, $m = 1$, $\Delta_x/\lambda_D = 1$, $v_{th} = 0.01c$), which provides a good representation of the standard behaviour in our simulation dataset.

In figure 3, we show the operators retrieved using the different methods. Overall, we observe the expected behaviour. Advection A_i is positive for $v_i < 0$ and negative

for $v_i > 0$, revealing a drag that forces particles to $v_i = 0$. Diagonal diffusion terms D_{ii} are greater or equal to zero everywhere (otherwise the system would be unstable), and slower particles diffuse faster. Finally, we observe a correlation between diffusion along the x and y axes ($D_{xy} \neq 0$) for particles moving at an angle. The higher the velocity and the closer the angle is to 45° , the higher the cross diffusion term.

The operator obtained from particle tracks (Tracks) is smooth except for regions where statistics are poor (high v). This leads, in fact, to numerical instabilities if the operator is used as is during the time integration. To tackle this issue, we always use a smaller time step when later integrating the dynamics with these operators (we use $\Delta t = \Delta t_{dump}/10$ where Δt_{dump} is the phase space diagnostic period, with a broader discussion on other considered alternatives provided in Supplementary Material S6).

When we consider a discrete operator learned from the phase space dynamics (PS-Tensor), values are only defined within the phase space region where data were available during training. The retrieved advection–diffusion values are similar to the ones estimated from tracks, however, the overall behaviour is noisier. Unlike for operators learned from tracks, the noise level does not cause the integrator to become unstable. This is one benefit of learning the operator with the integrator in the loop.

Regarding the NN-based models (PS-NN, PS-NN-Multi), since they are continuous operators, they can extrapolate outside the training data region and bias the solutions to be smooth. However, for high v values, we can still clearly see the presence of artefacts (e.g. A_y is not zero at $v_y = 0$ and $v_x \simeq 5v_{th}$). Furthermore, while PS-NN seems to be able to capture more accurately the overall form of the operator (if we consider the Tracks operator as the correct reference), the operator trained on multiple simulations (PS-NN-Multi) seems to lack expressiveness on its diffusion term (D_{ii} values do not significantly decrease at high v and D_{ij} terms are not visible). We conjecture this happens due to the limited size of the latter NN model, whose expressivity could be increased by training on larger amounts of simulation data while increasing its size. Another option, which will be explored in later sub-sections, is enforcing further symmetries to facilitate the learning procedure.

To illustrate the capabilities of the previous operators to predict the long-term phase space dynamics, we show in figure 4 the relaxation of a ring subpopulation (part of the test set). The top panels showcase the dynamics retrieved from the PIC simulation. The bottom panels show the integration of the dynamics using the operators from figure 3 and the respective prediction error. The errors, per phase space bin, are below 5 % of the maximum distribution function value for the corresponding time step. Random distribution of errors at later times for PS-Tensor/PS-NN can be attributed to the granularity of the original distribution function due to the finite particle number. Systematic errors at later times for Tracks/PS-NN-Multi seem to indicate a slight underestimation of advection. Nonetheless, all models behave fairly well despite their different forms. In the following sub-section, we will show that this conclusion holds across many simulations and subpopulations using more systematic tests. Finally, additional examples of collision operators retrieved for other simulation parameters and the predicted evolution of different subpopulations are provided in Supplementary Material S8.

3.2. Long-term accuracy of different methods

To systematically measure the long-term accuracy of the operators retrieved using different methods, and for each subpopulation, we take the phase space at the initial

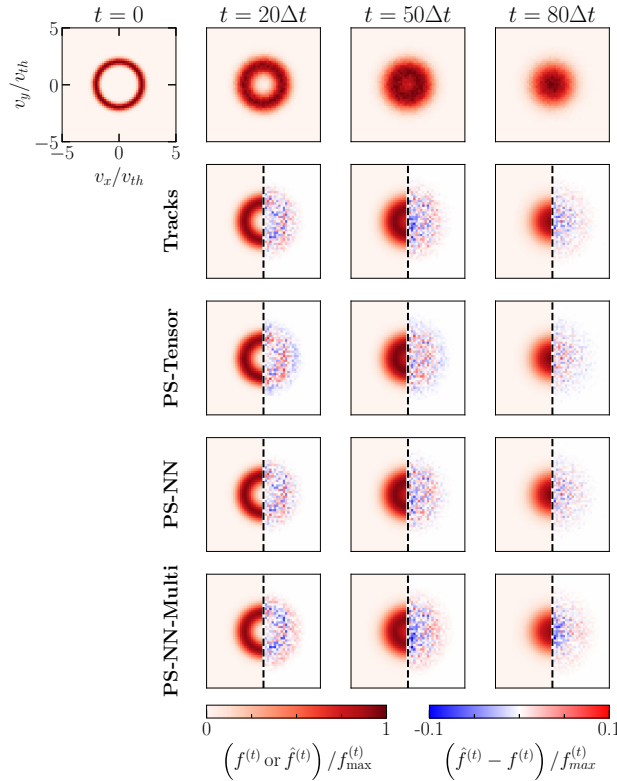


FIGURE 4. Phase space evolution for a ring subpopulation using operators recovered in [figure 3](#). This subpopulation was not used during the training of PS-Tensor, PS-NN and PS-NN-Multi models. The top row corresponds to the observed dynamics in the PIC simulation ($f^{(t)}$). The remaining rows represent the predicted phase space evolution on the left ($\hat{f}^{(t)}$ for $v_x/v_{th} \in [-5, 0]$) and the difference to the PIC data on the right ($\hat{f}^{(t)} - f^{(t)}$ for $v_x/v_{th} \in [0, 5]$). Values are normalised to the peak of the PIC distribution function at time t ($f_{max}^{(t)}$). All operators approximate the dynamics relatively well, and overall, the random distribution of errors can be attributed to the granularity of the original distribution function. However, both Tracks and PS-NN-Multi seem to systematically slightly underestimate advection (central blue error region and outer red halo). Examples for other subpopulations are provided in [Supplementary Material S8](#).

time step and use the FP solver to evolve it until the end of the simulation ($N_t \approx 100$) by applying the extracted operator. We refer to this process of simulating the phase space dynamics of an initial subpopulation from $t=0$ to $t=t_{max} = N_t \Delta t$ as performing a simulation rollout. For the operators retrieved using the differentiable simulator we use the same time step as the phase space diagnostic ($\Delta t = \Delta t_{dump}$) when integrating the dynamics over time. For operators obtained from particle tracks we reduce the integrator time step ($\Delta t = \Delta t_{dump}/10$) to avoid numerical instabilities caused by the noisy estimates of the coefficients at high v .

As an accuracy metric, we define the phase space mean absolute error (MAE) averaged over a simulation rollout (MAE-Rollout) as

$$\text{MAE-Rollout} = \frac{1}{N_t} \sum_t \left| \hat{f}^{(t)}(\mathbf{A}, \mathbf{D}, f^{(0)}) - f^{(t)} \right|, \quad (3.1)$$

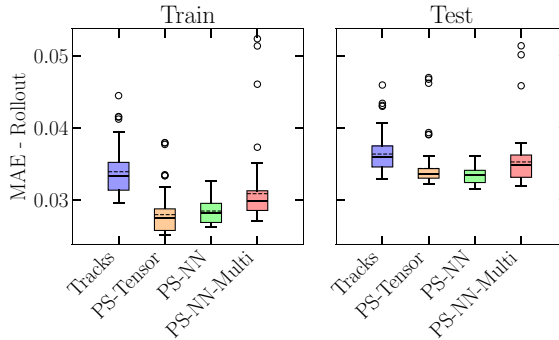


FIGURE 5. Distribution of rollout errors for advection and diffusion models obtained from particle tracks (Tracks) or phase space evolution of subpopulations (PS-Tensor, PS-NN, PS-NN-Multi). Boxplots represent statistics over the full dataset of PIC simulations (see table S1 for more details) averaged over initial subpopulations (Train – 9 subpopulations; Test – 19 subpopulations, more information in Supplementary Material S3). The mean error is shown with a dashed line, the median with a full line. The filled area corresponds to values between the first and third quartiles (Q_1 and Q_3). Whiskers represent the lowest values up to $Q_1 - 1.5(Q_3 - Q_1)$ and $Q_3 + 1.5(Q_3 - Q_1)$. Dots represent values outside this range. The performance of the different methods is, on average, equivalent on the test data. These results demonstrate that estimating the operators from phase space information using a differentiable simulator is a viable alternative approach to particle tracks.

where the sum over phase space dimensions is implicit. Since by definition $\|f^{(t)}\|_1 = 1$ the MAE-Rollout value can be interpreted as the average relative error over all time steps (e.g. MAE-Rollout = 0.03 implies that on average, we have a 3 % error across all time steps).

The statistics obtained over the full dataset of simulations for training and test subpopulations are shown in figure 5. We observe that methods based on the differentiable simulator perform on average better than the values estimated from the statistics on both the train and test subpopulations. Furthermore, PS-NN models trained on a single simulation seem to be more resilient to outliers than the alternative approaches. We attribute this to the enforcement of smooth coefficient values in regions where there is little statistics available, i.e. higher v . In fact, subpopulations that contain a larger number of particles at high v , which are mostly present in the test set and not the train set, are the reason why the error is slightly higher for test scenarios (even for operators extracted from particle tracks). We elaborate further on this topic in Supplementary Material S9.1.

Regarding the outliers, they are consistently associated with the same simulations, where a smaller number of macroparticles was used. This makes it harder both for the statistical method and the subpopulation approach to correctly capture the dynamics in the full phase space since there is a lack of statistics at high v . A more detailed analysis of these results is provided in Supplementary Material S9.2.

3.3. Impact of temporal unrolling during training

To justify a differentiable simulator approach to optimise the coefficients over longer prediction times, we compare, in figure 6, the rollout error of operators

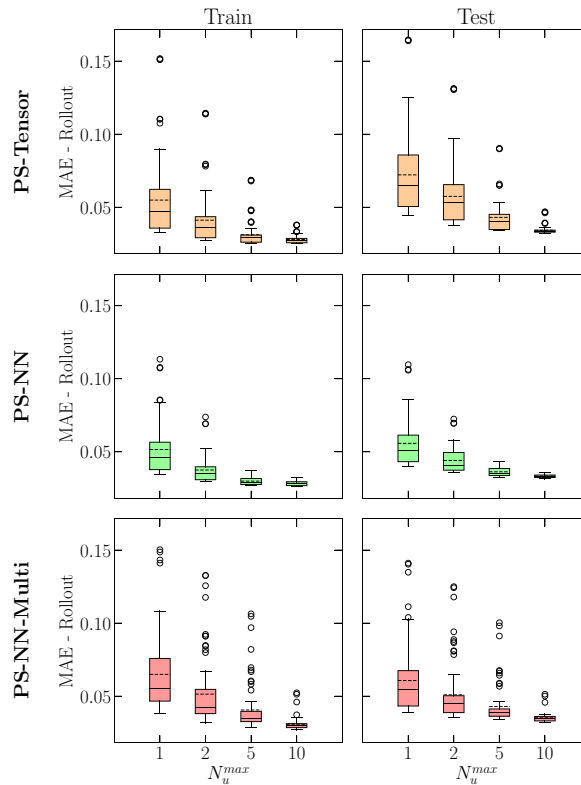


FIGURE 6. Impact of maximum temporal unroll length during training (N_u^{max}) on the long-term rollout error of different models. Rollout length is $N_t \approx 100$ across the full dataset of subpopulations and simulations. Larger temporal unrolling at train time consistently leads to improved rollout performance across all models, showcasing the importance of optimising the operators for long-term prediction.

extracted using different maximum training temporal unroll lengths N_u^{max} . It is clear that increasing the rollout duration during training significantly improves long-term performance across all models. We expect the optimal value of this rollout length to be problem-dependent, for instance, with respect to the noise/smoothness of the dynamics, relative time scales between the collisional dynamics and other processes, as well as the phase space diagnostic frequency. Future works should tune the rollout length to maximise long-term accuracy, bearing in mind that increasing this length leads to significant increases in training memory requirements and duration. In our case, we did not find meaningful improvements when using $N_u^{max} > 10$ and this is why we set this chosen maximum value.

To visualise the impact of the training temporal unrolling in the final operator extracted, we showcase an example for a PS-Tensor model in [figure 7](#). Increasing the temporal interval not only promotes smoother solutions but also allows us to capture the correct values for a larger range of the phase space. The impact is not as visually clear in NN-based models, for which we provide some examples in [Supplementary Material S10](#), but the results in [figure 6](#) demonstrate that indeed the operators are being meaningfully modified.

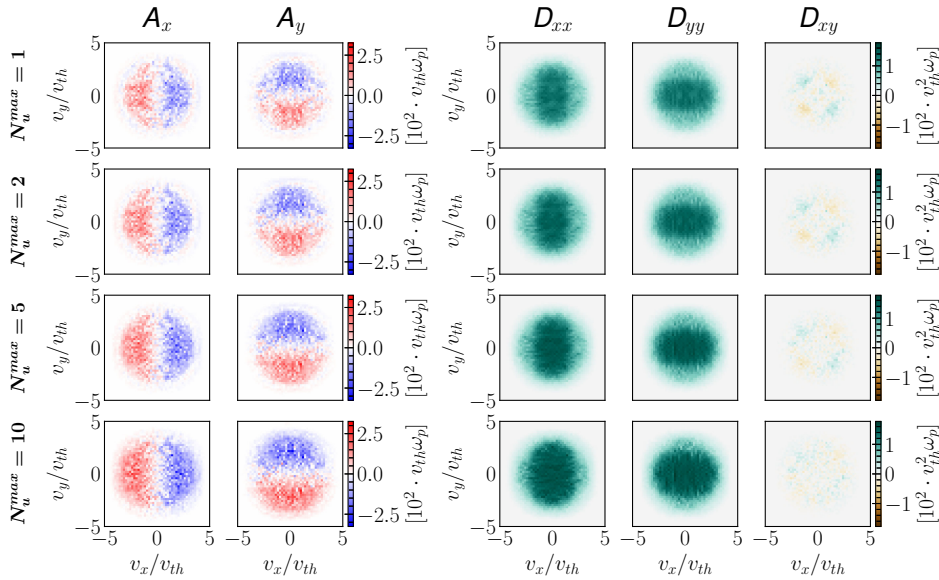


FIGURE 7. Illustration of the impact of maximum training temporal unroll length on PS-Tensor models. The example shown corresponds to simulation *index* = 0 ($N_{ppc} = 4$, $m = 1$, $\Delta_x/\lambda_D = 1$, $v_{th} = 0.01c$). It is clear that increasing N_u^{max} leads to a smoother operator defined over a larger region of the phase space while also significantly changing the average advection and diffusion values in regions of $v \approx 0$.

3.4. Enforcing known symmetries

Given known physical priors, such as symmetries, it is possible to enforce these into the learned advection–diffusion models. This is a common strategy in the physics-inspired machine learning literature which is particularly useful when handling inverse problems in the presence of sparse noisy data, such as the case under study in this work.

To test the impact of imposing such symmetries, we define a hierarchy of increasingly more restrictive models: AD – no symmetries imposed (same as before); AD_T – enforces x/y coefficients to be equivalent (i.e. $A_y(v_x, v_y) = A_x(v_y, v_x)$ and $D_{yy}(v_x, v_y) = D_{xx}(v_y, v_x)$); AD_{Sym} – extends AD_T to include (anti-)symmetry along $v_x = 0$ (i.e. $A_x(v_x, v_y) = -A_x(-v_x, v_y)$, $D_{xx}(v_x, v_y) = D_{xx}(-v_x, v_y)$ and $D_{xy}(v_x, v_y) = -D_{xy}(-v_x, v_y)$); $AD_{\parallel, \perp}$ – considers only parallel and perpendicular coefficients (i.e. $A_{\parallel}(v)$, $D_{\parallel}(v)$ and $D_{\perp}(v)$ which are projected to obtain A_i and D_{ij}). Further details on the implementation of these models are provided in Supplementary Material S6.

The rollout performance of the different symmetries is shown in figure 8. The impact on the training subpopulations is not significant. In fact, a slight increase in average error is observed for PS-Tensor and PS-NN models, while for PS-NN-Multi, we observe a slight decrease. On the other hand, it is clear that imposing symmetries leads to an improvement in performance for test subpopulations. We attribute this improvement to a more accurate estimate of the coefficients at larger values of v since the symmetries remove significant degrees of freedom, which artificially increases the statistics for estimating the coefficients in a given region of phase space. This leads to overall smoother operators. Notably, PS-Tensor (discrete) models end up performing slightly better than NN (continuous) models, which strengthens the argument that the smoothing bias imposed by the NN is no longer as important.

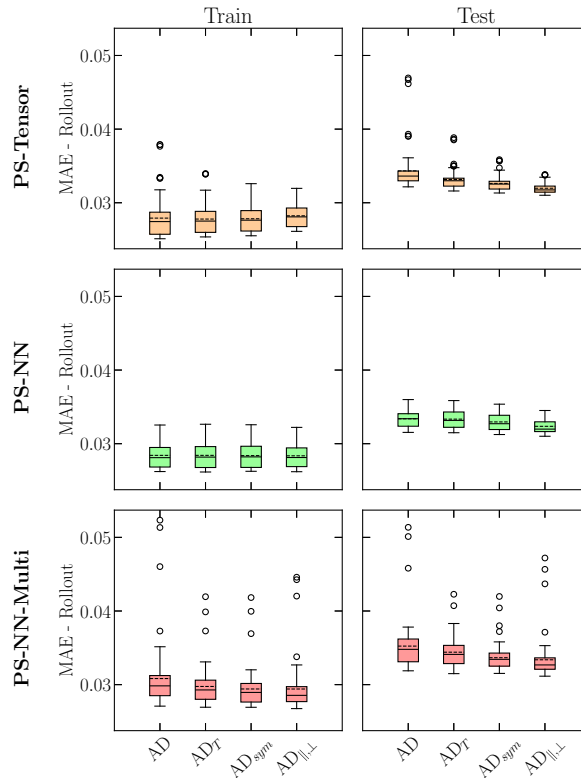


FIGURE 8. Impact of enforcing increasingly stricter symmetries on the rollout error. Introducing symmetries can lead to a slight increase in training error (removing some possible overfit), but consistently reduces the average test error and mitigates the appearance of outliers.

To illustrate these points, we provide in [figure 9](#) the impact of symmetries in a PS-Tensor operator. It is clear that, as we enforce stronger symmetries, a larger region of phase space is covered and smoother descriptions are recovered. However, due to the very limited train statistics at high v , artefacts (even if smooth) are still present (the thicker high advection regions and a small zero diffusion halo at high v). For NN-based approaches, we do not observe differences as striking as these when imposing more symmetries. However, it is still clear that symmetries improve results at high v , and PS-NN-Multi models can better capture the overall form of the operator. Additional examples are provided in Supplementary Material [S11](#).

The results shown in this subsection clearly highlight the benefits of enforcing known symmetries (which are problem dependent) for the task of retrieving collisional operators from sparse noisy phase space data. Enforcing stronger symmetries allows for the recovery of operators with less numerical artefacts and improves the operator generalisation to unseen dynamics.

3.5. Comparisons against theory

Finally, we compare the retrieved operators against theoretical predictions. To compute the theoretical curves, we numerically evaluate $\mathbf{A}$, $\mathbf{D}$ directly from (1.2) and (1.5). The derivative of the plasma dispersion function in (1.6) is evaluated

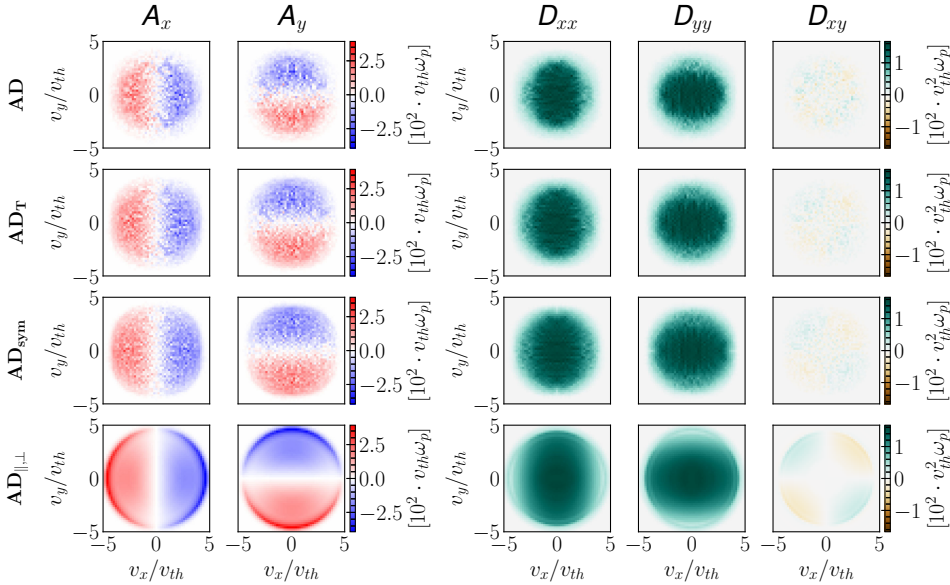


FIGURE 9. Impact of enforcing symmetries into Tensor models. Example shown corresponds to simulation $index = 0$ ($N_{ppc} = 4$, $m = 1$, $\Delta_x/\lambda_D = 1$, $v_{th} = 0.01c$). Enforcing symmetries allows us to recover smoother and more accurate coefficients for a larger phase space region. Artefacts are nonetheless present at high v regions since very limited statistics are available at train time.

using the implementation available in PlasmaPy (PlasmaPy Community *et al.* 2026). For all scenarios we consider that the test macroparticle moves with velocity $\mathbf{v} = v\hat{\mathbf{e}}_x$ (we do not observe any meaningful difference when setting the velocity at a different angle) such that $A_x(\mathbf{v}) = A_{\parallel}(v)$, $A_y(\mathbf{v}) = A_{\perp}(v) = 0$, $D_{xx}(\mathbf{v}) = D_{\parallel}(v)$, $D_{yy}(\mathbf{v}) = D_{\perp}(v)$, $D_{xy}(\mathbf{v}) = 0$.

A comparison of the theoretical curves against the results obtained from particle tracks and PS-Tensor for simulations with $N_{ppc} = 25$ and $v_{th} = 0.01c$ is provided in figure 10. We show only the region up to $v/v_{th} = 4$ since estimates beyond this value are not meaningful due to lack of statistics. Additionally, the PS-NN results are not included solely for readability purposes since they are equivalent to the PS-Tensor case.

Overall, we observe an excellent agreement between the theoretical predictions and the estimated values across different grid resolutions and shape functions. Differences are only visible for $A_{\parallel}$ when $v/v_{th} > 3$, a region where tracks measurements are also noisier and PS-Tensor shows some numerical artefacts at high v (see e.g. figure 9). Regarding both $D_{\parallel}$ and $D_{\perp}$, much better agreements are observed across all values of v .

The theoretical estimates in (1.2) and (1.5) state that the coefficients should be inversely proportional to $N_{Dmac} = N_{ppc}(\Delta_x/\lambda_D)^{-2}$. To verify this scaling, we plot in figure 11 the advection and diffusion coefficients with respect to N_{ppc} for varied grid resolutions, while fixing $m = 2$ and $v_{th} = 0.01c$ (for readability purposes since the conclusions are equivalent for other tested values). We provide the results for both PS-Tensor and PS-NN models to showcase slight differences in behaviour at high v . We do not make use of Tracks since the coefficients do not reproduce

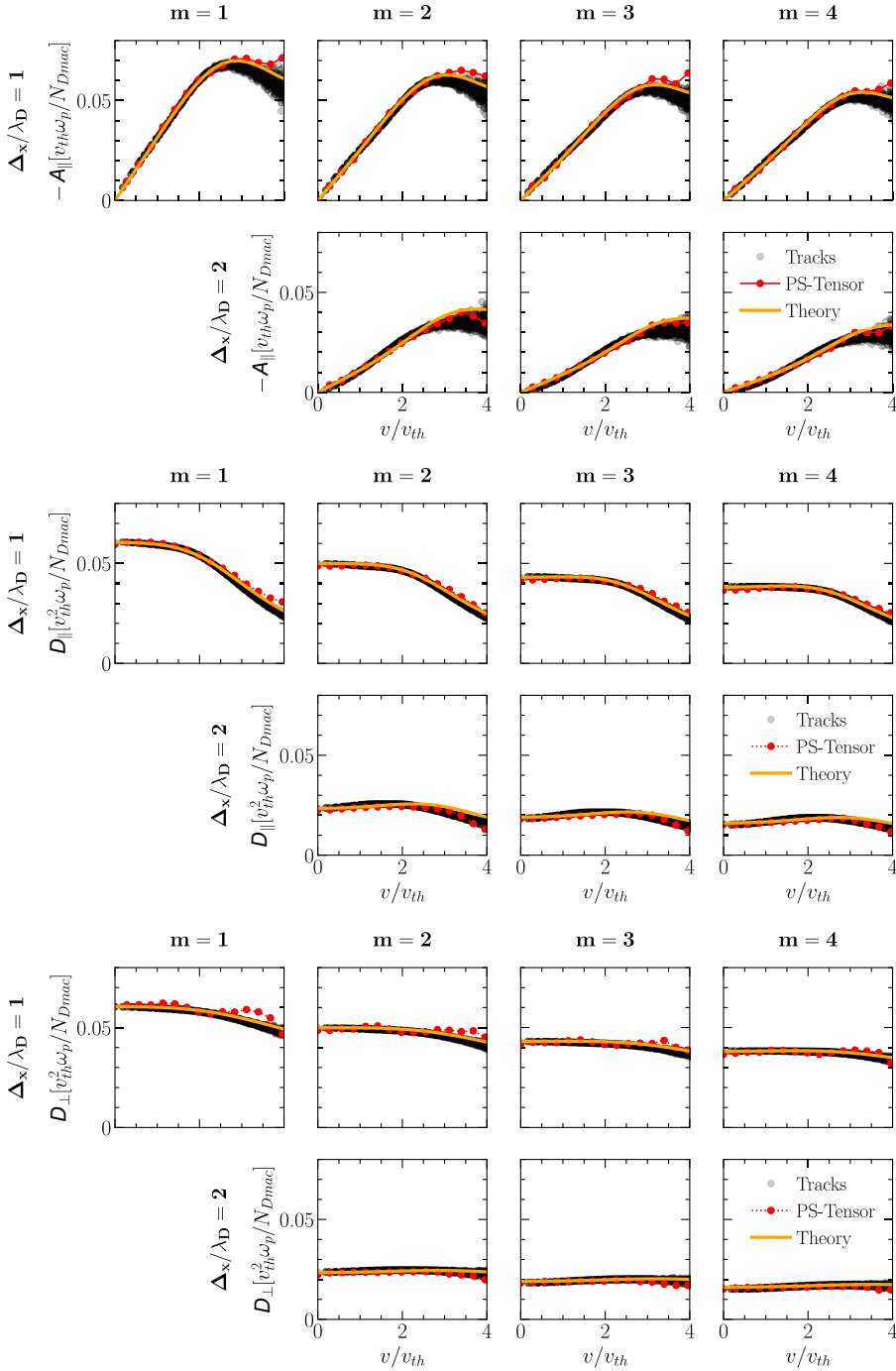


FIGURE 10. Comparison between theoretical values for advection/diffusion coefficients and those obtained from particle tracks and the PS-Tensor method for different shape functions and grid resolutions. All simulations shown use $N_{ppc} = 25$ and $v_{th} = 0.01c$. There is overall an excellent agreement with theory, particularly up to $v/v_{th} = 3$. PS-Tensor is expected to not correctly capture well values above this threshold since training data did not contain significant statistics. Particle tracks measurements are also noisier after this value similarly due to lack of statistics.

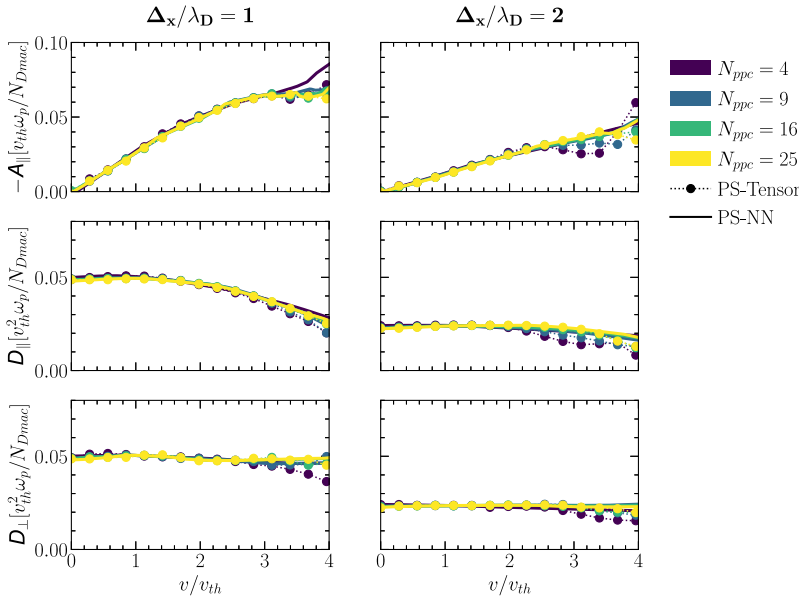


FIGURE 11. Comparison between $AD_{\parallel\perp}$ operators for PS-Tensor and PS-NN models in function of the number of particles per cell. Curves are shown for $m = 2$, $v_{th} = 0.01c$ and varied grid resolutions. We observe that the advection and diffusion coefficients are inversely proportional to N_{Dmac} as predicted by the theory (Langdon 1970; Langdon & Birdsall 1970; Touati *et al.* 2022).

the phase space dynamics of subpopulations with the same level of accuracy at smaller values of N_{ppc} (cf. figure S14 in the Supplemental Material). It is clear that for smaller values of $v/v_{th} < 2.5$, both models recover the expected dependency, i.e. there exists no visible difference between the normalised curves. For $v/v_{th} > 2.5$ the conclusions are model-dependent. For PS-NN, the expected scaling is still observed, for PS-Tensor, we obtain slightly lower coefficients for lower N_{ppc} . We attribute the difference in behaviour to the lack of statistics at high v and lower N_{ppc} , which makes the coefficient retrieval less accurate. While the PS-NN tends to extrapolate using a smoother estimate, the PS-Tensor model is more prone to noisy estimates.

Finally, in figure 12, we plot the advection and diffusion coefficients with respect to v_{th} for varied grid resolutions, while fixing $N_{ppc} = 25$ and $m = 2$. Once again, we recover the dependency predicted by theory, i.e. $A_{\parallel} \propto v_{th}$ and $D_{\parallel,\perp} \propto v_{th}^2$. This indicates that at these temperatures, electromagnetic interactions do not play a significant role in collisional dynamics since the collision operator is not modified.

These results allow us to demonstrate not only the accuracy of the existing theory, and its applicability to electromagnetic PIC simulations, but also the validity of the proposed approach to learn collision operators from phase space data. As far as we know, this is also the first time that such an in-depth verification of the theoretical PIC collision operator has been performed. This comparison also highlights that the methods proposed here can be used in physical regimes where analytical or closed-form expressions for $\mathbf{A}$ and $\mathbf{D}$ are not available or can not easily be determined. This will be explored in future publications.

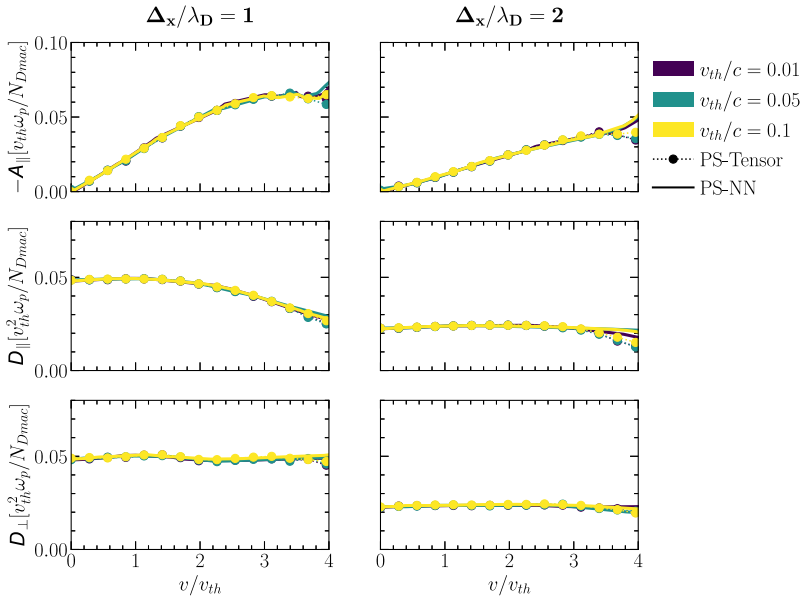


FIGURE 12. Comparison between $AD_{\parallel\perp}$ operators for PS-Tensor and PS-NN models in function of v_{th} . Curves are shown for $N_{ppc} = 25$, $m = 2$ and varied grid resolutions. We observe that advection is proportional to v_{th} and diffusion is proportional to v_{th}^2 as predicted by the theory (Langdon 1970; Langdon & Birdsall 1970; Touati *et al.* 2022).

4. Conclusions

In this paper we proposed a methodology which combines a differentiable simulator with gradient based optimisation methods to learn FP collision operators from self-consistent PIC simulation data. We demonstrated that our method can efficiently learn an optimal operator based on phase space dynamics of subpopulations of particles across a wide range of simulations. The learned operators are on average more accurate than those estimated from particle tracks since they are optimised to predict the long-term phase space dynamics, and the method does not make assumptions about the relevant time scales of different plasma processes. We have also shown that the recovered operators are in excellent agreement with the theoretical PIC collisional operator for a wide variety of numerical simulation parameters, further validating the theory and the proposed machine learning approach.

This work is a proof of principle, where we learn a collision operator which captures the self-consistent collisional dynamics of thermal plasmas composed of finite-size particles. The same approach can be used to learn collisional operators from self-consistent point-like particle dynamics, e.g. using MD simulations (Zhao & Lei 2025; Zhao *et al.* 2025) or PIC simulations which resolve the inter-particle distance (Decker *et al.* 1994; Acciarri *et al.* 2024). These data-driven operators could then be integrated into Vlasov simulators (Thomas *et al.* 2012) or collisional modules in PIC codes (Takizuka & Abe 1977; Manheimer *et al.* 1997), to statistically reproduce the collisional effects without the need to self-consistently resolve the collisions. Furthermore, we note that, in turbulent collisionless systems, the evolution of the distribution function can be described by a kinetic equation with an anomalous collision

operator. Therefore our approach can also be used to learn reduced models of turbulent systems when provided with data of such set-ups (Wong *et al.* 2020, 2025; Camporeale *et al.* 2022). Finally, while a FP description is used in this work, the differentiable simulator framework here proposed can be easily extended to include other operator forms (Zhao & Lei 2025; Zhao *et al.* 2025).

Since our method relies on phase space diagnostics rather than particle tracks, it is substantially more memory efficient than other post-processing approaches based on tracking individual particles. This advantage is particularly important for future studies employing large-scale, state-of-the-art 3-D simulations, where storing complete particle trajectories is prohibitively expensive and where the relevant time scales or operator forms are not known *a priori* and therefore cannot be identified during run time. Our approach enables diagnostics to be stored at significantly higher temporal cadence under the same memory constraints, allowing for a more comprehensive and better informed analysis during post-processing. Using phase space diagnostics also makes it possible to learn operators from observational (Camporeale *et al.* 2022) or experimental (Bergeson *et al.* 2025) data, although this might require extending the methodology to address the presence of (possibly significant) stochastic noise in the measurements.

In future works, we will focus both on the generalisation of the methodology and on exploring challenging scenarios for which the operator form is not known or the existing theory is expected to fail. We will extend the method to encompass time-varying background distributions, more general operator forms, the inclusion of external field contributions and non-homogeneous plasmas. Our goal is to then apply this method to recover reduced models for non-thermal particle acceleration and collision operators for relativistic, electromagnetically dominated scenarios using self-consistent, first-principles simulations.

Supplementary material

Supplementary materials are available at <https://doi.org/10.1017/S0022377826101755>.

Acknowledgements

The authors would like to thank S. Degen, G. Guttormsen, V. Decyk and I. Kaganovich for valuable discussions, and the anonymous referees for their help improving the quality of the manuscript. The authors acknowledge the OSIRIS Consortium, consisting of UCLA, University of Michigan and IST (Portugal) for the use of the OSIRIS 4.0 framework.

Editor Alex Schekochihin thanks the referees for their advice in evaluating this article.

Funding

Simulations and machine learning workloads were performed in Deucalion (Portugal) within the FCT I.P. project Masers in Astrophysical Plasmas (MAPs, P.B.) 2024.11062.CPCA.A3, FCT I.P. project Machine-learned closures for plasma simulations 2024.12682.CPCA.A1 (D.C.); and EuroHPC proposal No. EHPC-DEV-2025D02-069 (D.C.). This work was supported by the FCT (Portugal) Grants No. 2022.13261.BD (D.C.), No. 2022.02230.PTDC (X-Maser, L.S.) and No. UIDB/FIS/50010/2020-PESTB 2020-23 (L.S.); and by the National Science

Foundation Grant No. PHY-2108089 (E.A.). D.C. research visit to the UCLA was sponsored by a Fulbright Grant for Research with the support of FCT and by the Mani L. Bhaumik Institute for Theoretical Physics at UCLA.

Declaration of interests

The authors report no conflict of interest.

Data availability

The data that support the findings of this study are openly available in Zenodo at: <https://doi.org/10.5281/zenodo.18865875>, reference number 18865875. The software developed is openly available at: <https://github.com/diogodcarvalho/ml-pic-collision-operators>.

REFERENCES

- ACCIARRI, M.D., MOORE, C., BEVING, L.P. & BAALRUD, S.D. 2024 When should PIC simulations be applied to atmospheric pressure plasmas? Impact of correlation heating. *Plasma Sources Sci. Technol.* **33**, 035009.
- ANSEL, J. *et al.* 2024 Pytorch 2: faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 2, pp. 929–947.
- ATZENI, S. & MEYER-TER VEHN, J. 2004 *The Physics of Inertial Fusion: BeamPlasma Interaction, Hydrodynamics, Hot Dense Matter*. Oxford University Press.
- AVERKIN, S.N. & GATSONIS, N.A. 2018 A parallel electrostatic particle-in-cell method on unstructured tetrahedral grids for large-scale bounded collisionless plasma simulations. *J. Comput. Phys.* **363**, 178–199.
- BAALRUD, S.D. 2012 Transport coefficients in strongly coupled plasmas. *Phys. Plasmas* **19**, 030701.
- BAALRUD, S.D. & DALIGAULT, J. 2014 Extending plasma transport theory to strong coupling through the concept of an effective interaction potential. *Phys. Plasmas* **21**, 055707.
- BELL, A.R., ROBINSON, A.P.L., SHERLOCK, M., KINGHAM, R.J. & ROZMUS, W. 2006 Fast electron transport in laser-produced plasmas and the KALOS code for solution of the Vlasov–Fokker–Planck equation. *Plasma Phys. Control. Fusion* **48**, R37.
- BELL, A.R. & SHERLOCK, M. 2024 The fastVFP code for solution of the Vlasov–Fokker–Planck equation. *Plasma Phys. Control. Fusion* **66**, 035014.
- BERGESON, S., SCHLITERS, M., MILLER, M., FARLEY, B., SIEVERTS, D., MURILLO, M.S. & HAACK, J.R. 2025 Experimental and computational study of phase space dynamics in strongly coupled plasmas with steep density gradients. *Phys. Plasmas* **32**, 032104.
- BIRDSALL, C.K. & LANGDON, A.B. 2018 *Plasma Physics Via Computer Simulation*. CRC Press.
- BORIS, J.P. & SHANNY, R.A. 1972, *Proceedings [of the] 4th Conference on Numerical Simulation of Plasmas*. Naval Research Laboratory.
- BRAAMS, B.J. & KARNEY, C.F.F. 1987 Differential form of the collision integral for a relativistic plasma. *Phys. Rev. Lett.* **59**, 1817.
- BRANDSTETTER, J., WORRALL, D. & WELLING, M. 2022 Message passing neural pde solvers. arXiv:[2202.03376](https://arxiv.org/abs/2202.03376).
- BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **113**, 3932–3937.
- CAMPOREALE, E., WILKIE, G.J., DROZDOV, A.Y. & BORTNIK, J. 2022 Data-driven discovery of Fokker–Planck equation for the Earth’s radiation belts electrons using physics-informed neural networks. *J. Geophys. Res.: Space Phys.* **127**, e2022JA030377.
- CARVALHO, D.D., FERREIRA, D.R. & SILVA, L.O. 2024 Learning the dynamics of a one-dimensional plasma model with graph neural networks. *Mach. Learn.: Sci. Technol.* **5**, 025048.
- CHABRIER, G., SAUMON, D. & POTEKHIN, A.Y. 2006 Dense plasmas in astrophysics: from giant planets to neutron stars. *J. Phys. A: Math. Gener.* **39**, 4411.

- CHEN, X., YANG, L., DUAN, J. & KARNIADAKIS, G.E. 2021 Solving inverse stochastic problems from discrete particle observations using the Fokker–Planck equation and physics-informed neural networks. *SIAM J. Sci. Comput.* **43**, B811–B830.
- CHUNG, K., FEI, F., GORJI, M.H. & JENNY, P. 2023 Data-driven stochastic particle scheme for collisional plasma simulations. *J. Comput. Phys.* **492**, 112400.
- DECKER, C.D., MORI, W.B., DAWSON, J.M. & KATSIOULEAS, T. 1994 Nonlinear collisional absorption in laser-driven plasmas. *Phys. Plasmas* **1**, 4043–4049.
- ESIRKEPOV, T.Z. 2001 Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor. *Comput. Phys. Commun.* **135**, 144–153.
- FONSECA, R.A. *et al.* 2002 Osiris: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators. In *Computational Science – ICCS 2002: International Conference Amsterdam, The Netherlands, April 21–24, 2002 Proceedings, Part III 2*, vol. 2, pp. 342–351. Springer.
- FRIED, B.D. & CONTE, S.D. 1961 *The Plasma Dispersion Function: The Hilbert Transform of the Gaussian*. Academic Press.
- GATSONIS, N.A. & SPIRKIN, A. 2009 A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay–Voronoi grids. *J. Comput. Phys.* **228**, 3742–3761.
- GUREVICH, D.R., GOLDEN, M.R., REINBOLD, P.A.K. & GRIGORIEV, R.O. 2024 Learning fluid physics from highly turbulent data using sparse physics-informed discovery of empirical relations (SPIDER). *J. Fluid Mech.* **996**, A25.
- HOCKNEY, R.W. 1971 Measurements of collision and heating times in a two-dimensional thermal computer plasma. *J. Comput. Phys.* **8**, 19–44.
- HOLLOWAY, I., WOOD, A. & ALEKSEENKO, A. 2021 Acceleration of Boltzmann collision integral calculation using machine learning. *Mathematics* **9**, 1384.
- HWANG, H.J., JANG, J.W., JO, H. & LEE, J.Y. 2020 Trend to equilibrium for the kinetic Fokker–Planck equation via the neural network approach. *J. Comput. Phys.* **419**, 109665.
- JOGLEKAR, A.S. 2024 Generative neural reparameterization for differentiable PDE-constrained optimization. arXiv:[2410.12683](https://arxiv.org/abs/2410.12683).
- JOGLEKAR, A.S. & THOMAS, A.G.R. 2022 Unsupervised discovery of nonlinear plasma physics using differentiable kinetic simulations. *J. Plasma Phys.* **88**, 905880608.
- JOGLEKAR, A.S. & THOMAS, A.G.R. 2023 Machine learning of hidden variables in multiscale fluid simulation. *Mach. Learn.: Sci. Technol.* **4**, 035049.
- JUBIN, S., POWIS, A.T., VILLAFANA, W., SYDORENKO, D., RAUF, S., KHRABROV, A.V., SARWAR, S. & KAGANOVICH, I.D. 2024 Numerical thermalization in 2D PIC simulations: Practical estimates for low-temperature plasma simulations. *Phys. Plasmas* **31**, 023902.
- KARNAKOV, P., LITVINOV, S. & KOUMOUTSAKOS, P. 2022 Optimizing a discrete loss (ODIL) to solve forward and inverse problems for partial differential equations using machine learning tools. arXiv:[2205.04611](https://arxiv.org/abs/2205.04611).
- KAWAGUCHI, S. & MURAKAMI, T. 2022 Physics-informed neural networks for solving the Boltzmann equation of the electron velocity distribution function in weakly ionized plasmas. *Jpn. J. Appl. Phys.* **61**, 086002.
- KILLIAN, T.C., PATTARD, T., POHL, T. & ROST, J.M. 2007 Ultracold neutral plasmas. *Phys. Rep.* **449**, 77–130.
- KINGMA, D.P. 2014 Adam: a method for stochastic optimization. arXiv:[1412.6980](https://arxiv.org/abs/1412.6980).
- KLIMONTOVICH, Y.L. 1967 *The Statistical Theory of Non-Equilibrium Processes in a Plasma*. Pergamon Press.
- KOVACHKI, N., LI, Z., LIU, B., AZIZZADENESHELI, K., BHATTACHARYA, K., STUART, A. & ANANDKUMAR, A. 2023 Neural operator: learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res.* **24**, 1–97.
- KUSHNER, M.J. 2020 Plasma science 2020 decadal survey: overview and opportunities for the low temperature plasma community. In *73rd Annual Gaseous Electronics Virtual Conference*. APS.
- LANDAU, L.D. 1936 Die kinetische gleichung fuer den fall coulombscher wechselwirkung. *Phys. Z. Sowjet* **10**, 163.

- LANGDON, A.B. 1970 Nonphysical modifications to oscillations, fluctuations, and collisions due to space-time differencing. In *Proceedings of the 4th Conference on Numerical Simulation of Plasmas*, pp. 467–495.
- LANGDON, A.B. & BIRDSALL, C.K. 1970 Theory of plasma simulation using finite-size particles. *Phys. Fluids* **13**, 2115–2122.
- LEE, J.Y., JANG, J. & HWANG, H.J. 2023 opPINN: physics-informed neural network with operator learning to approximate solutions to the Fokker–Planck–Landau equation. *J. Comput. Phys.* **480**, 112031.
- LEE, J.Y., JANG, J.W. & HWANG, H.J. 2021 The model reduction of the Vlasov–Poisson–Fokker–Planck system to the Poisson–Nernst–Planck system via the deep neural network approach. *ESAIM: Math. Model. Numer. Anal.* **55**, 1803–1846.
- LEE, J.Y., JUNG, G.J., LIM, B.C. & HWANG, H.J. 2025 Fourierspecnet: neural collision operator approximation inspired by the fourier spectral method for solving the boltzmann equation. arXiv:2504.20408.
- LEE, J.Y., SCHOTTHÖFER, S., XIAO, T., KRUMSCHEID, S. & FRANK, M. 2024 Structure-preserving operator learning: modeling the collision operator of kinetic equations. arXiv:2402.16613.
- LI, Z., KOVACHKI, N., AZIZZADENESHELI, K., LIU, B., BHATTACHARYA, K., STUART, A. & ANANDKUMAR, A. 2020 Fourier neural operator for parametric partial differential equations. arXiv:2010.08895.
- LI, Z., WANG, Y., LIU, H., WANG, Z. & DONG, B. 2024 Solving the Boltzmann equation with a neural sparse representation. *SIAM J. Sci. Comput.* **46**, C186–C215.
- LIST, B., CHEN, L.-W., BALI, K. & THUEREY, N. 2025 Differentiability in unrolled training of neural physics simulators on transient dynamics. *Comput. Meth. Appl. Mech. Engng* **433**, 117441.
- LOU, Q., MENG, X. & KARNIADAKIS, G.E. 2021 Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann–BGK formulation. *J. Comput. Phys.* **447**, 110676.
- LU, L., JIN, P. & KARNIADAKIS, G.E. 2019 DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv:1910.03193.
- MANHEIMER, W.M., LAMPE, M. & JOYCE, G. 1997 Langevin representation of Coulomb collisions in PIC simulations. *J. Comput. Phys.* **138**, 563–584.
- MATSUDA, Y. & OKUDA, H. 1975 Collisions in multi-dimensional plasma simulations. *Phys. Fluids* **18**, 1740–1747.
- MCDEVITT, C.J. 2023 A physics-informed deep learning model of the hot tail runaway electron seed. *Phys. Plasmas* **30**, 092501.
- MCDEVITT, C.J., ARNAUD, J.S. & TANG, X.-Z. 2025a An efficient surrogate model of secondary electron formation and evolution. *Phys. Plasmas* **32**, 042504.
- MCDEVITT, C.J., ARNAUD, J.S. & TANG, X.Z. 2025b A physics-constrained deep learning treatment of runaway electron dynamics. *Phys. Plasmas* **32**, 042503.
- MERLINO, R.L. & GOREE, J.A. 2004 Dusty plasmas in the laboratory, industry, and space. *Phys. Today* **57**, 32–38.
- MILLER, M.A., CHURCHILL, R.M., DENER, A., CHANG, C.-S., MUNSON, T. & HAGER, R. 2021 Encoder–decoder neural network for solving the nonlinear Fokker–Planck–Landau collision operator in XGC. *J. Plasma Phys.* **87**, 905870211.
- MILLER, S.T., ROBERTS, N.V., BOND, S.D. & CYR, E.C. 2022 Neural-network based collision operators for the Boltzmann equation. *J. Comput. Phys.* **470**, 111541.
- NOH, H., LEE, J. & YOON, E. 2025 FPL-net: a deep learning framework for solving the nonlinear Fokker–Planck–Landau collision operator for anisotropic temperature relaxation. *J. Comput. Phys.* **523**, 113665.
- OH, J., CHO, S.Y., YUN, S.-B., PARK, E. & HONG, Y. 2025 Separable physics-informed neural networks for solving the BGK model of the Boltzmann equation. *SIAM J. Sci. Comput.* **47**, C451–C474.
- OKUDA, H. & BIRDSALL, C.K. 1970 Collisions in a plasma of finite-size particles. *Phys. Fluids* **13**, 2123–2134.
- PARK, R.M., MOORE, C.H. & BAALRUD, S.D. 2025 Numerical thermalization in 3d pic simulations. In *DPP 2025*. APS.
- PIKE, O.J. & ROSE, S.J. 2016 Transport coefficients of a relativistic plasma. *Phys. Rev. E* **93**, 053208.

- PlasmaPy Community *et al.* 2026 PlasmaPy, version 2026.2.0. Zenodo. <https://doi.org/10.5281/zenodo.18706665>
- RACKAUCKAS, C., MA, Y., MARTENSEN, J., WARNER, C., ZUBOV, K., SUPEKAR, R., SKINNER, D., RAMADHAN, A. & EDELMAN, A. 2020 Universal differential equations for scientific machine learning. *arXiv:2001.04385*.
- RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G.E. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.
- REYNOLDS, M.A., FRIED, B.D. & MORALES, G.J. 1997 Velocity–space drag and diffusion in a model, two-dimensional plasma. *Phys. Plasmas* **4**, 1286–1296.
- ROSENBLUTH, M.N., MACDONALD, W.M. & JUDD, D.L. 1957 Fokker–Planck equation for an inverse-square force. *Phys. Rev.* **107**, 1.
- RUDY, S.H., BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614.
- TAKIZUKA, T. & ABE, H. 1977 A binary collision model for plasma simulation with a particle code. *J. Comput. Phys.* **25**, 205–219.
- THOMAS, A.G.R., TZOUFRAS, M., ROBINSON, A.P.L., KINGHAM, R.J., RIDGERS, C.P., SHERLOCK, M. & BELL, A.R. 2012 A review of Vlasov–Fokker–Planck numerical modeling of inertial confinement fusion plasma. *J. Comput. Phys.* **231**, 1051–1079.
- TOUATI, M., CODUR, R., TSUNG, F., DECYK, V.K., MORI, W.B. & SILVA, L.O. 2022 Kinetic theory of particle-in-cell simulation plasma and the ensemble averaging technique. *Plasma Phys. Control. Fusion* **64**, 115014.
- TZOUFRAS, M., BELL, A.R., NORREYS, P.A. & TSUNG, F.S. 2011 A Vlasov–Fokker–Planck code for high energy density physics. *J. Comput. Phys.* **230**, 6475–6494.
- TZOUFRAS, M., TABLEMAN, A., TSUNG, F.S., MORI, W.B. & BELL, A.R. 2013 A multi-dimensional Vlasov–Fokker–Planck code for arbitrarily anisotropic high-energy-density plasmas. *Phys. Plasmas* **20**, 056303.
- WANG, T., HU, Z., KAWAGUCHI, K., ZHANG, Z. & KARNIADAKIS, G.E. 2025 Tensor neural networks for high-dimensional Fokker–Planck equations. *Neural Netw.* **185**, 107165.
- WONG, K., ZHDANKIN, V., UZDENSKY, D.A., WERNER, G.R. & BEGELMAN, M.C. 2020 First-principles demonstration of diffusive-advective particle acceleration in kinetic simulations of relativistic plasma turbulence. *Astrophys. J. Lett.* **893**, L7.
- WONG, K.W., ZHDANKIN, V., UZDENSKY, D.A., WERNER, G.R. & BEGELMAN, M.C. 2025 Energy diffusion and advection coefficients in kinetic simulations of relativistic plasma turbulence. *arXiv:2502.03042*.
- XIAO, T. & FRANK, M. 2021 Using neural networks to accelerate the solution of the Boltzmann equation. *J. Comput. Phys.* **443**, 110521.
- XIAO, T. & FRANK, M. 2023 RelaxNet: a structure-preserving neural network to approximate the Boltzmann collision operator. *J. Comput. Phys.* **490**, 112317.
- XU, Y., ZHANG, H., LI, Y., ZHOU, K., LIU, Q. & KURTHS, J. 2020 Solving Fokker–Planck equation using deep learning. *Chaos: Interdiscip. J. Nonlinear Sci.* **30**, 013133.
- YEE, K. 1966 Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media. *IEEE Trans. Antenn. Propag.* **14**, 302–307.
- ZAHARIA, M.A. *et al.* 2018 Accelerating the machine learning lifecycle with MLflow. *IEEE Data Engng Bull.* **41**, 39–45.
- ZHAI, J., DOBSON, M. & LI, Y. 2022 A deep learning method for solving Fokker–Planck equations. In *Mathematical and Scientific Machine Learning*, pp. 568–597. PMLR.
- ZHANG, Y. & YUEN, K.-V. 2022 Physically guided deep learning solver for time-dependent Fokker–Planck equation. *Intl J. Nonlinear Mech.* **147**, 104202.
- ZHAO, Y., BURBY, J., CHRISTLIEB, A. & LEI, H. 2025 Data-driven construction of a generalized kinetic collision operator from molecular dynamics. *Phys. Rev. Lett.* **135**, 185101.
- ZHAO, Y. & LEI, H. 2025 Fast spectral separation method for kinetic equation with anisotropic non-stationary collision operator retaining micro-model fidelity. *arXiv:2510.15093*.
- ZHONG, L., WU, B. & WANG, Y. 2022 Low-temperature plasma simulation based on physics-informed neural networks: frameworks and preliminary applications. *Phys. Fluids* **34**, 087116.