In search of the (ground) truth & kinetic plasma simulators

Luis O. Silva*, **Diogo Carvalho***,**

Diogo Ferreira*, Paulo Alves**

* IST, Lisbon,Portugal, ** UCLA

epp.tecnico.ulisboa.pt || **golp**.tecnico.ulisboa.pt

In search of the (ground) truth & kinetic plasma simulators

Luis O. Silva*, **Diogo Carvalho***,**

Diogo Ferreira*, Paulo Alves**

* IST, Lisbon,Portugal, ** UCLA

epp.tecnico.ulisboa.pt || **golp**.tecnico.ulisboa.pt

Motivation & questions

Some (of my) questions:

Can ML help us speed up standard plasma simulators?

Our early attempts: ML replaces Monte Carlo modules in PIC - Badiali et al., JPP 2022; Amaro et al., arXiv:2406.02491

Can we build faster ML based simulators?

Rethinking architecture of simulators to match ML uniqueness: 1D collisional plasma model - Carvalho et al.; MLST 2023

What can we learn from data-driven approaches + ML?

Learning physics (following Alves & Fiuza) e.g. collision operators: Carvalho et al., in preparation for submission to JPP

Can standard plasma simulators provide "high quality data" for data-driven discovery?

Capturing collisions in PIC codes: D. Carvalho et al., in preparation

Can we understand qualitative modifications of plasma behavior from "Learning what we already know"

e.g. Waterbag vs Maxwellian; nonlinear waves vs unstable (and then turbulent) scenarios; nonrelativistic to relativistic, Casimir invariants evolution

Motivation & questions

Some (of my) questions:

Can ML help us speed up standard plasma simulators?

Our early attempts: ML replaces Monte Carlo modules in PIC - Badiali et al., JPP 2022; Amaro et al., arXiv:2406.02491

Can we build faster ML based simulators?

Rethinking architecture of simulators to match ML uniqueness: 1D collisional plasma model - Carvalho et al.; MLST 2023

What can we learn from data-driven approaches + ML?

Learning physics (following Alves & Fiuza) e.g. collision operators: Carvalho et al., in preparation for submission to JPP

Can standard plasma simulators provide "high quality data" for data-driven discovery?

Capturing collisions in PIC codes: D. Carvalho et al., in preparation

Can we understand qualitative modifications of plasma behavior from "Learning what we already know"

e.g. Waterbag vs Maxwellian; nonlinear waves vs unstable (and then turbulent) scenarios; nonrelativistic to relativistic, Casimir invariants evolution

"There are unknown unknowns" (and "know unknowns"), Jon Arons citing D. Rumsfled

MC models in PIC simulations

New simulator models - 1D GNN collisional plasma model

Learning advection and diffusion coefficients

The (ground) truth? - collisions in PIC codes

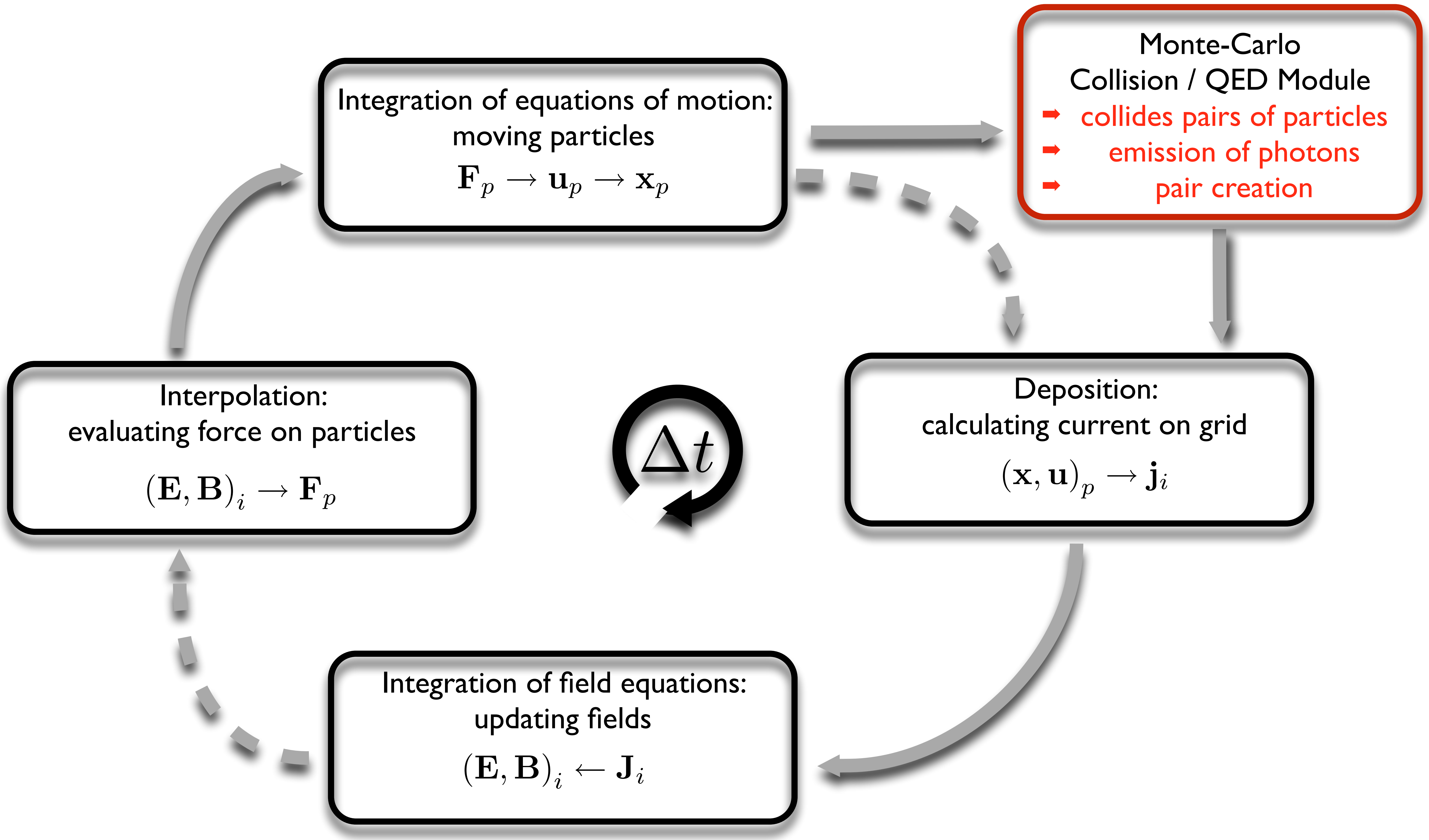
MC models in PIC simulations

New simulator models - 1D GNN collisional plasma model

Learning advection and diffusion coefficients

The (ground) truth? - collisions in PIC codes

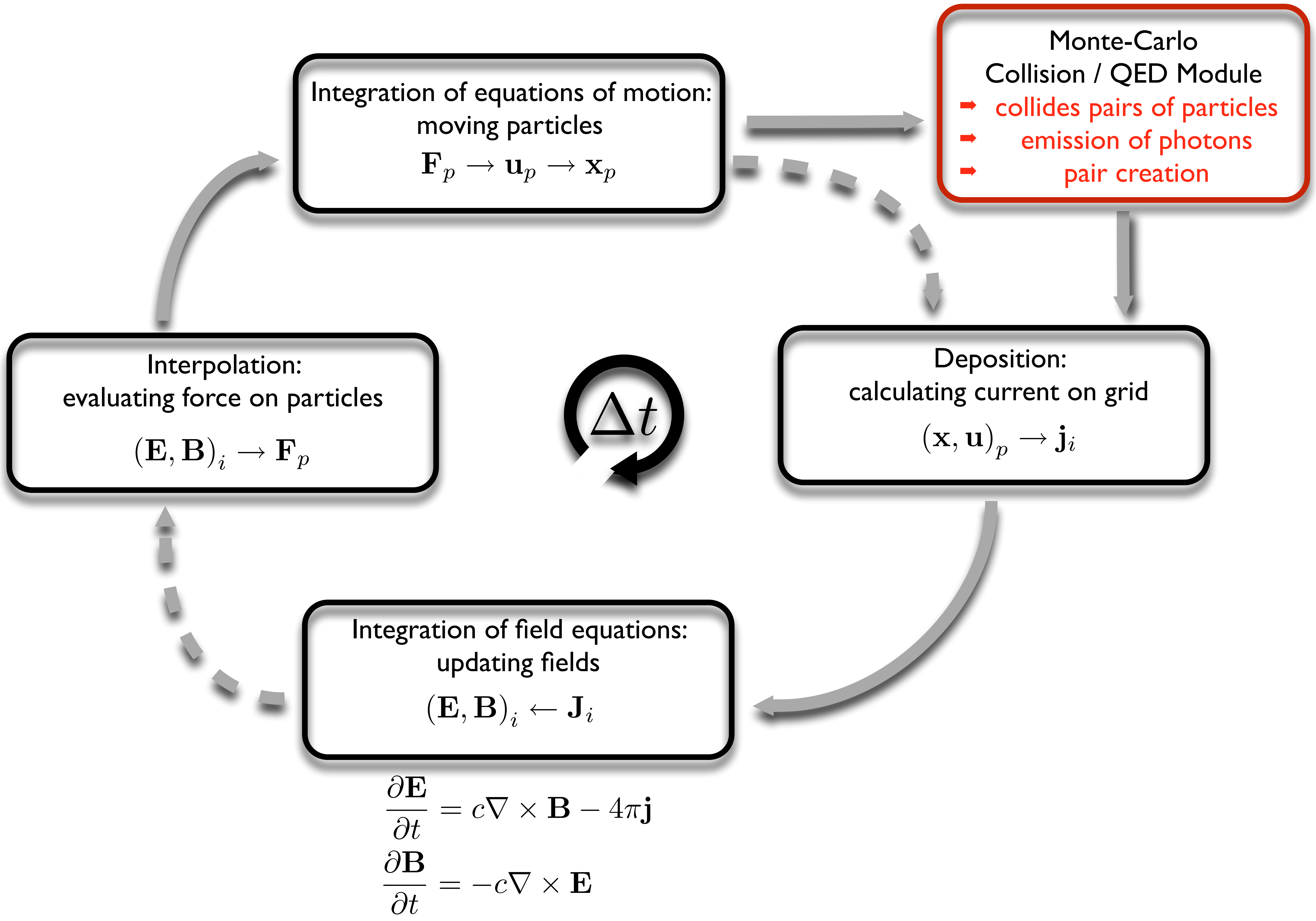
Collisions/QED Physics are modelled using Monte-Carlo routines



$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}$$

Collisions/QED Physics are modelled using Monte-Carlo routines



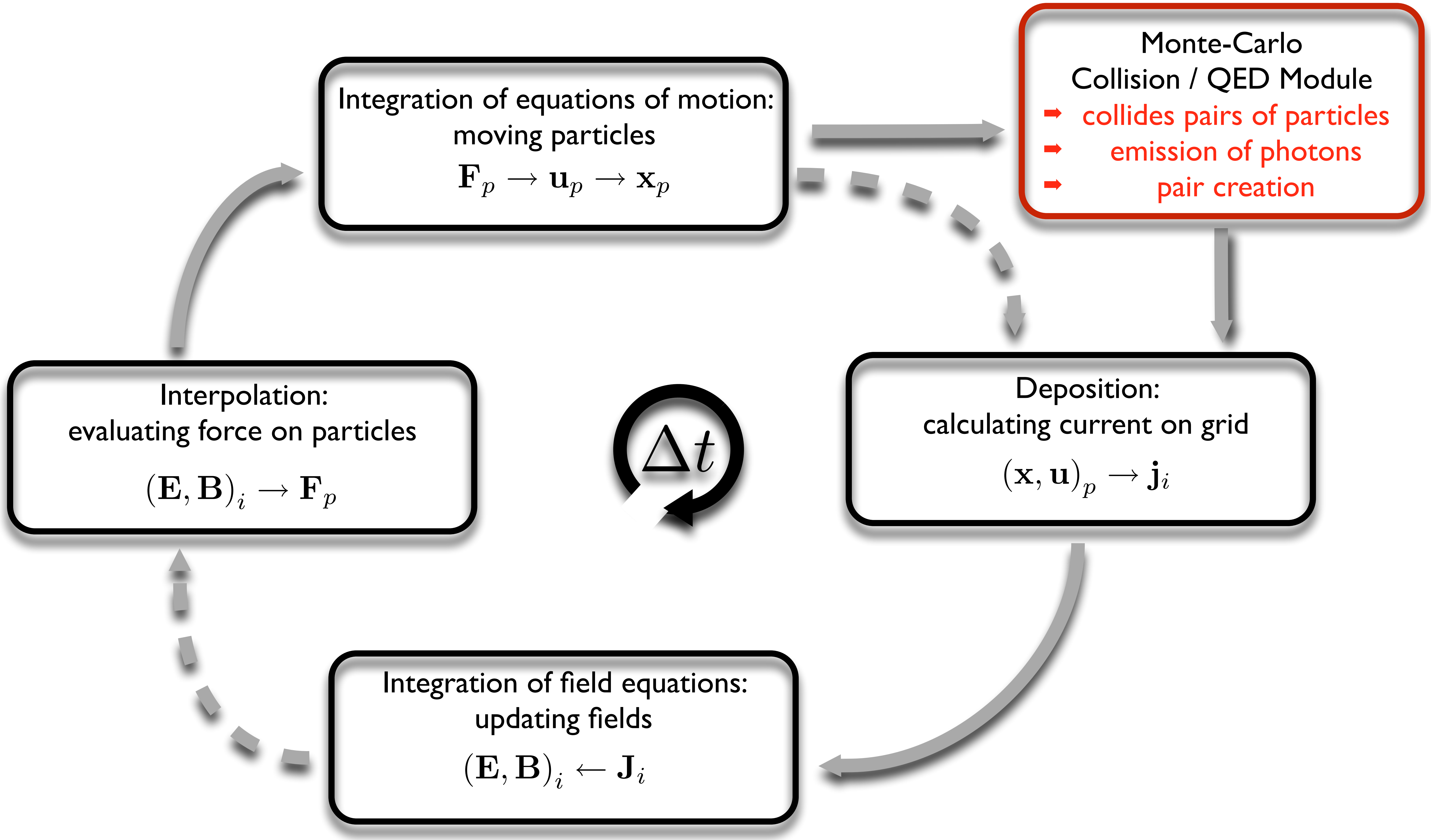
Problems

Computationally intensive
Memory & run-time

Can lead to numerical issues
e.g. increased numerical heating

Theory valid in limited scenarios
e.g. small angle-scattering

Collisions/QED Physics are modelled using Monte-Carlo routines



$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}$$
$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}$$

Problems

Computationally intensive
Memory & run-time

Can lead to numerical issues
e.g. increased numerical heating

Theory valid in limited scenarios
e.g. small angle-scattering

Can ML tackle these issues?

Reduce computational cost
Design new (stable) numerical algorithms
Learn corrections to existing theory

MC routines require the calculation of collision cross-sections

TÉCNICO LISBOA

How are cross-sections calculated?

Theory

Usually impracticable at run-time

Interpolation Tables

Fast to query

Limited to few input parameter values

Chebyshev Polynomials

Exponentially convergent

Impractical for $\geq$ 3D input parameter space

MC routines require the calculation of collision cross-sections

How are cross-sections calculated?

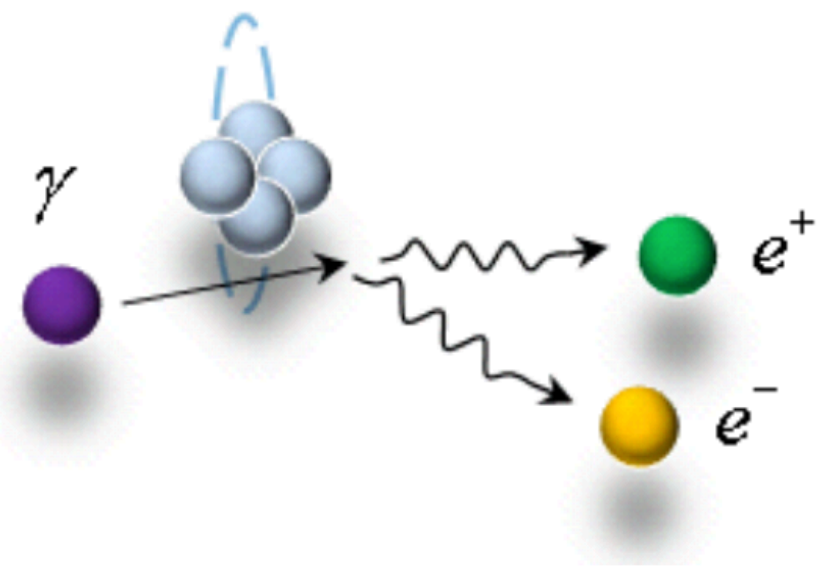
Theory
Usually impracticable at run-time

Interpolation Tables
Fast to query
Limited to few input parameter values

Chebyshev Polynomials
Exponentially convergent
Impractical for ≥ 3D input parameter space

Input parameter spaces are often > 3D

Bethe-Heitler



Usual MC Inputs
Photon Energy
Ion Atomic Number

Inputs that should also influence cross-section
Angle of incoming photon
Plasma temperature
Plasma density
Ionisation degree
Local electromagnetic fields

MC routines require the calculation of collision cross-sections

TÉCNICO LISBOA

How are cross-sections calculated?

Theory

Usually impracticable at run-time

Interpolation Tables

Fast to query

Limited to few input parameter values

Chebyshev Polynomials

Exponentially convergent

Impractical for ≥ 3D input parameter space

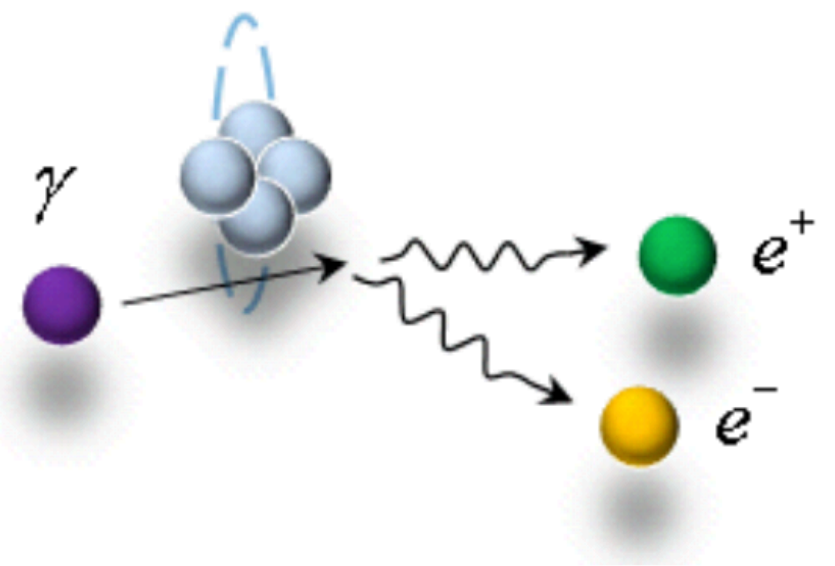
Neural Networks*

Memory efficient for any input parameter space

Run-time dependent on model size

Input parameter spaces are often > 3D

Bethe-Heitler



Usual MC Inputs

Photon Energy

Ion Atomic Number

Inputs that should also influence cross-section

Angle of incoming photon

Plasma temperature

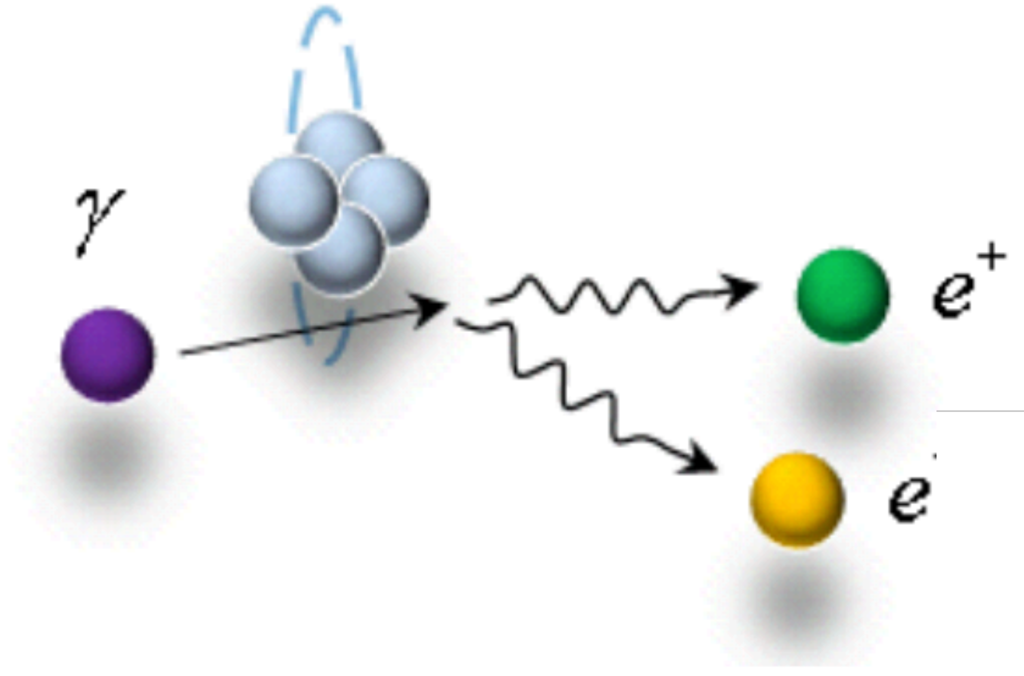
Plasma density

Ionisation degree

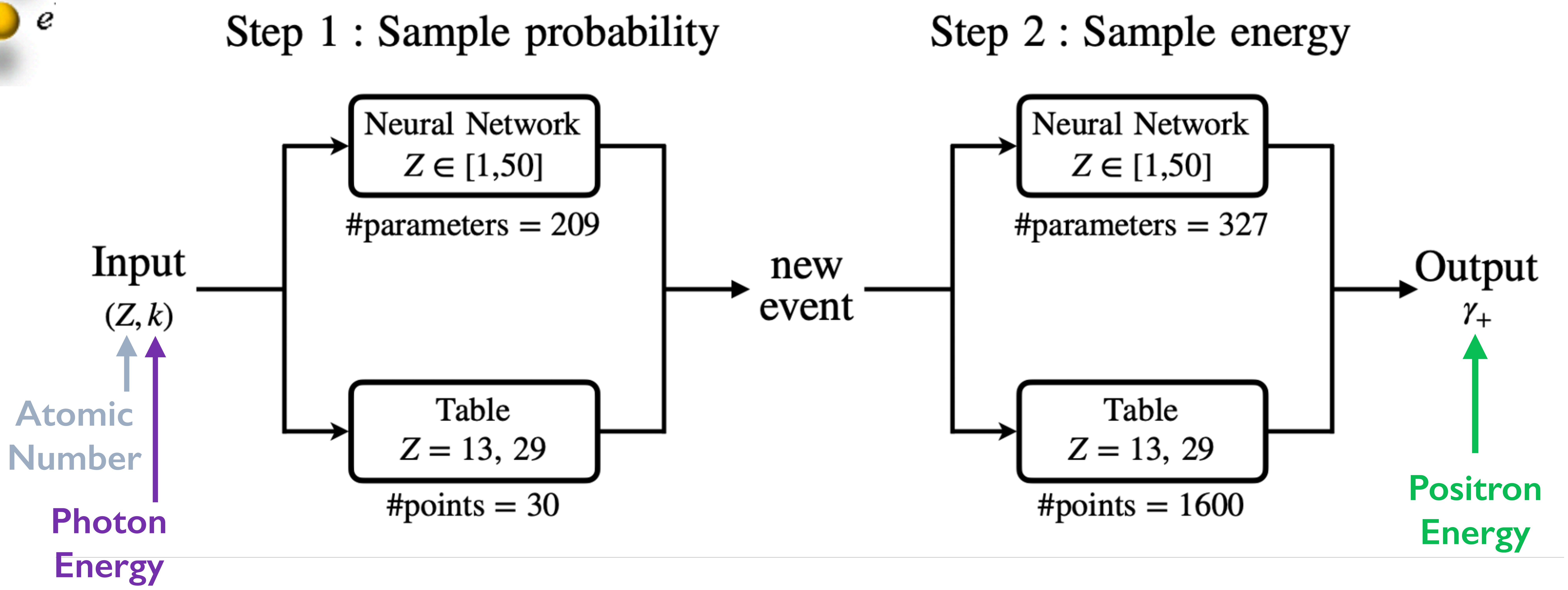
Local electromagnetic fields

* C. Badiali et al., J. Plasma Phys. 88(6) (2022) and O. Amaro et al., arXiv:2406.02491 (2024)

Example in production: Bethe-Heitler pair creation



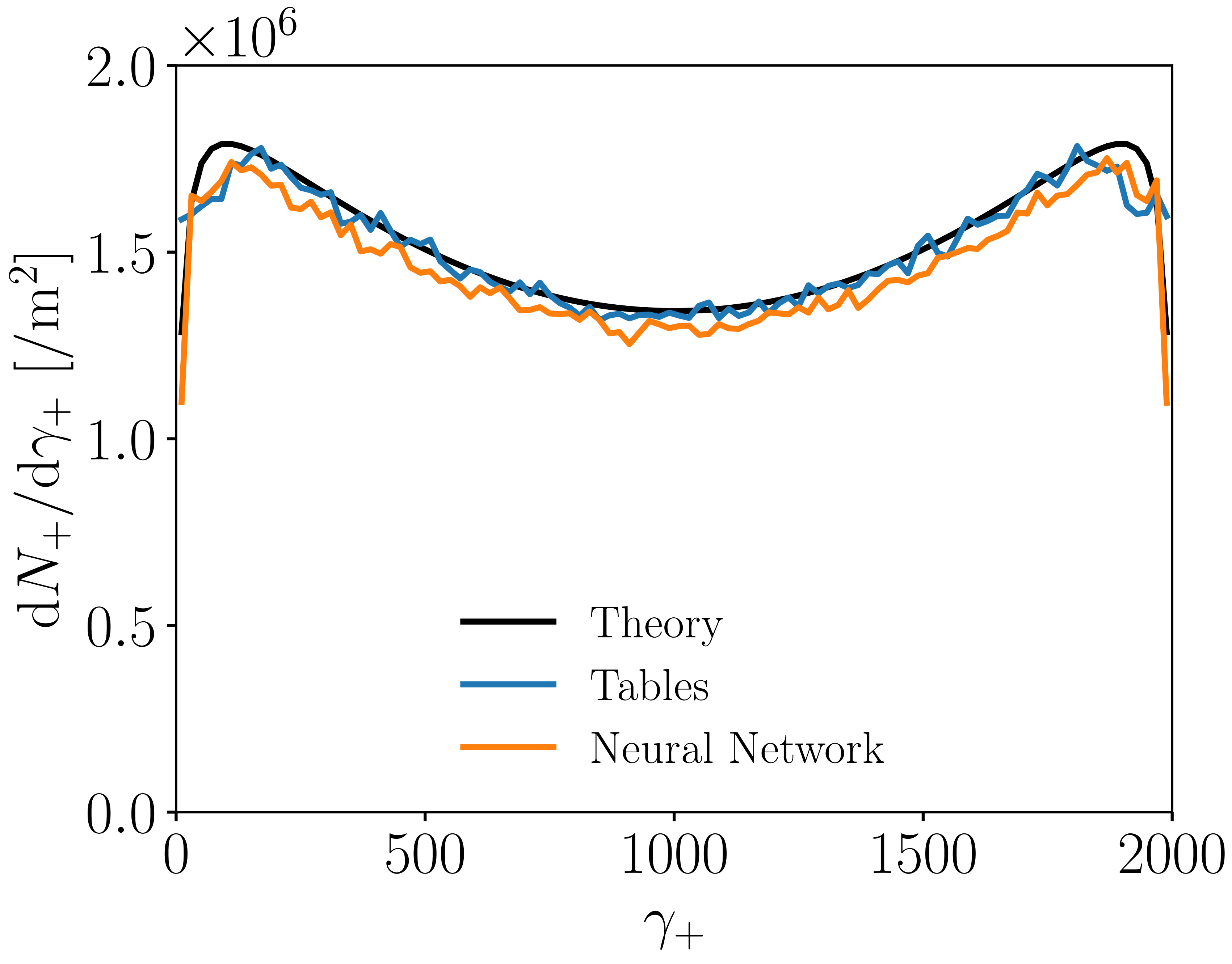
OSIRIS NN Implementation



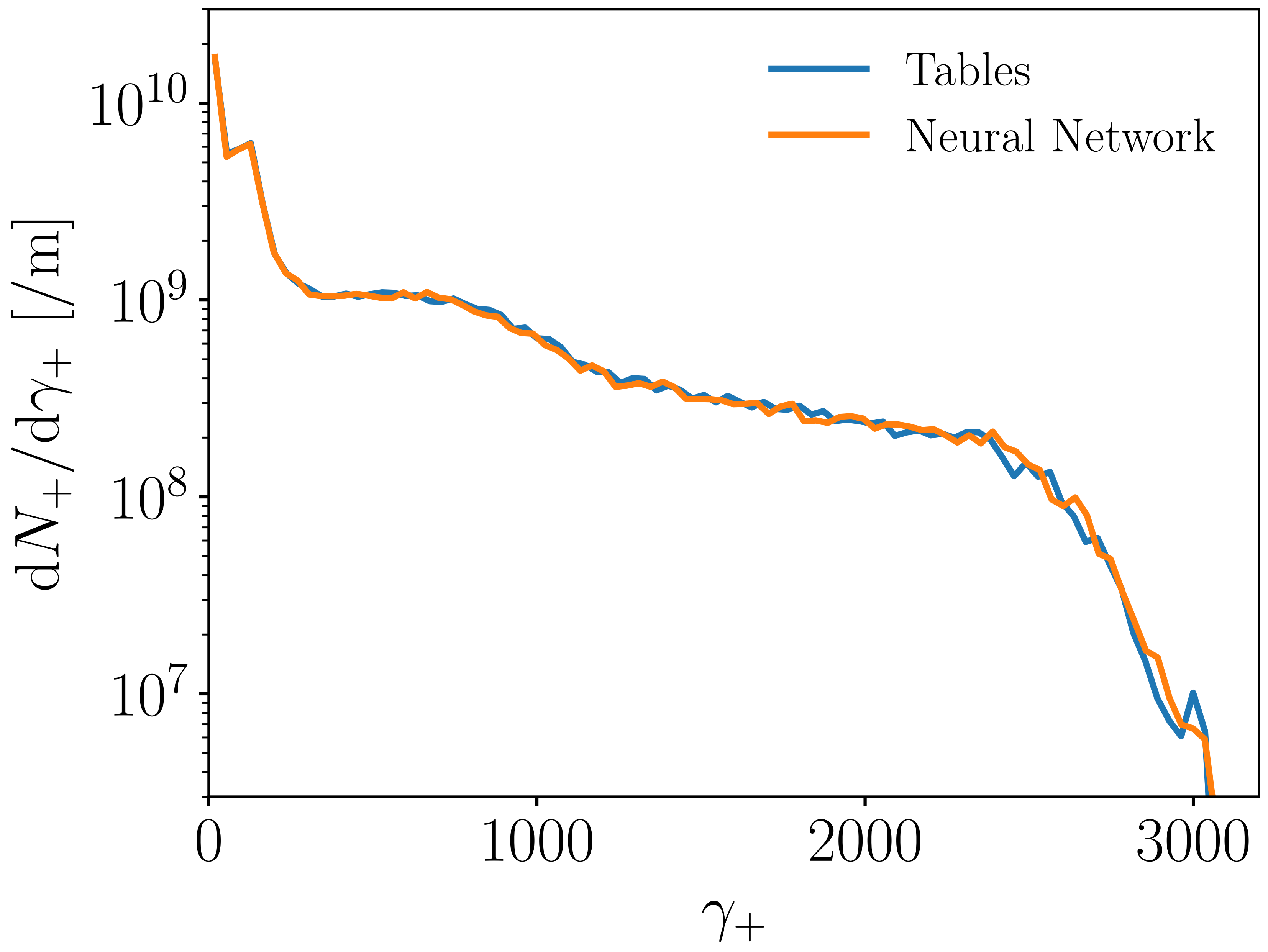
B Martinez et al, Phys. Plasmas 26, 103109 (2019)
C. Badiali et al., J. Plasma Phys. 88(6) (2022)
O. Amaro et al., arXiv:2406.02491 (2024)

NN vs Table: Similar performance at reduced memory cost

Benchmark 1D (Early time evolution)



Production 2D (Laser-solid target long-time evolution)



Light-weight and fast OSIRIS-SFQED

Neural Networks are **as accurate** as pre-calculated tables

Require **x100 less memory** to store and are of **comparable runtime**

"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." von Neumann

O. Amaro et al., arXiv:2406.02491 (2024)

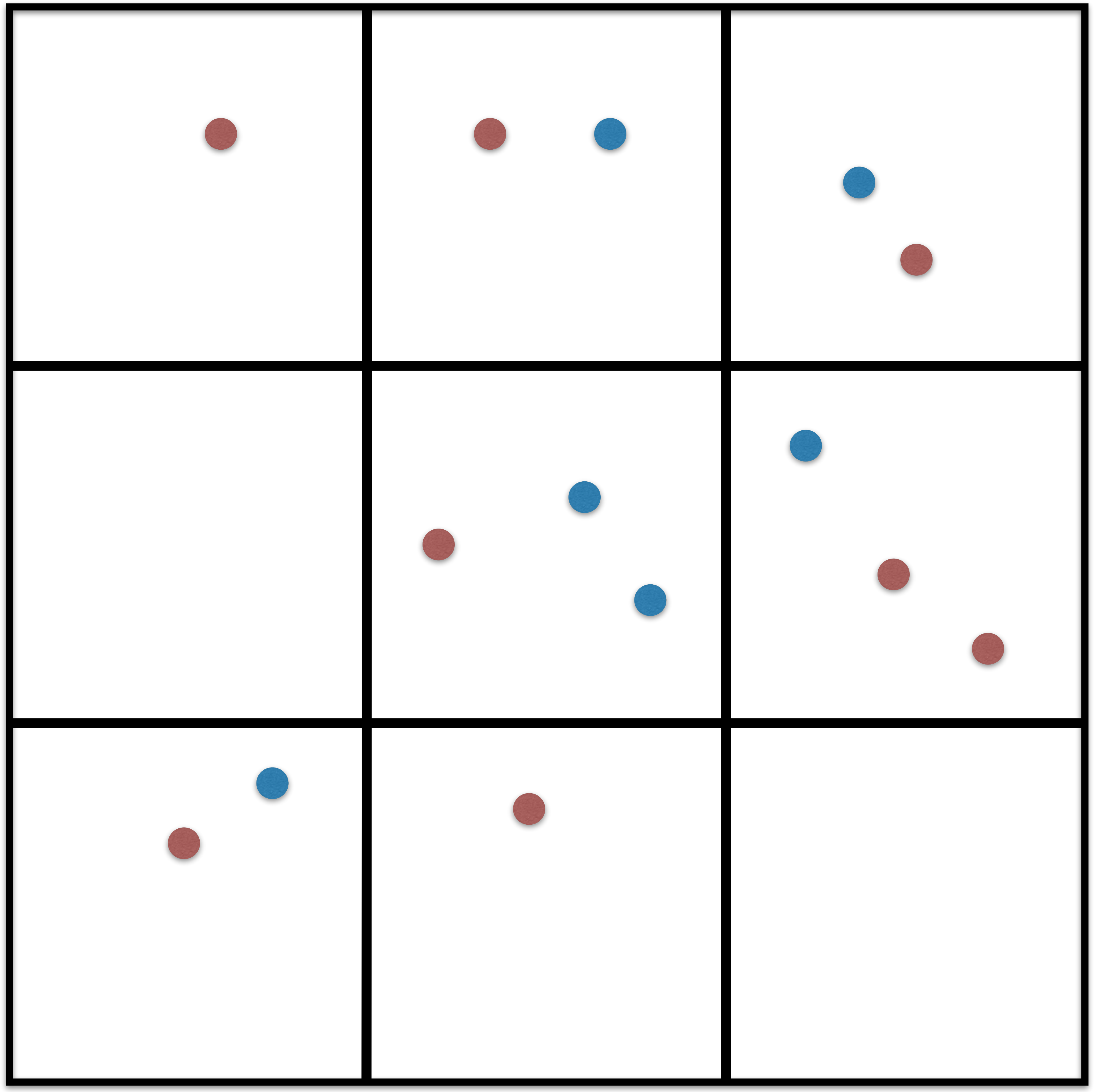
MC models in PIC simulations

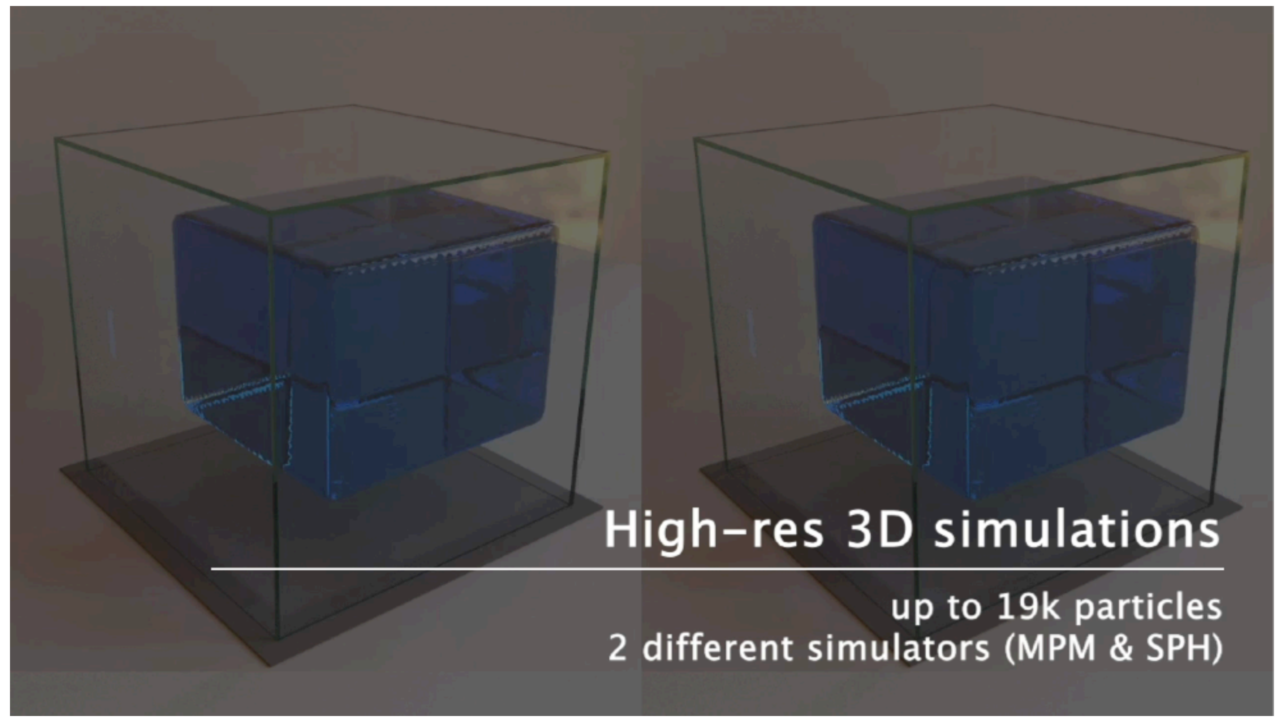
New simulator models - GNN collisional plasma model

Learning advection and diffusion coefficients

The (ground) truth? - collisions in PIC codes

PIC codes (and others) can be seen from a graph perspective

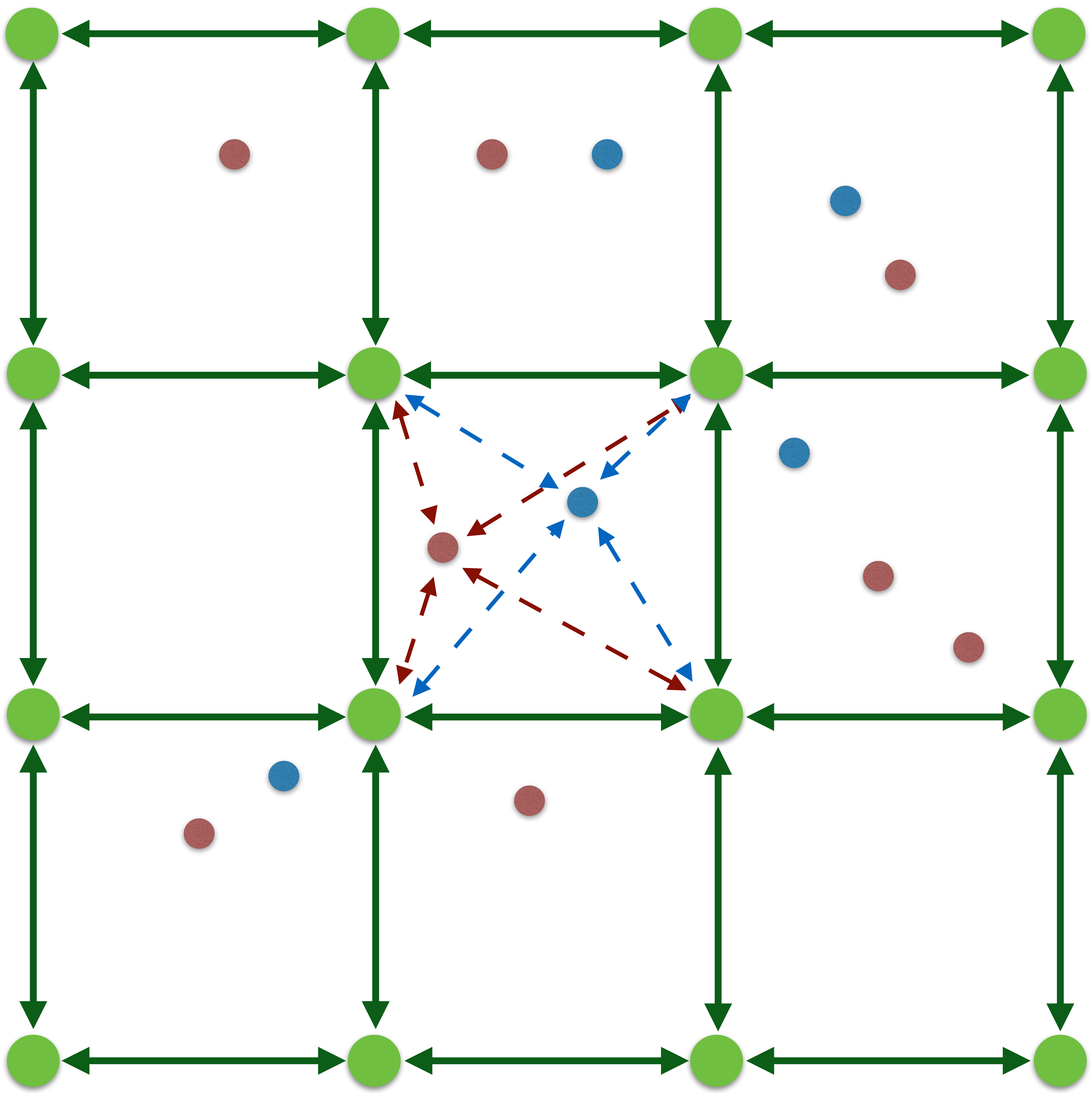


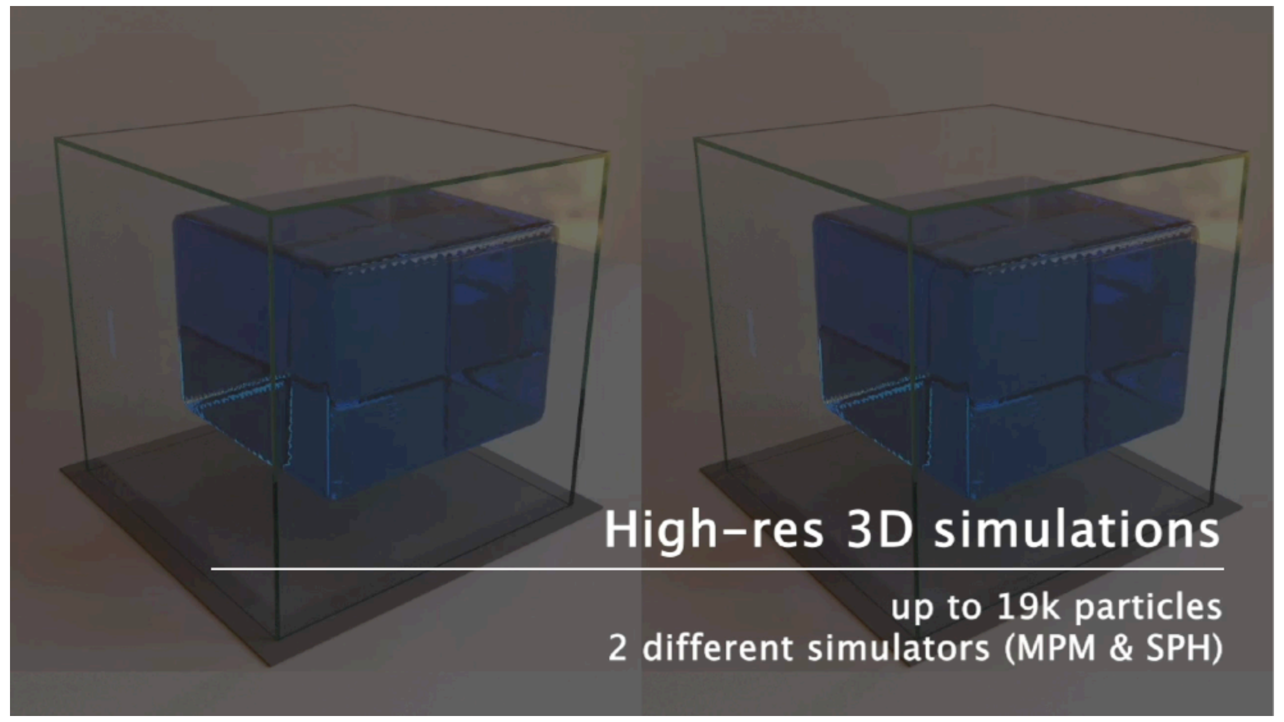


A. Sanchez-Gonzalez et al., ICML PMLR 8459–8468 (2020)
R. Lam et al., Science 382.6677 1416-1421 (2023)

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

PIC codes (and others) can be seen from a graph perspective

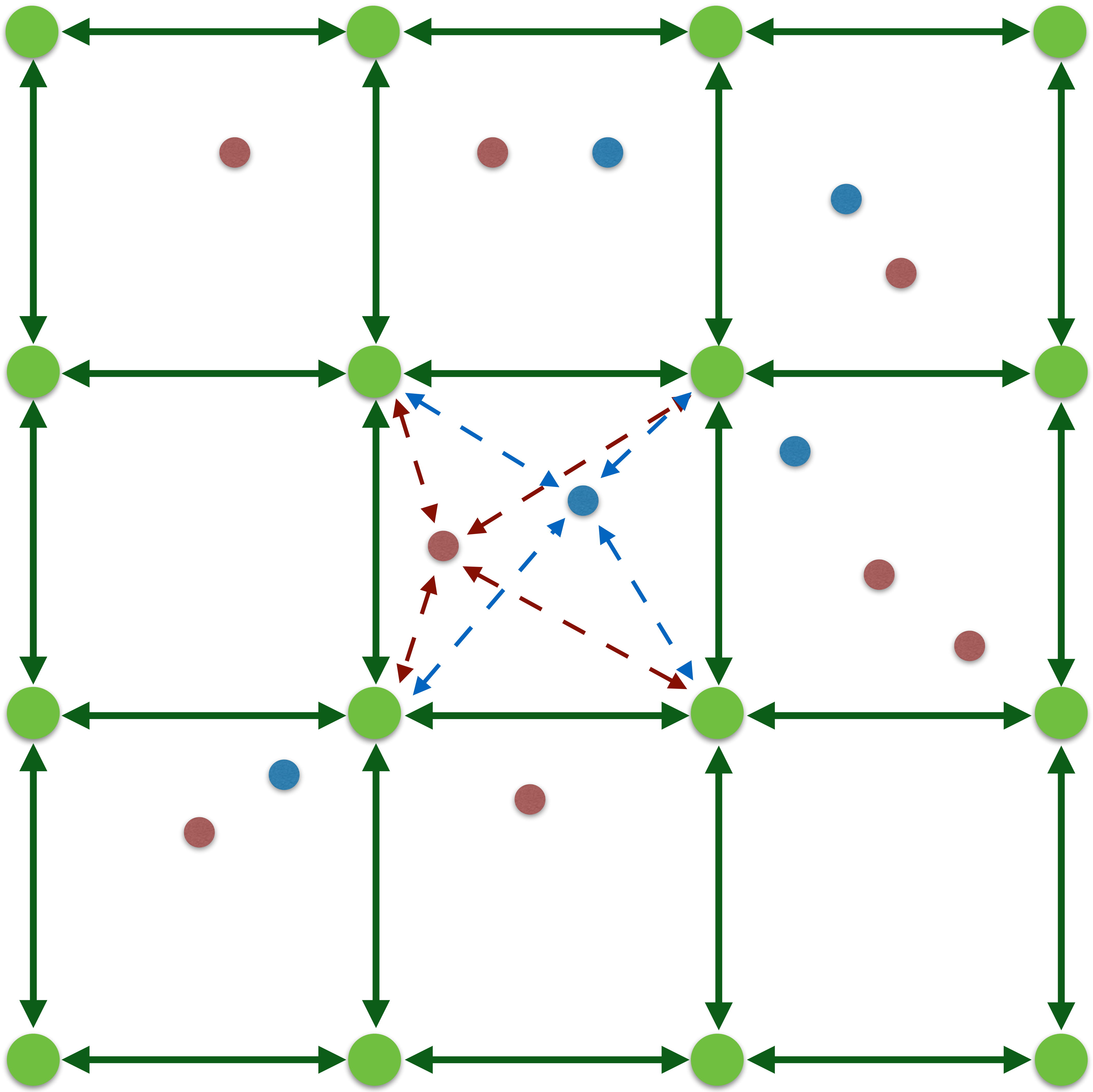


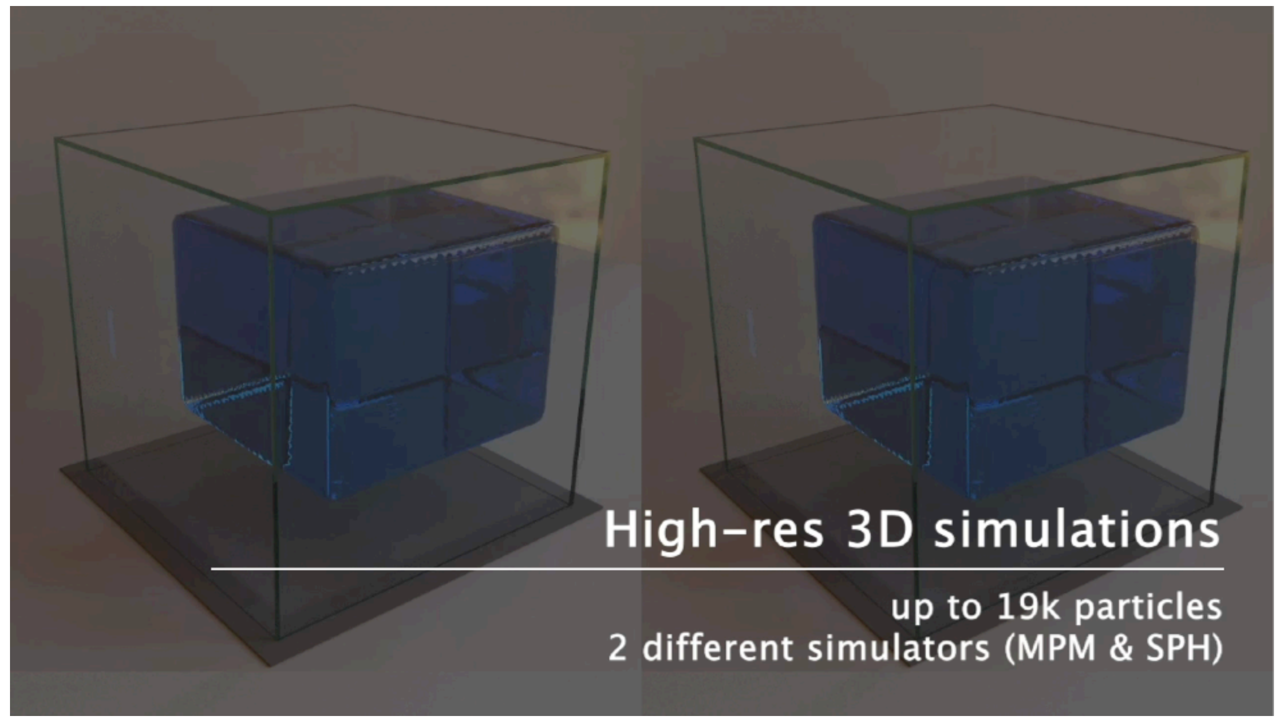


A. Sanchez-Gonzalez et al., ICML PMLR 8459–8468 (2020)
R. Lam et al., Science 382.6677 1416-1421 (2023)

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

PIC codes (and others) can be seen from a graph perspective





A. Sanchez-Gonzalez et al., ICML PMLR 8459–8468 (2020)
R. Lam et al., Science 382.6677 1416-1421 (2023)

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

The Breaking of Finite Amplitude Plasma Oscillations

by

John Dawson

June 1959

MATT - 4

VET-TES-TA-MEN-TUM NOV-UM

DEI SUB NUMINE VIGET

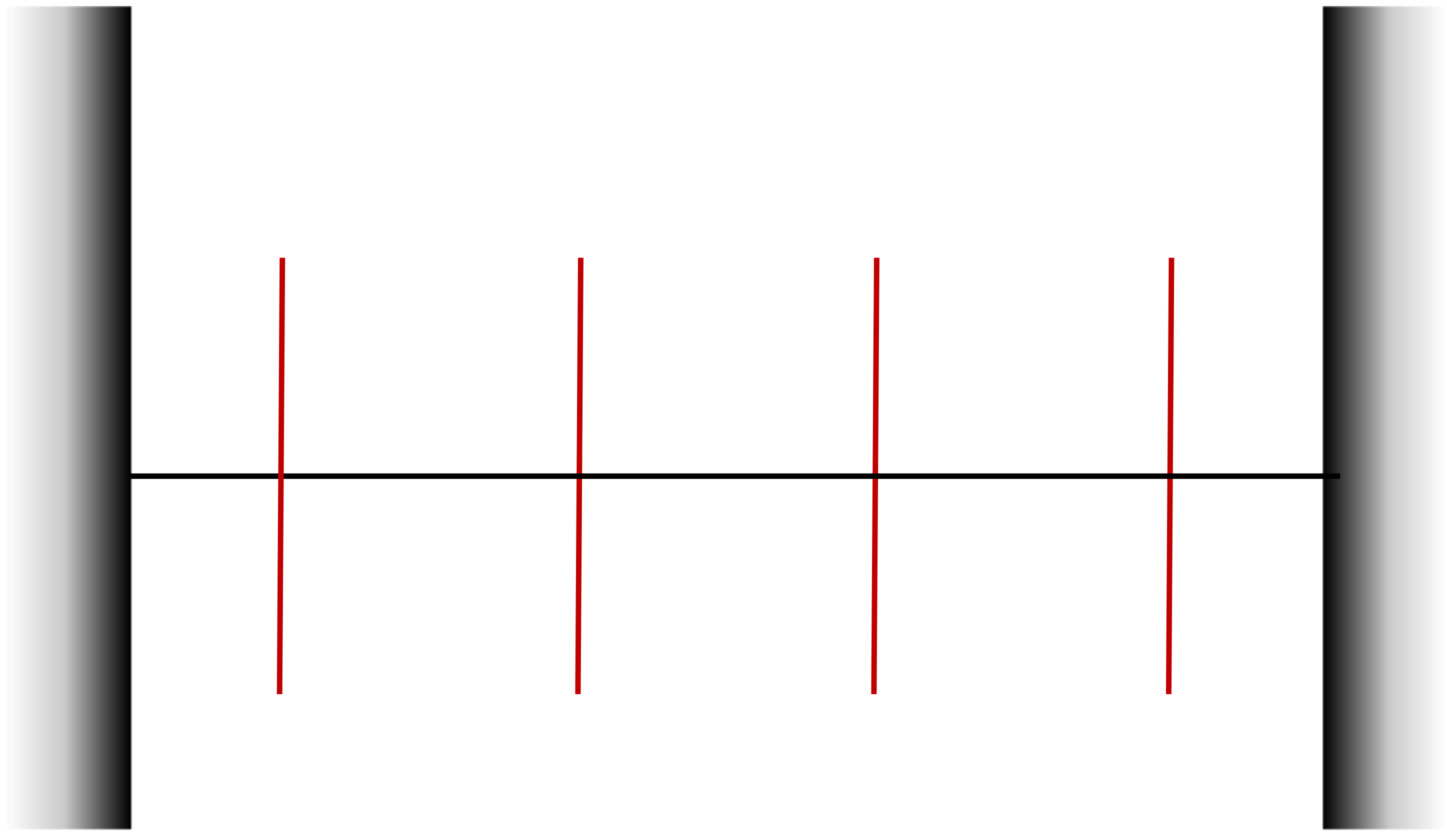
PROJECT MATTERHORN

Contract AT(30-1) – 1238 with the
US Atomic Energy Commission

AEC RESEARCH AND DEVELOPMENT REPORT

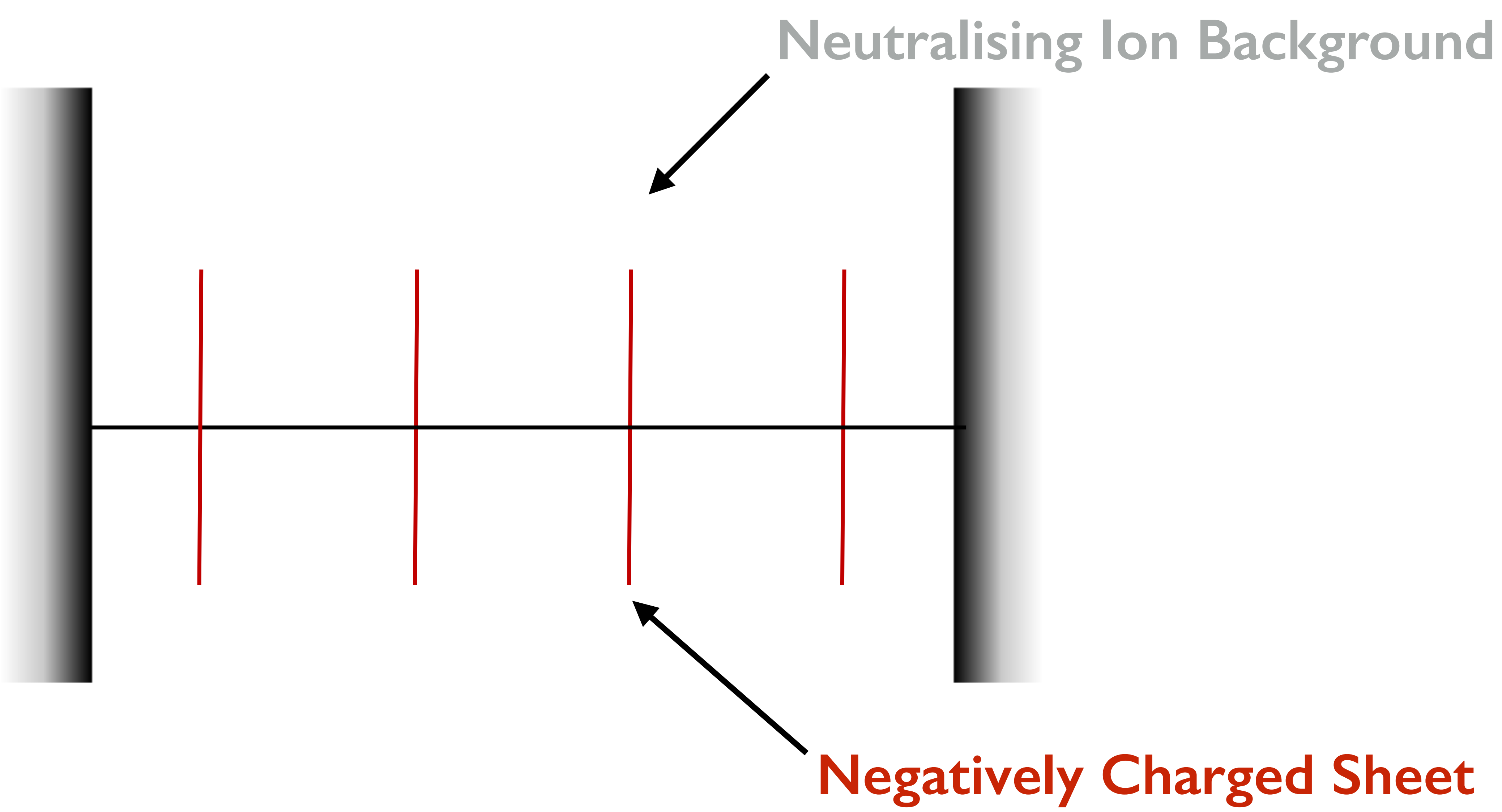
PRINCETON UNIVERSITY
PRINCETON, NEW JERSEY

1D Plasma Electrostatic Sheet Model



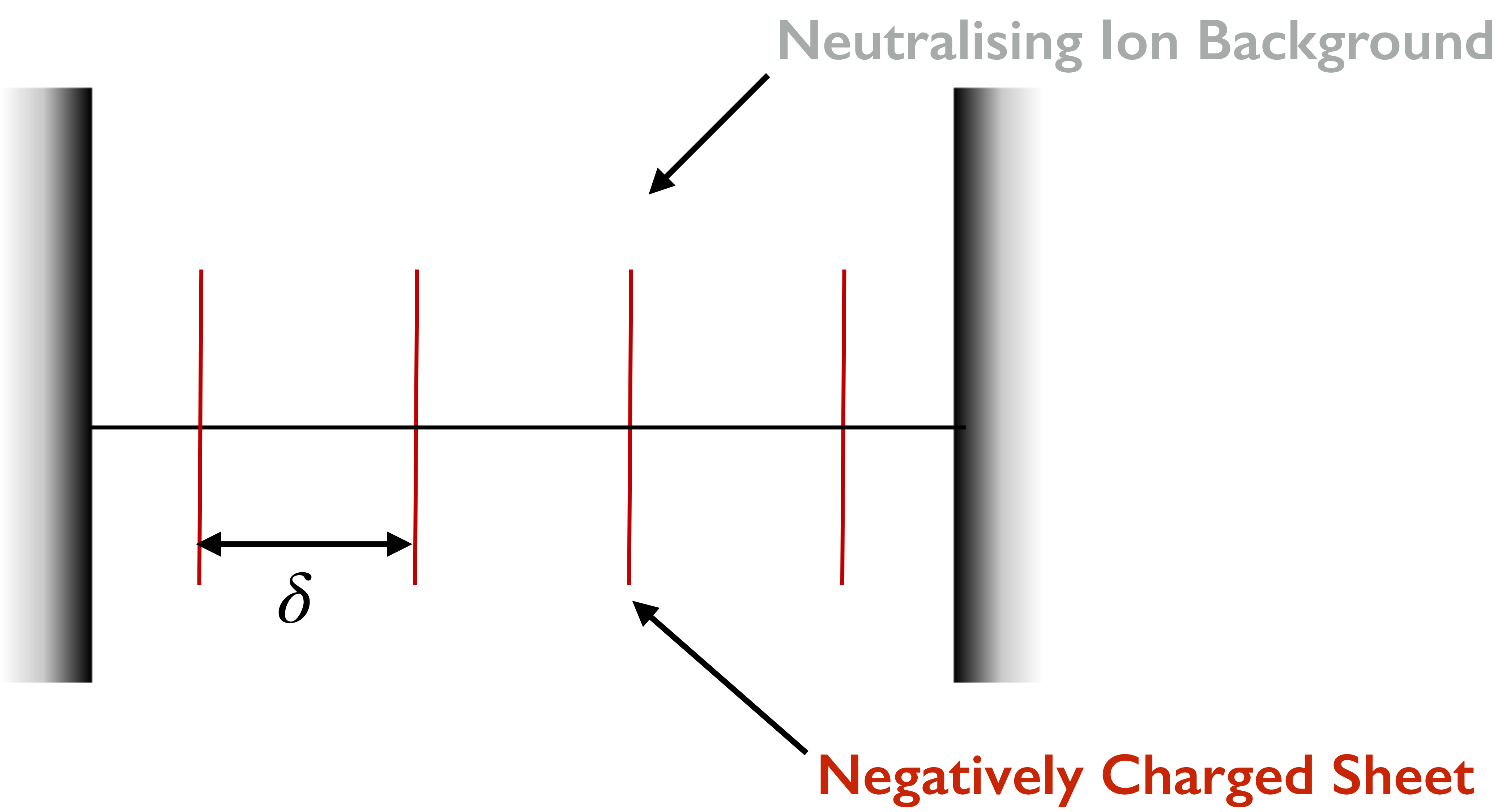
J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

1D Plasma Electrostatic Sheet Model



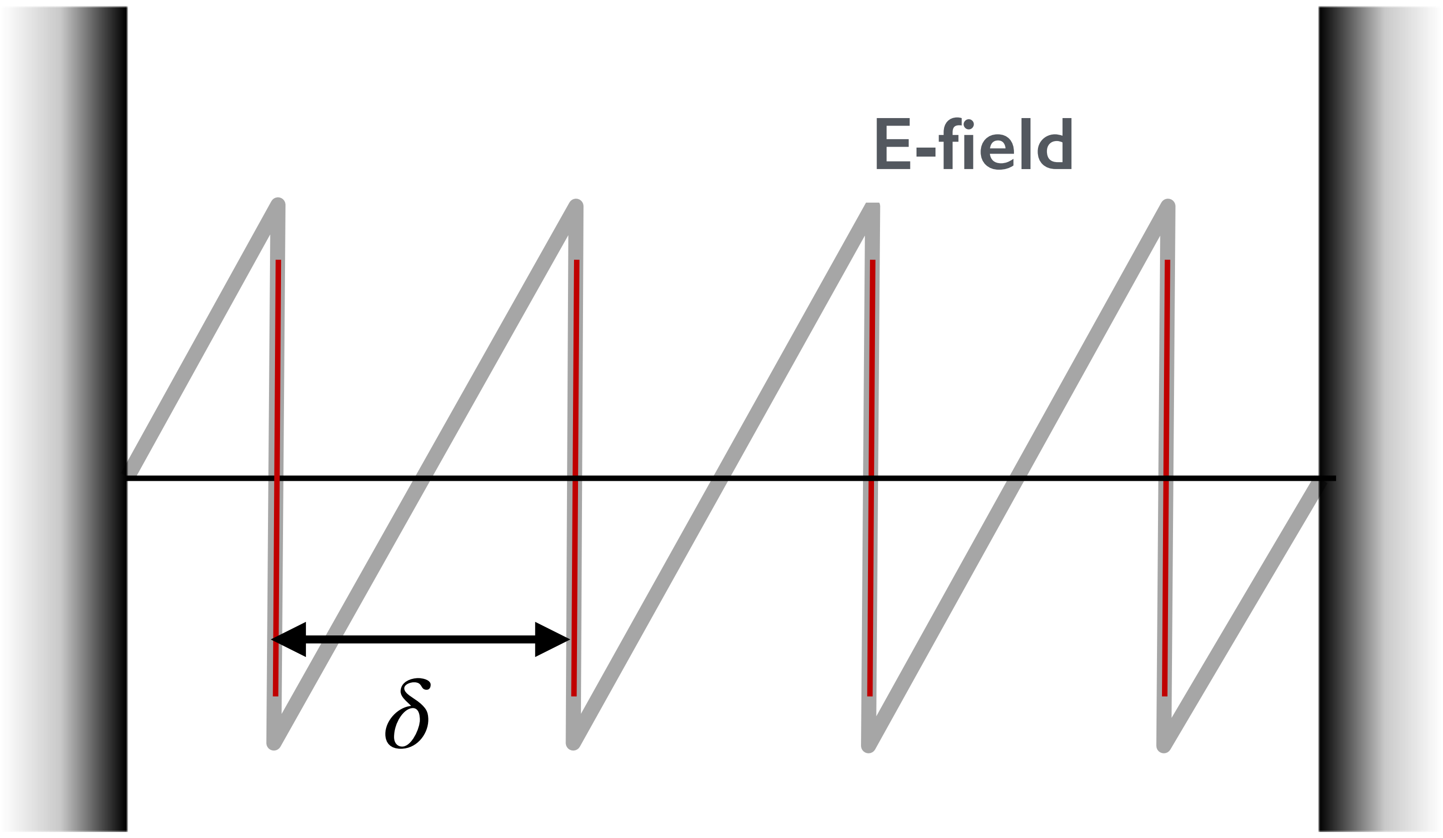
J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

1D Plasma Electrostatic Sheet Model



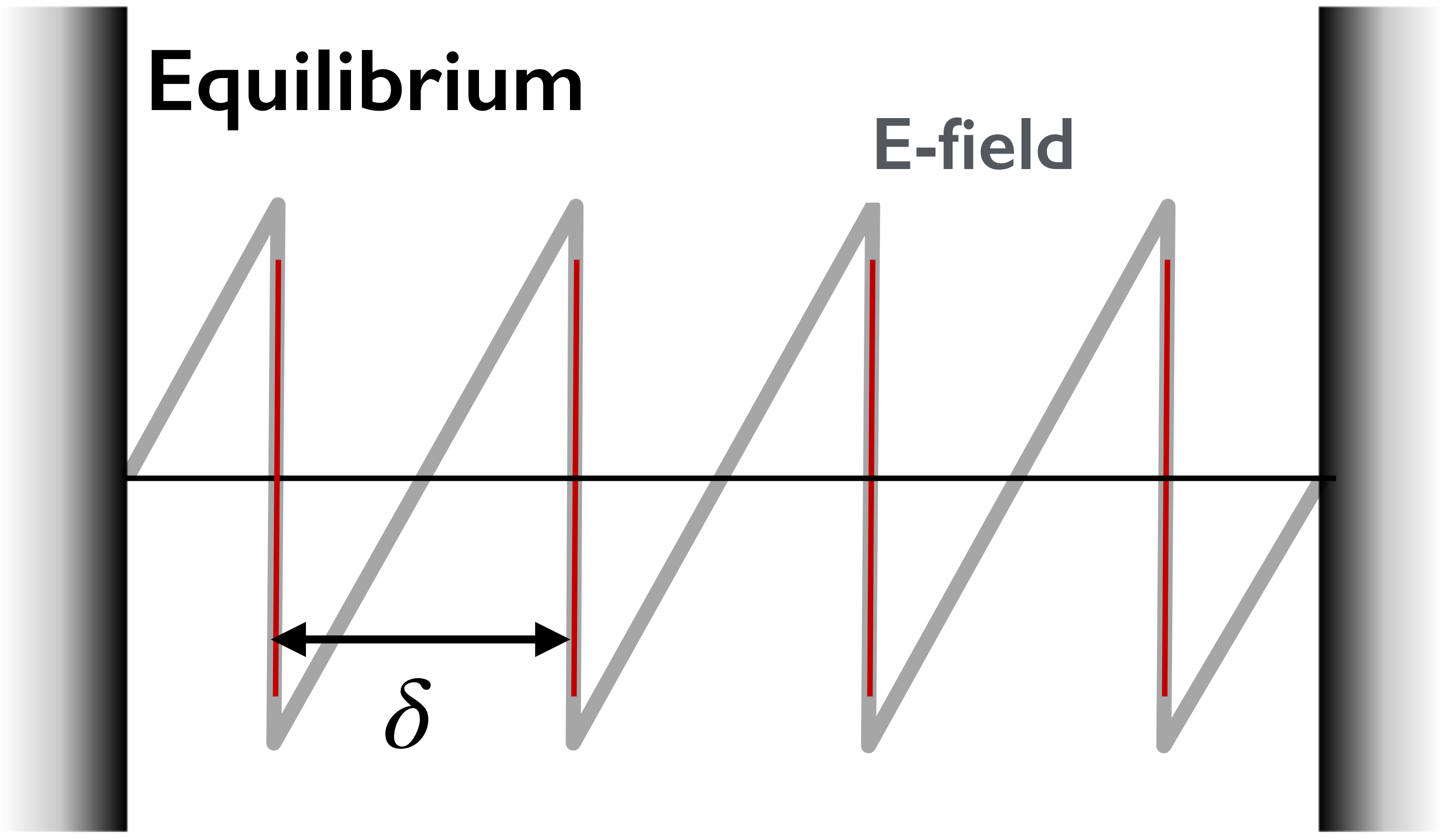
J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

1D Plasma Electrostatic Sheet Model



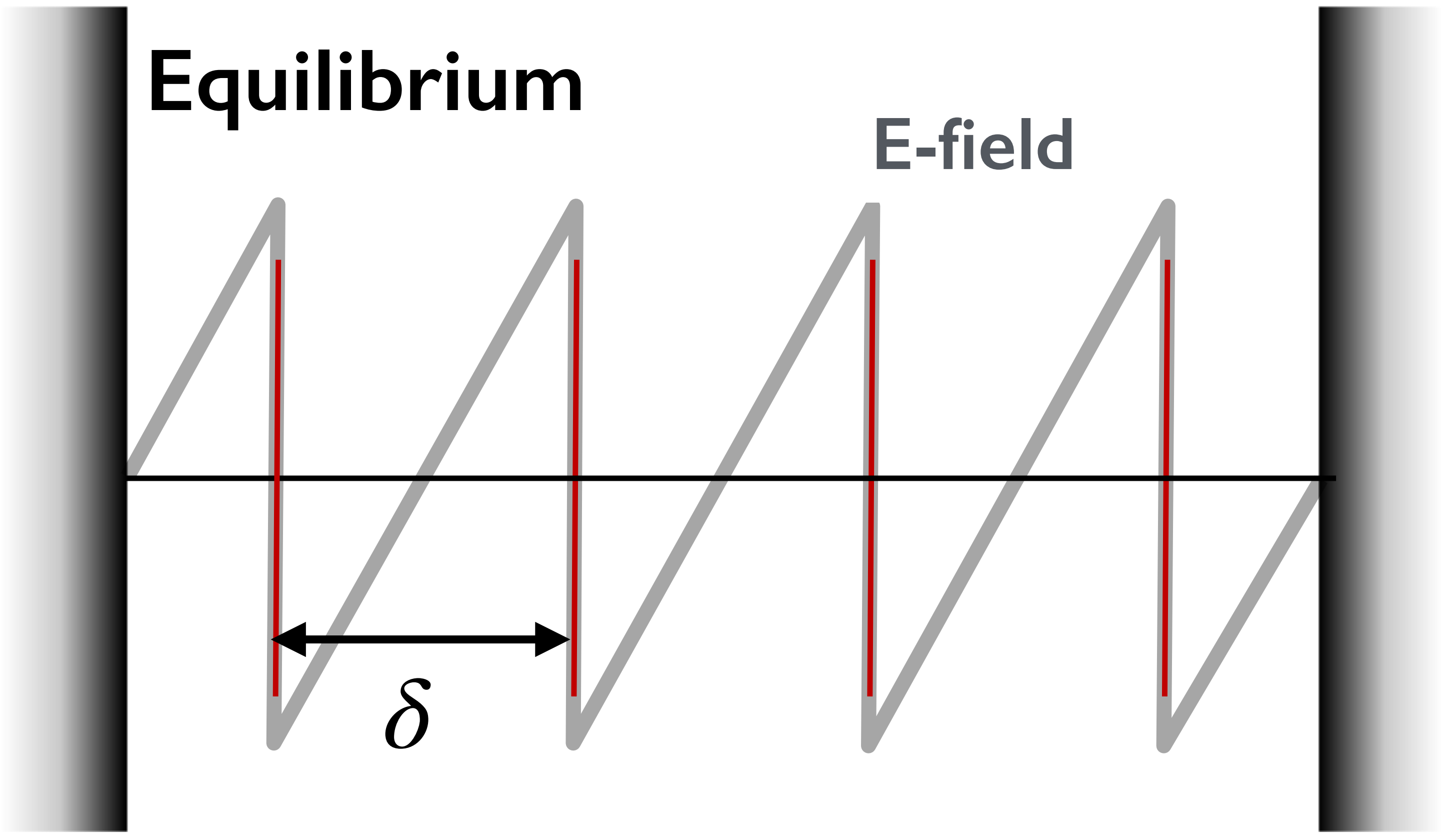
J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

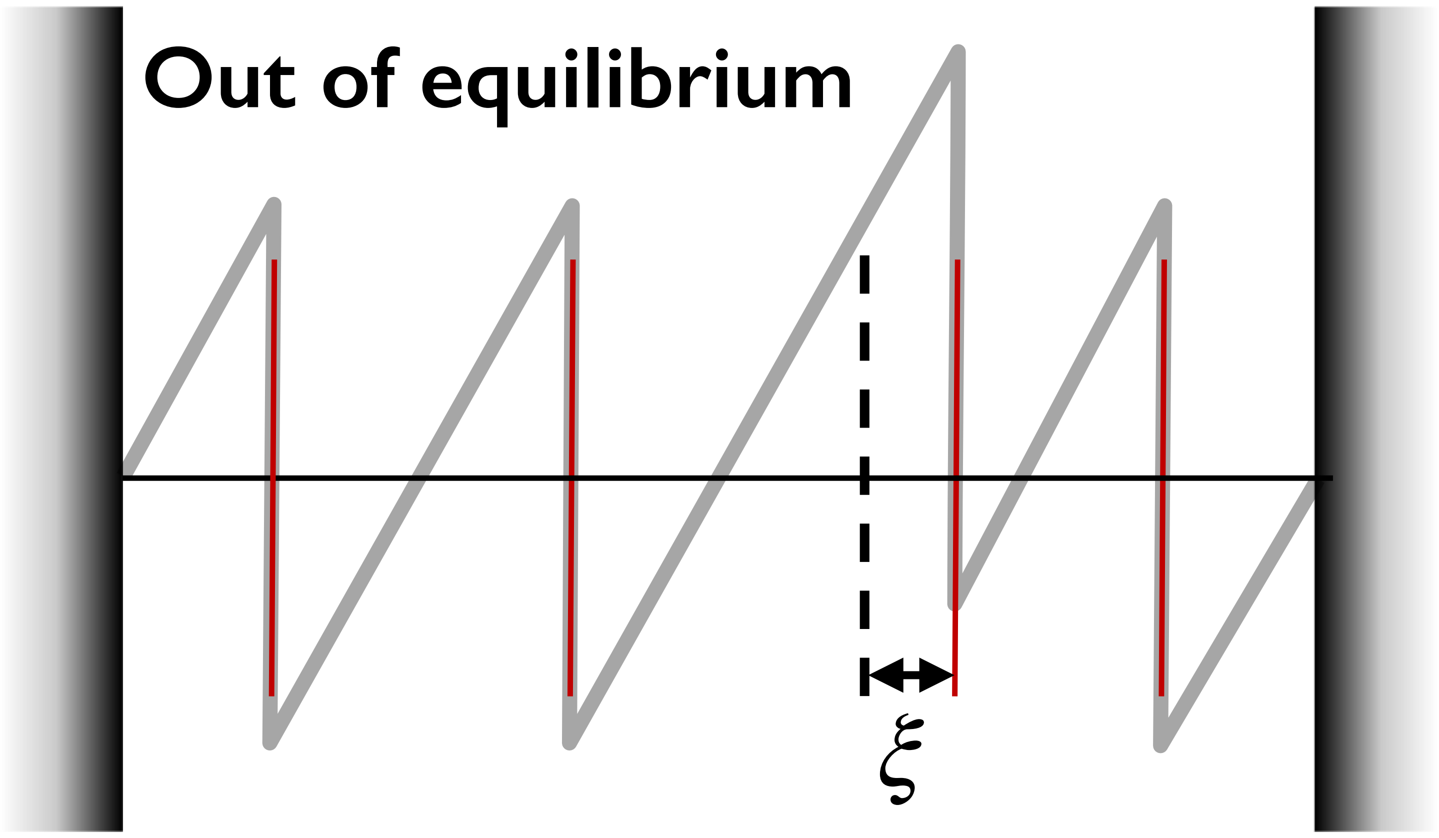
1D Plasma Electrostatic Sheet Model



J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

1D Plasma Electrostatic Sheet Model

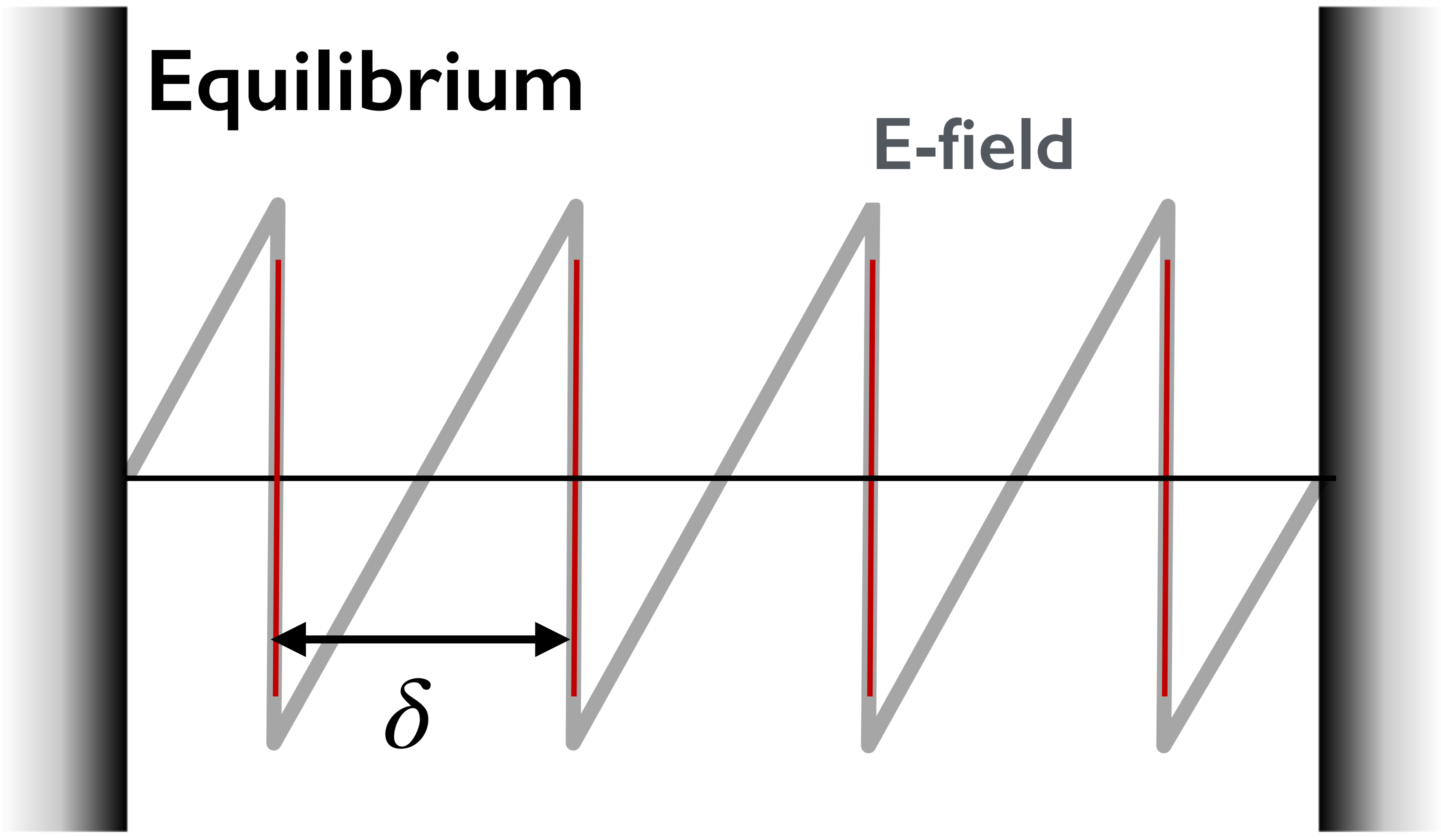


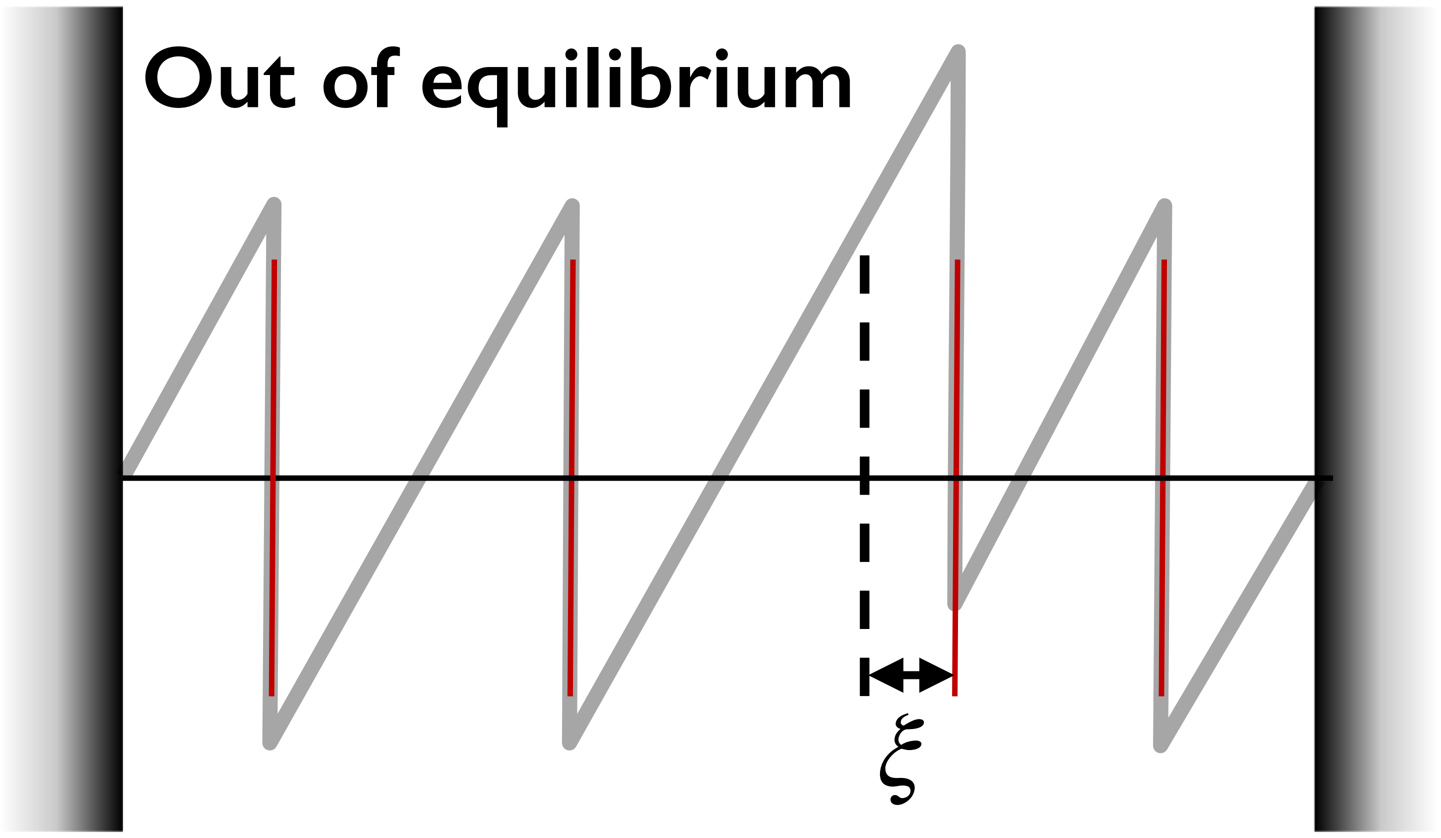


$$\ddot{\xi} = -\frac{4\pi e^2 n_0}{m_e}\xi = -\omega_p^2 \xi$$

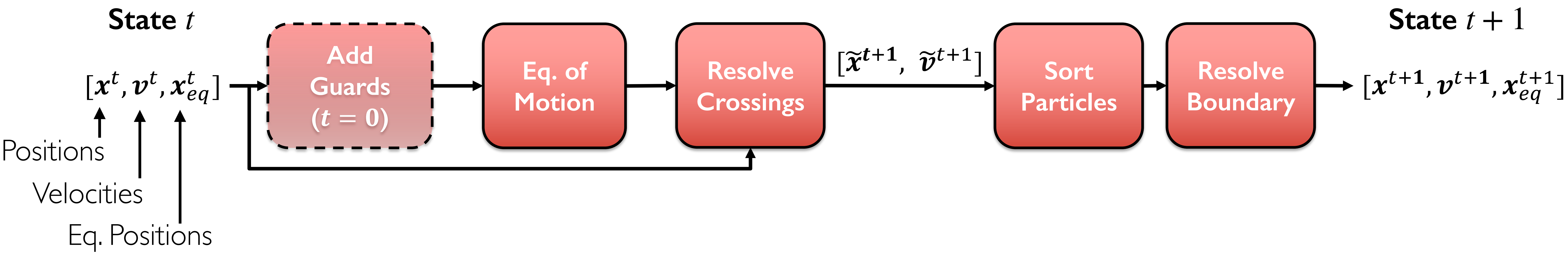
J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

1D Plasma Electrostatic Sheet Model



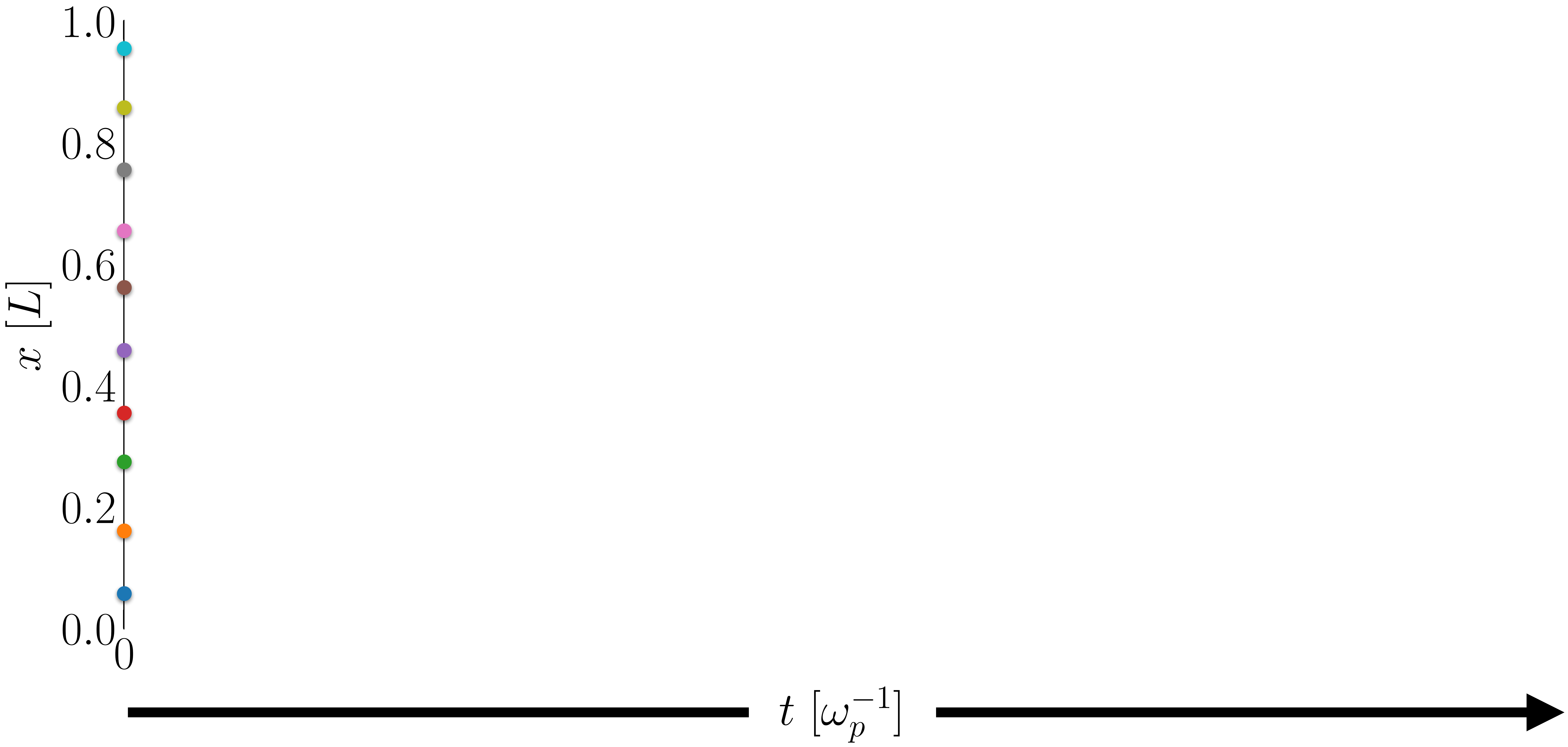


$$\ddot{\xi} = -\frac{4\pi e^2 n_0}{m_e}\xi = -\omega_p^2 \xi$$

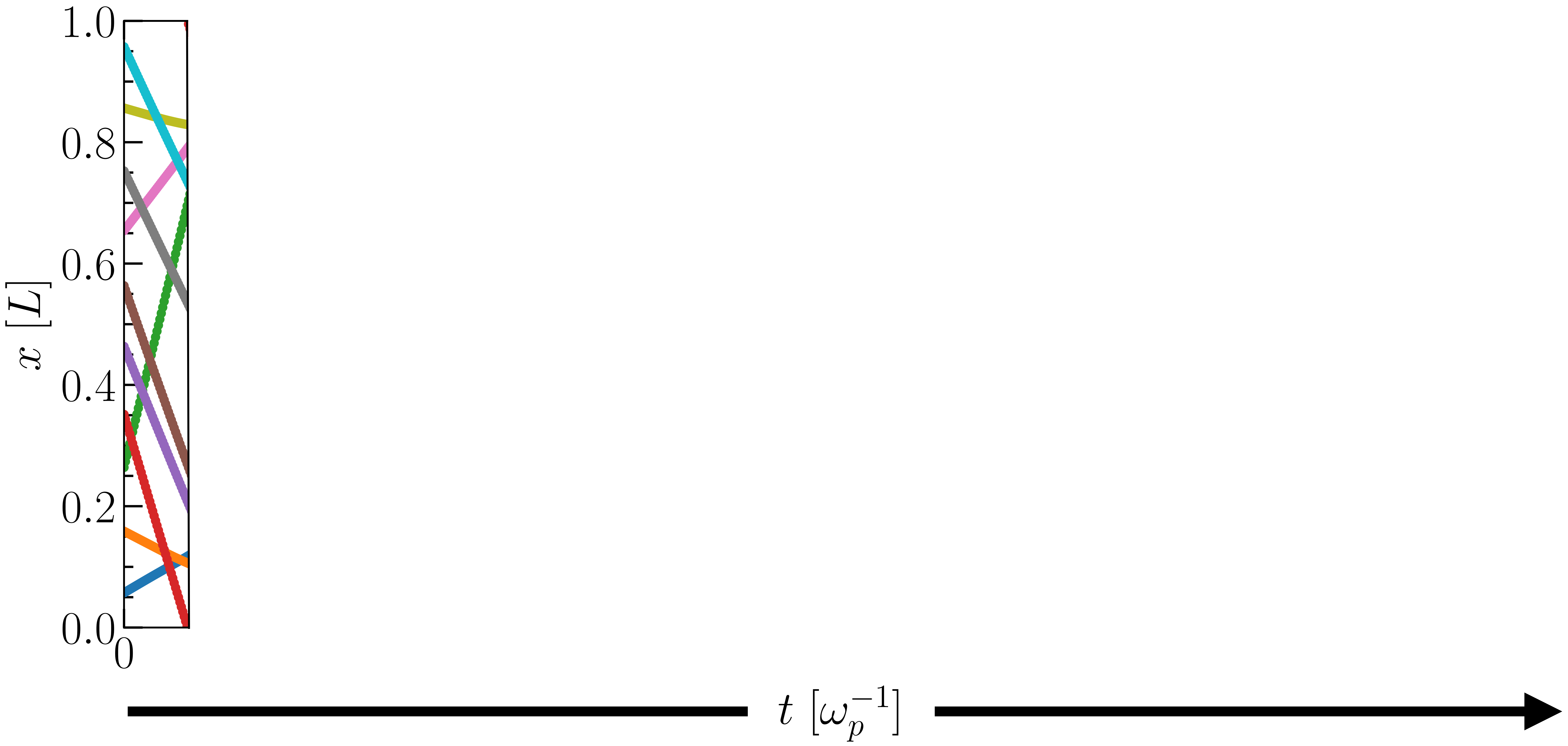


J. Dawson, Phys. Fluids 5.4, 445-459 (1962)
J. Dawson, Methods in Computational Physics 9, 1–28 (1970)

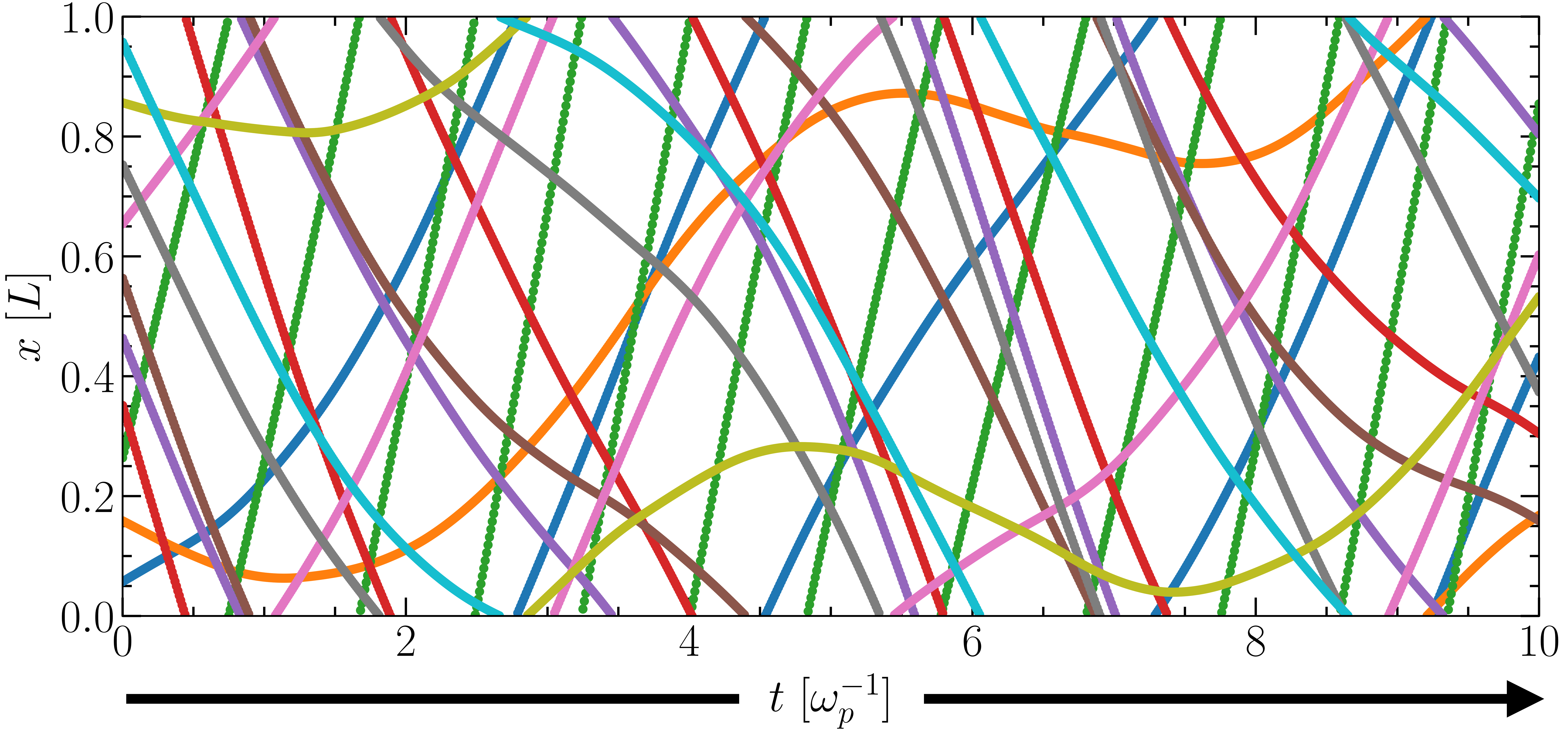
Example of Simulation



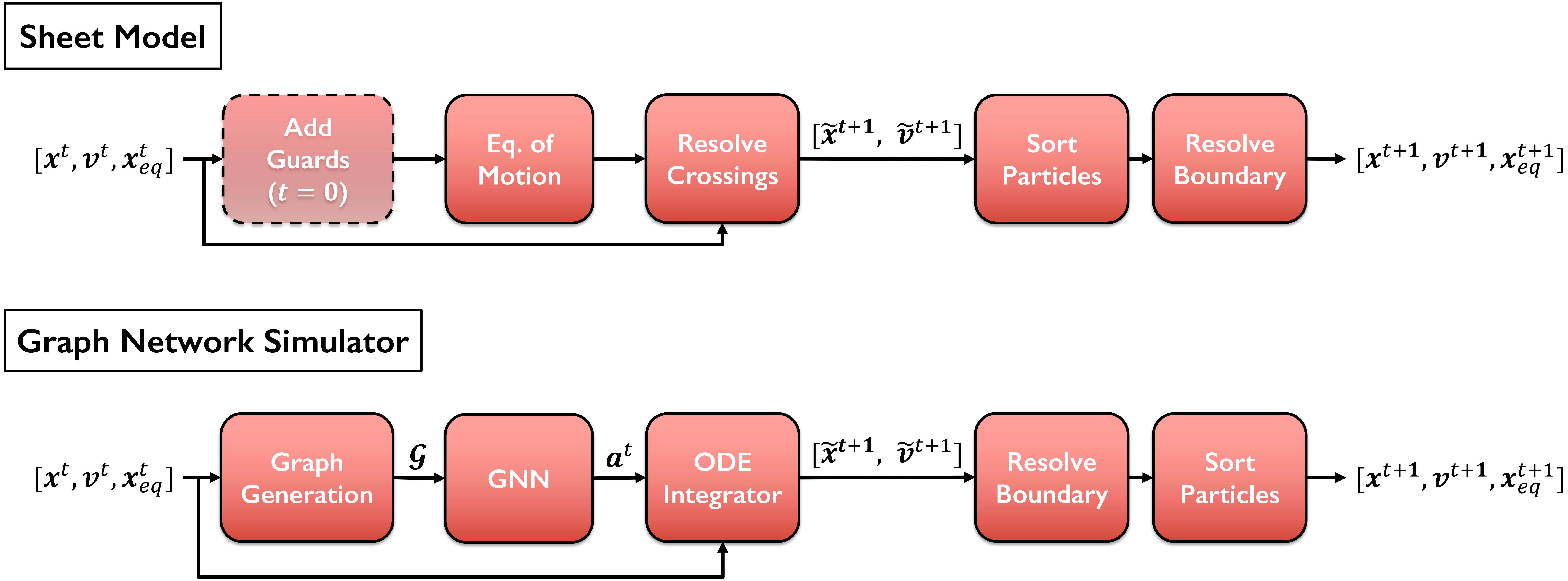
Example of Simulation



Example of Simulation



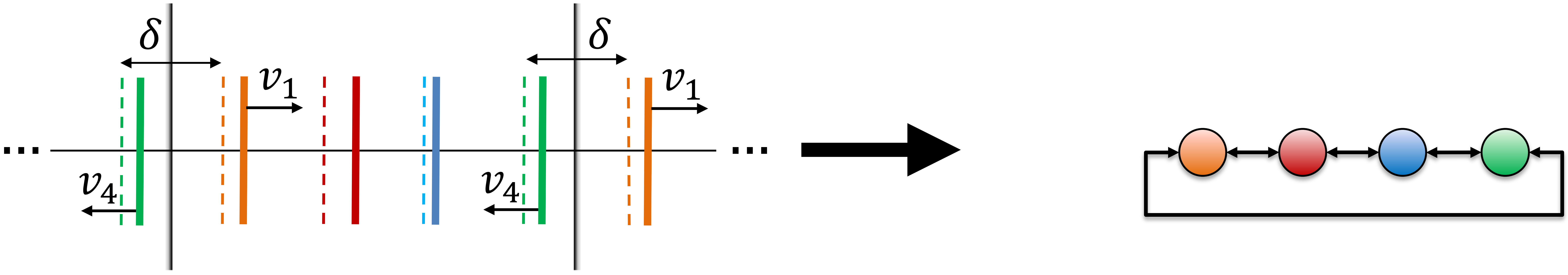
1D Plasma ESM Graph Network Simulator



J. Dawson, Methods in Computational Physics 9, 1–28 (1970)
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

How do we represent the plasma as a graph?

Periodic Boundaries



Node (i)

$$\mathbf{n}_i^t = \left[\xi_i^t \ , \ v_i^t\right]$$

Edge ($i \rightarrow j$)

$$\mathbf{r}_{ji}^t = \left[x_i^t - x_j^t\right]$$

Target

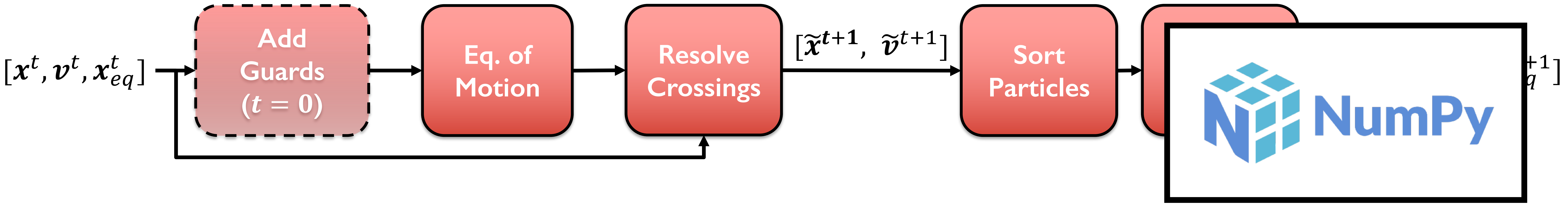
$$a_i^t = \frac{v_i^{t+1} - v_i^t}{\Delta t}$$

All values are normalised to the intersheet spacing δ

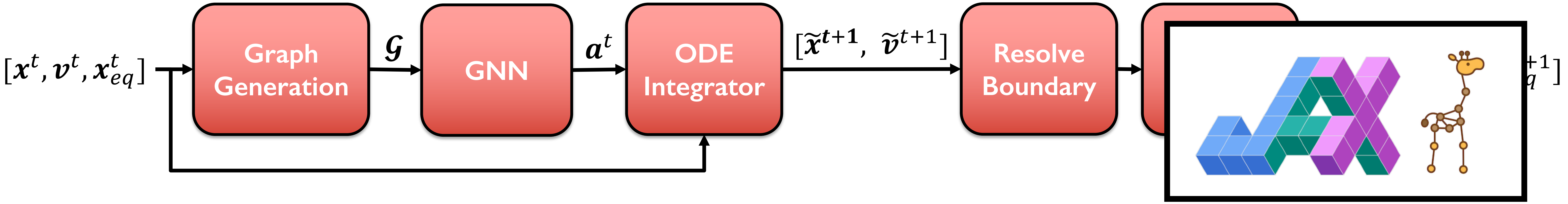
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

1D Plasma ESM Graph Network Simulator

Sheet Model



Graph Network Simulator



Code: https://github.com/diogodcarvalho/gns-sheet-model

https://github.com/google/jax

https://github.com/deepmind/jraph

J. Dawson, Methods in Computational Physics 9, 1–28 (1970)
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS generalizes to different number of sheets and boundary conditions

Trained on **subsampled high temporal resolution data** $\left(\Delta t_{orig} = 10^{-4}\ \omega_p^{-1}\right)$ of **10 sheets** moving inside a **periodic box** $\left(t_{sim} = 10\ \omega_p^{-1}\right)$

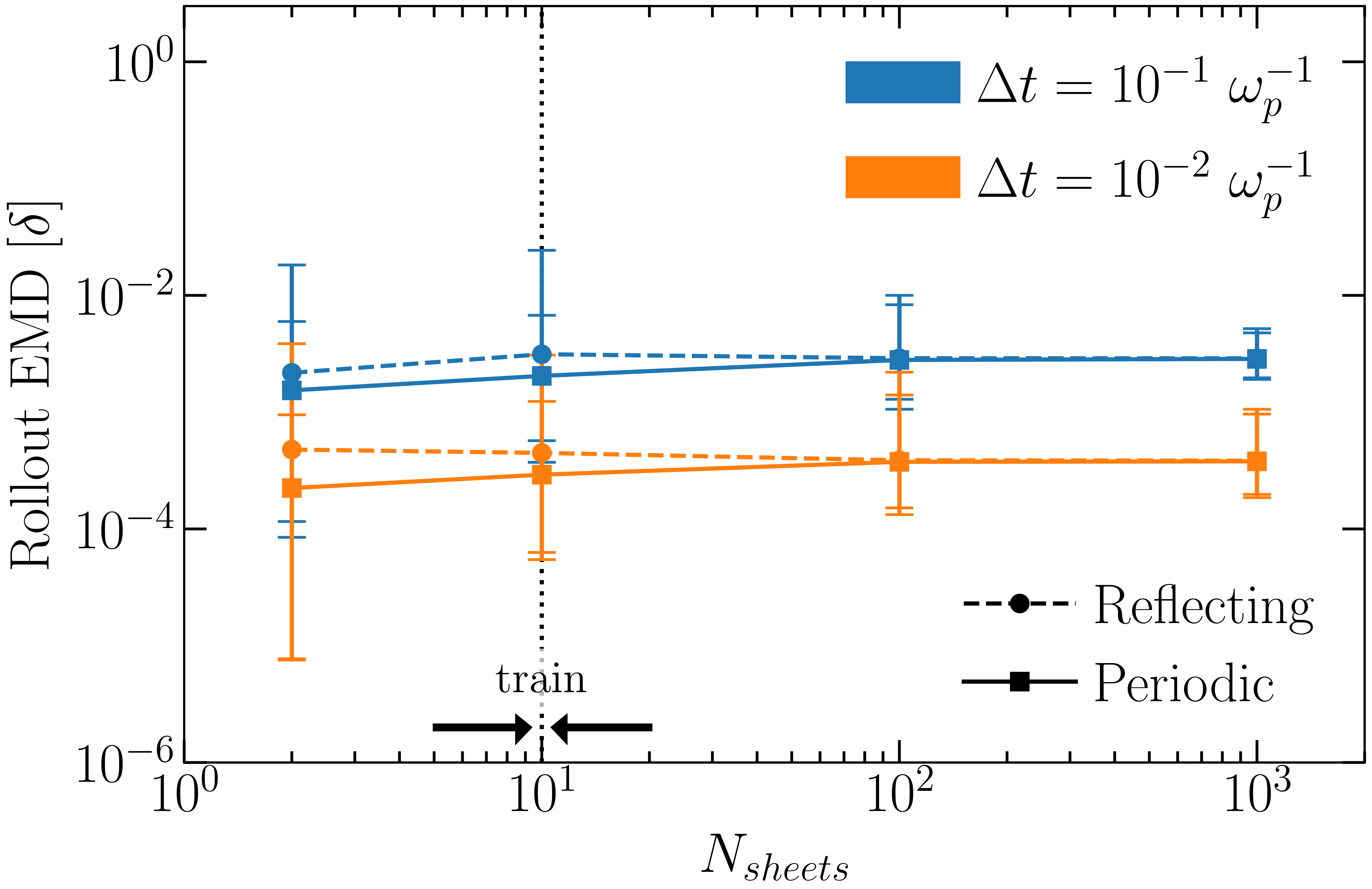
Initial positions and velocities are randomly sampled from a uniform distribution

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS generalizes to different number of sheets and boundary conditions

Trained on **subsampled high temporal resolution data** $\left(\Delta t_{orig} = 10^{-4}\ \omega_p^{-1}\right)$ of **10 sheets** moving inside a **periodic box** $\left(t_{sim} = 10\ \omega_p^{-1}\right)$

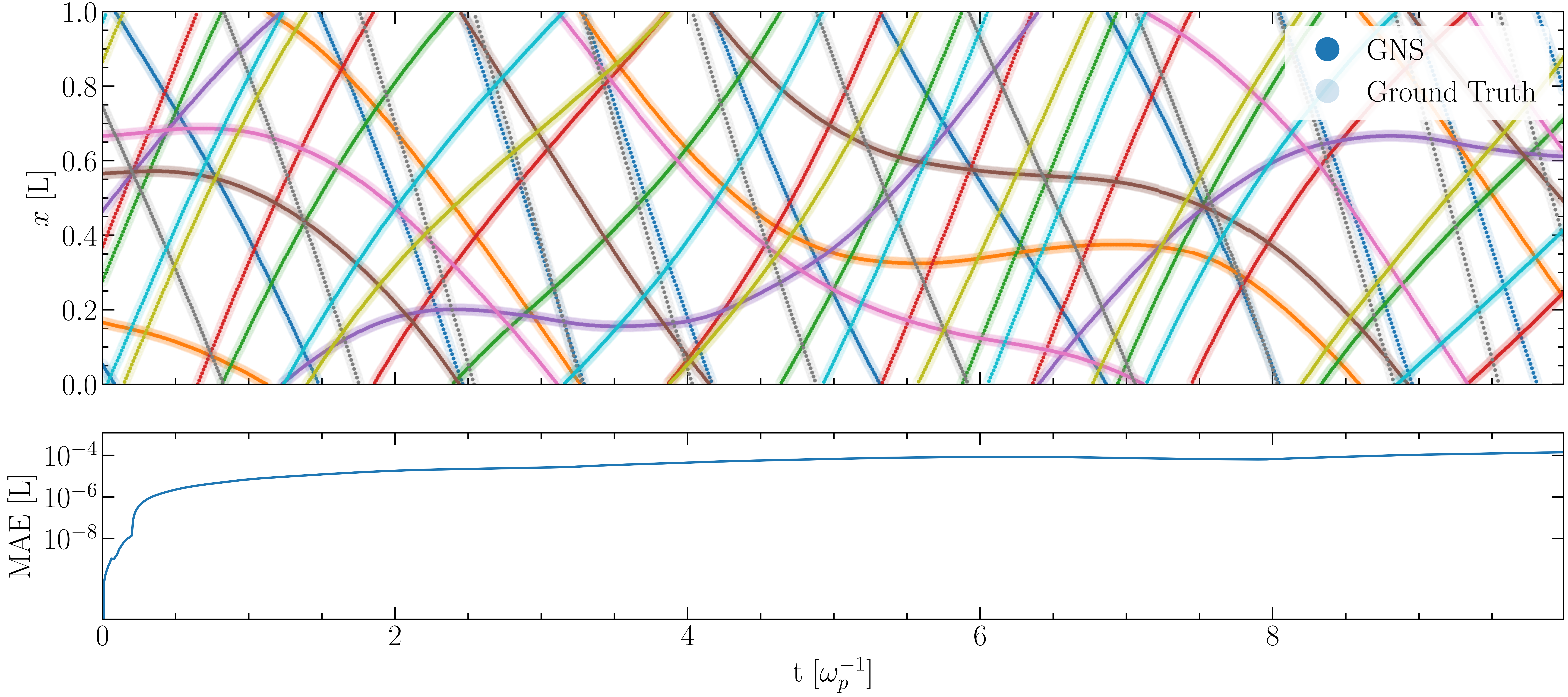
Initial positions and velocities are randomly sampled from a uniform distribution



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS rollout errors are very small

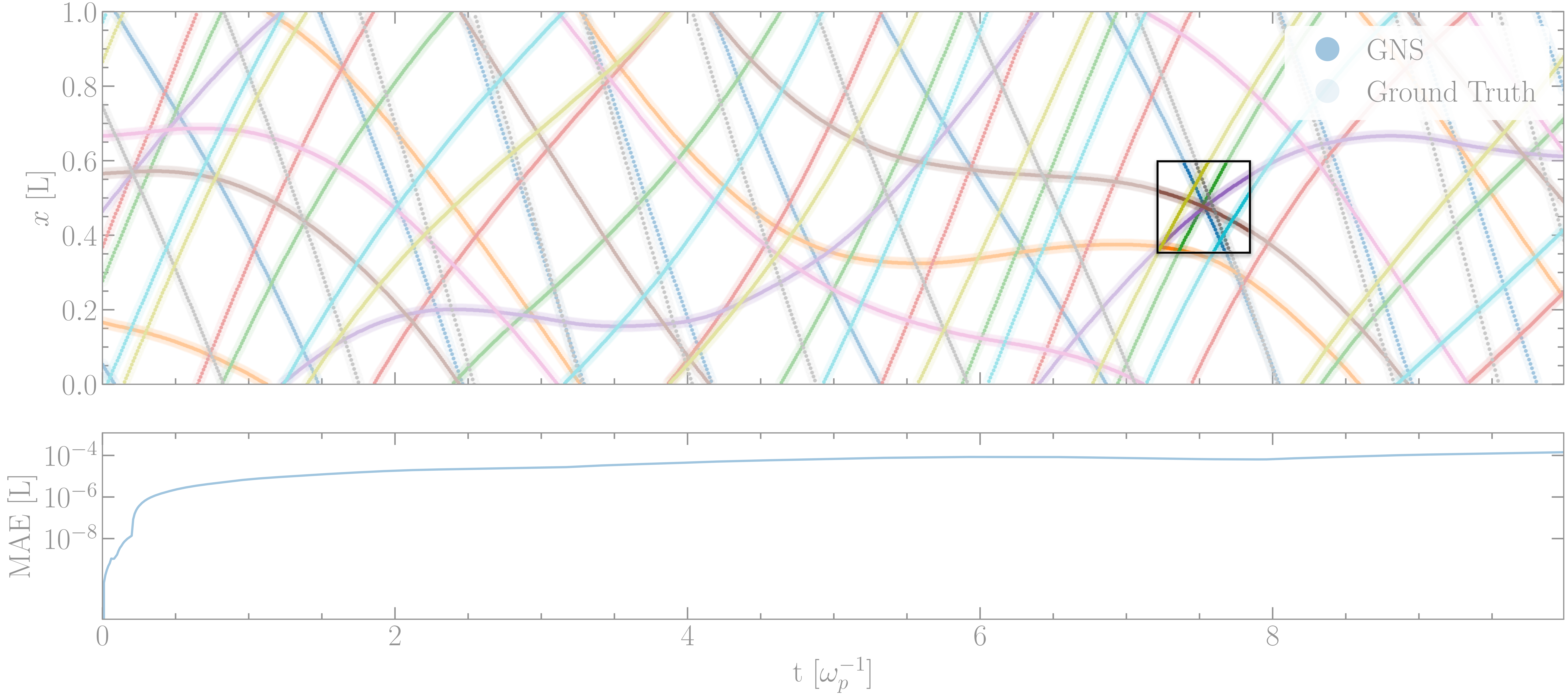
$\Delta t = 10^{-2}\ \omega_p^{-1}$ Rollout MAE $= 5.6 \times 10^{-4}\ \delta$



Example shown corresponds to the worst rollout error observed in the test set

GNS rollout errors are very small

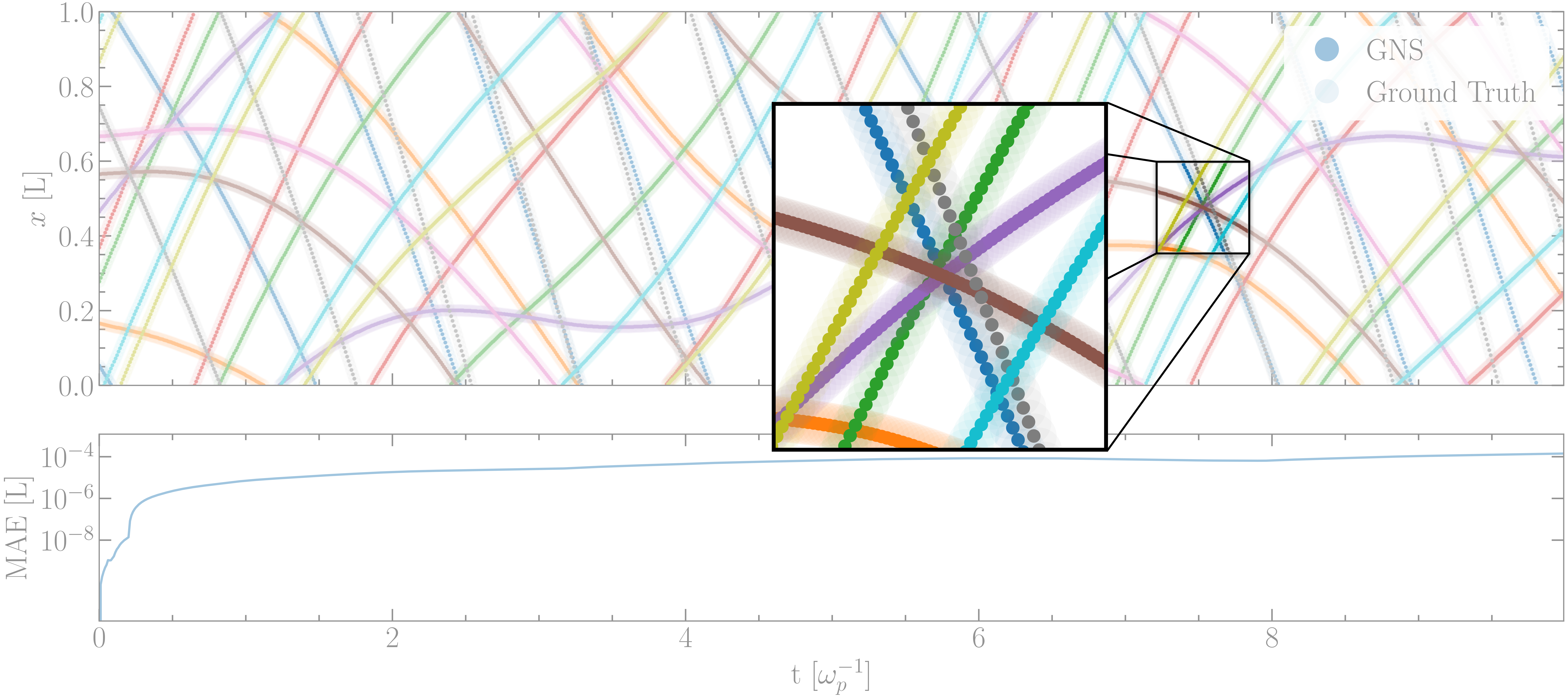
$\Delta t = 10^{-2}\ \omega_p^{-1}$ Rollout MAE $= 5.6 \times 10^{-4}\ \delta$



Example shown corresponds to the worst rollout error observed in the test set

GNS rollout errors are very small

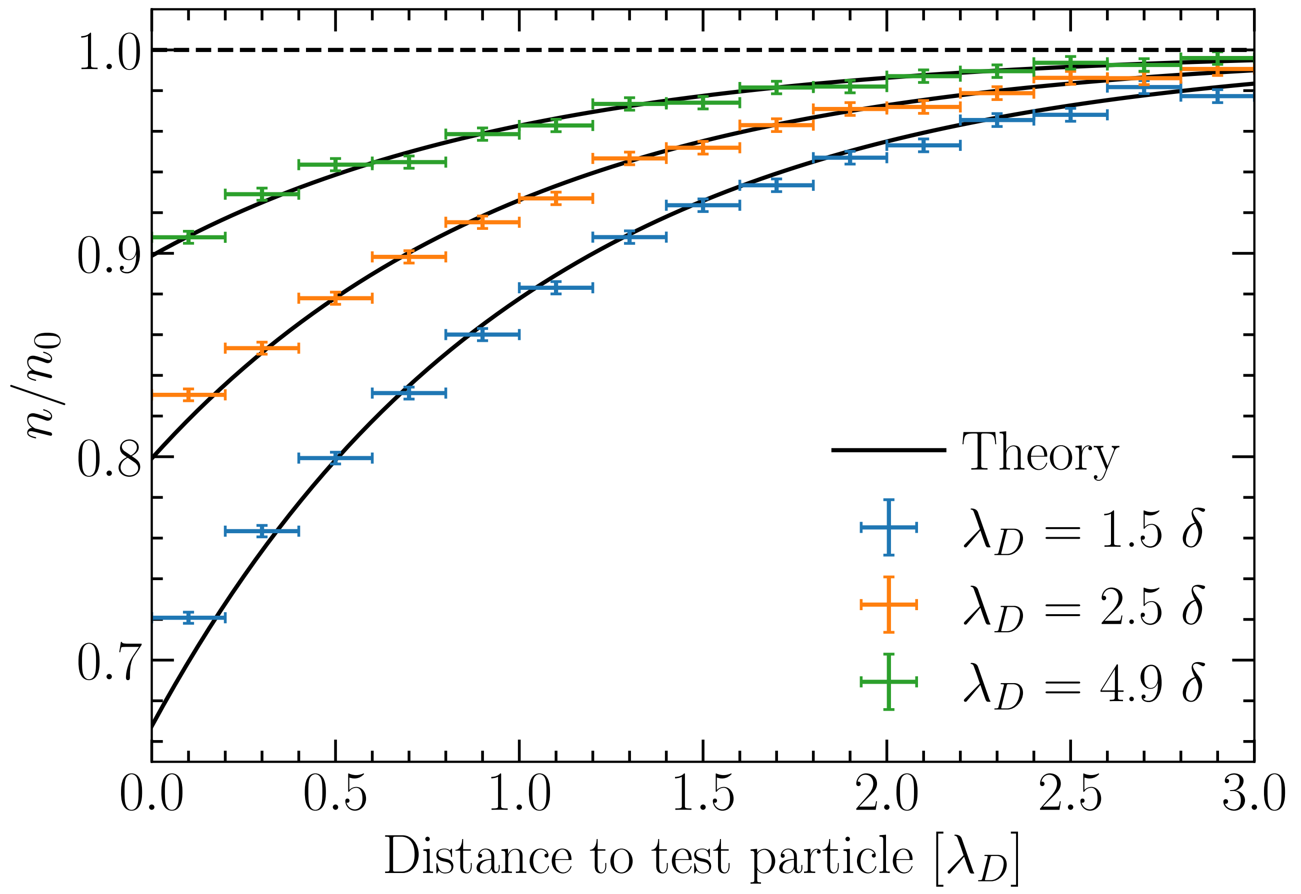
$\Delta t = 10^{-2}\ \omega_p^{-1}$ Rollout MAE $= 5.6 \times 10^{-4}\ \delta$



Example shown corresponds to the worst rollout error observed in the test set

GNS recovers a broad range of kinetic plasma processes

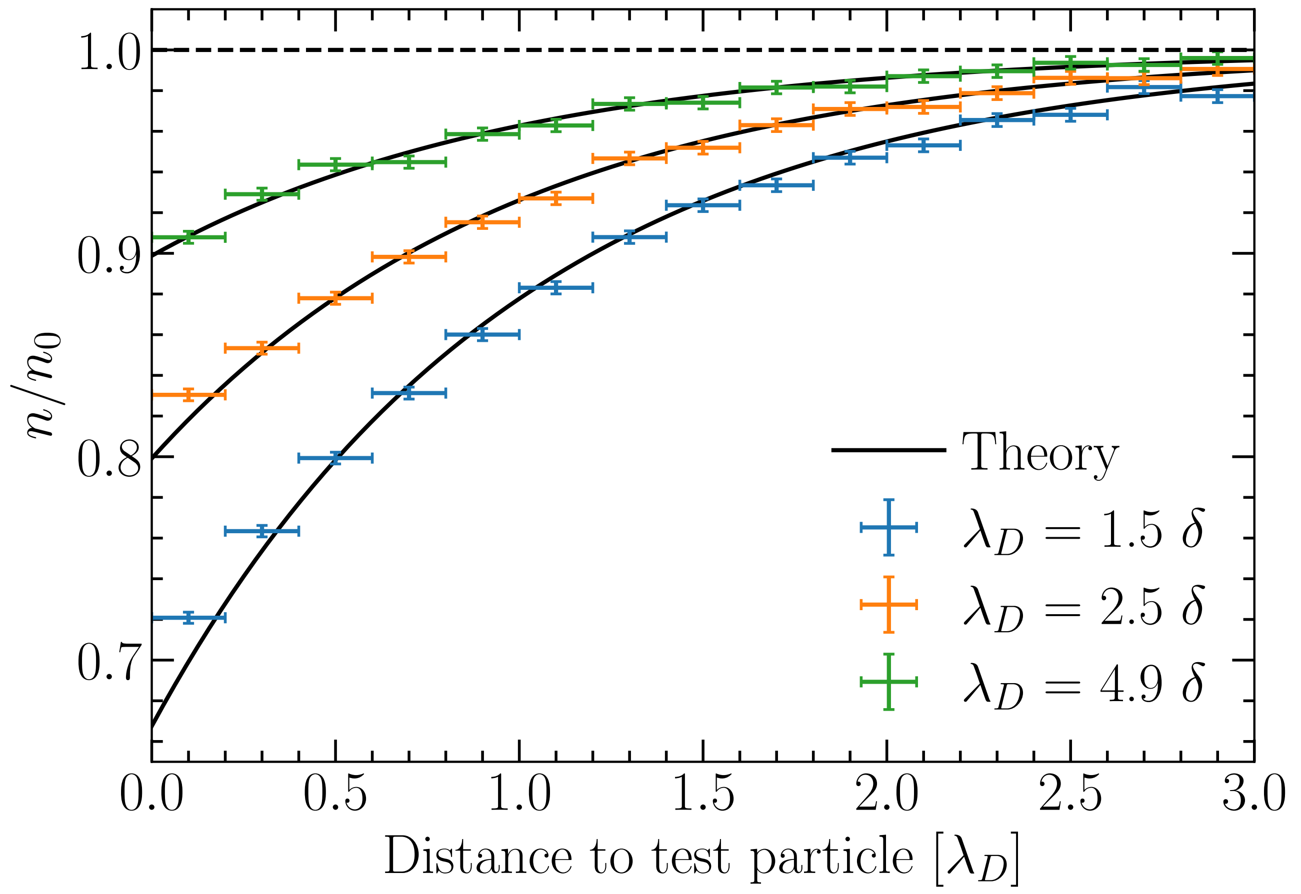
Debye Shielding



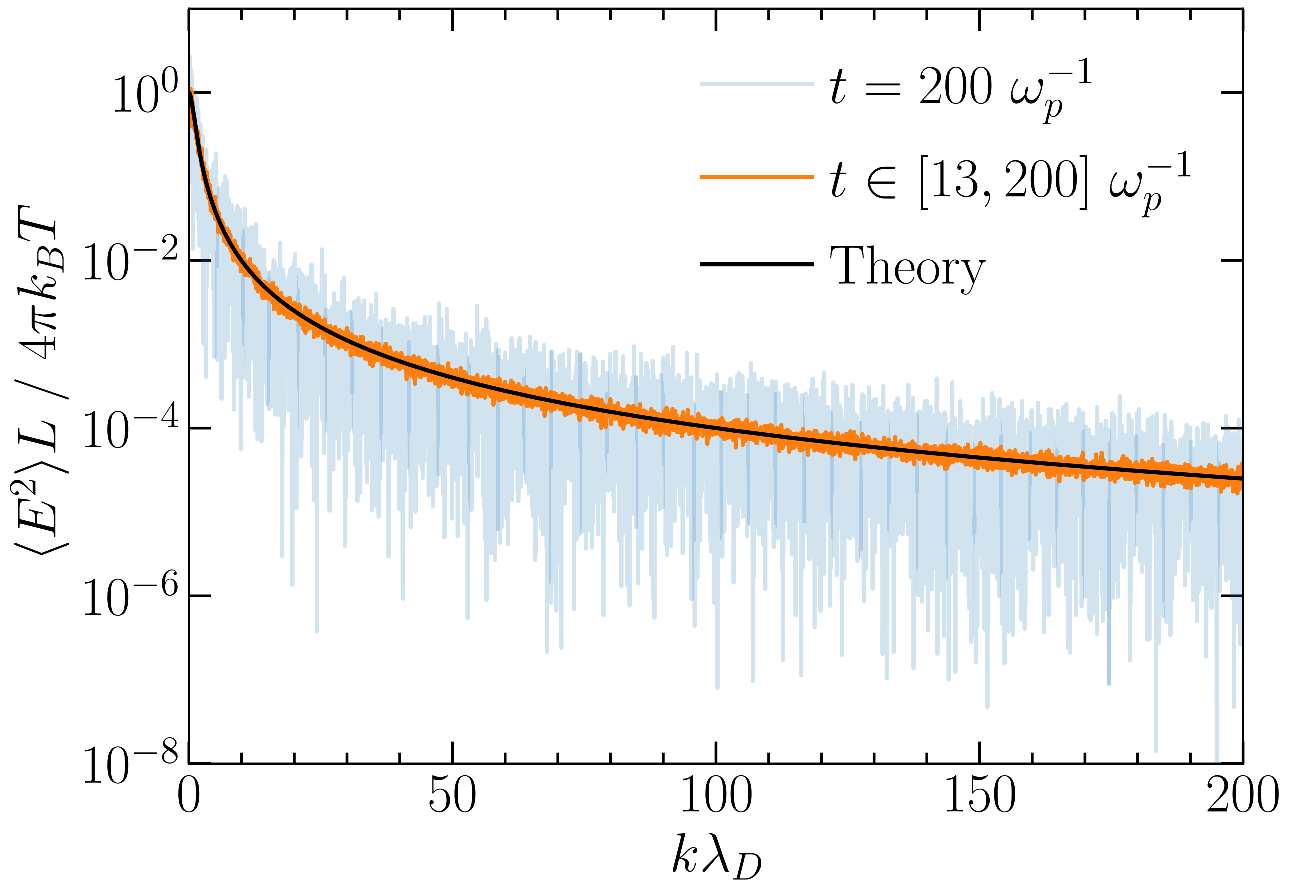
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers a broad range of kinetic plasma processes

Debye Shielding



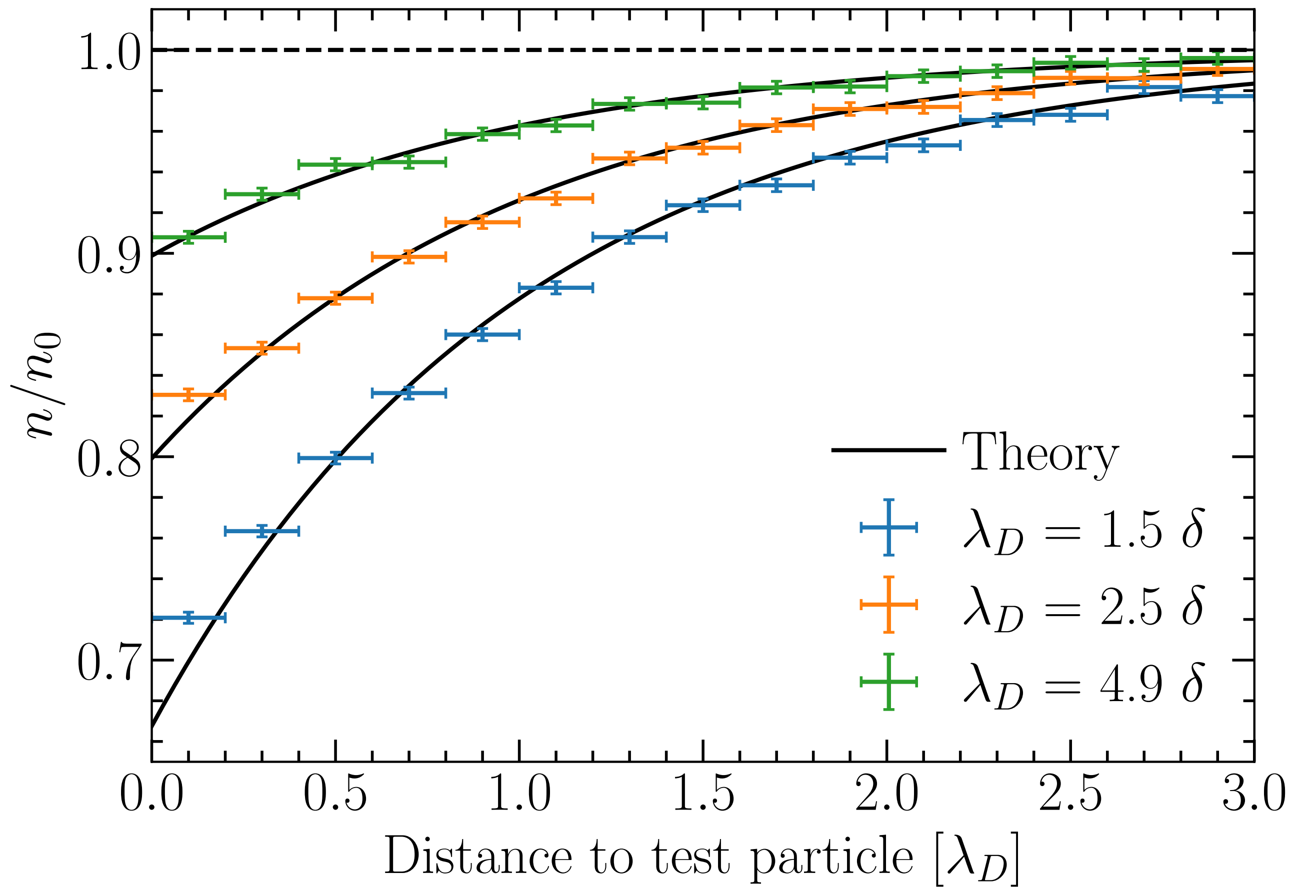
Electrostatic Fluctuations



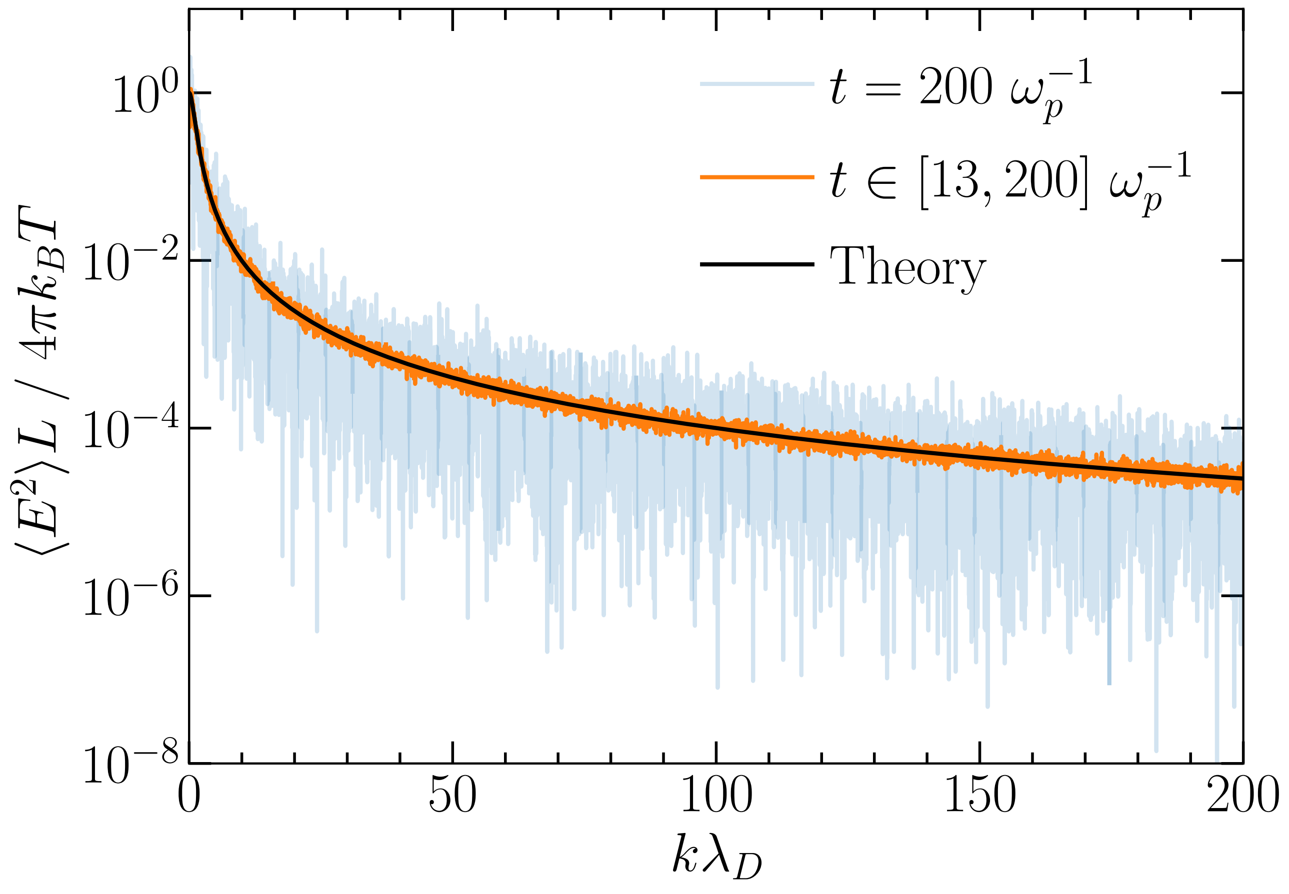
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers a broad range of kinetic plasma processes

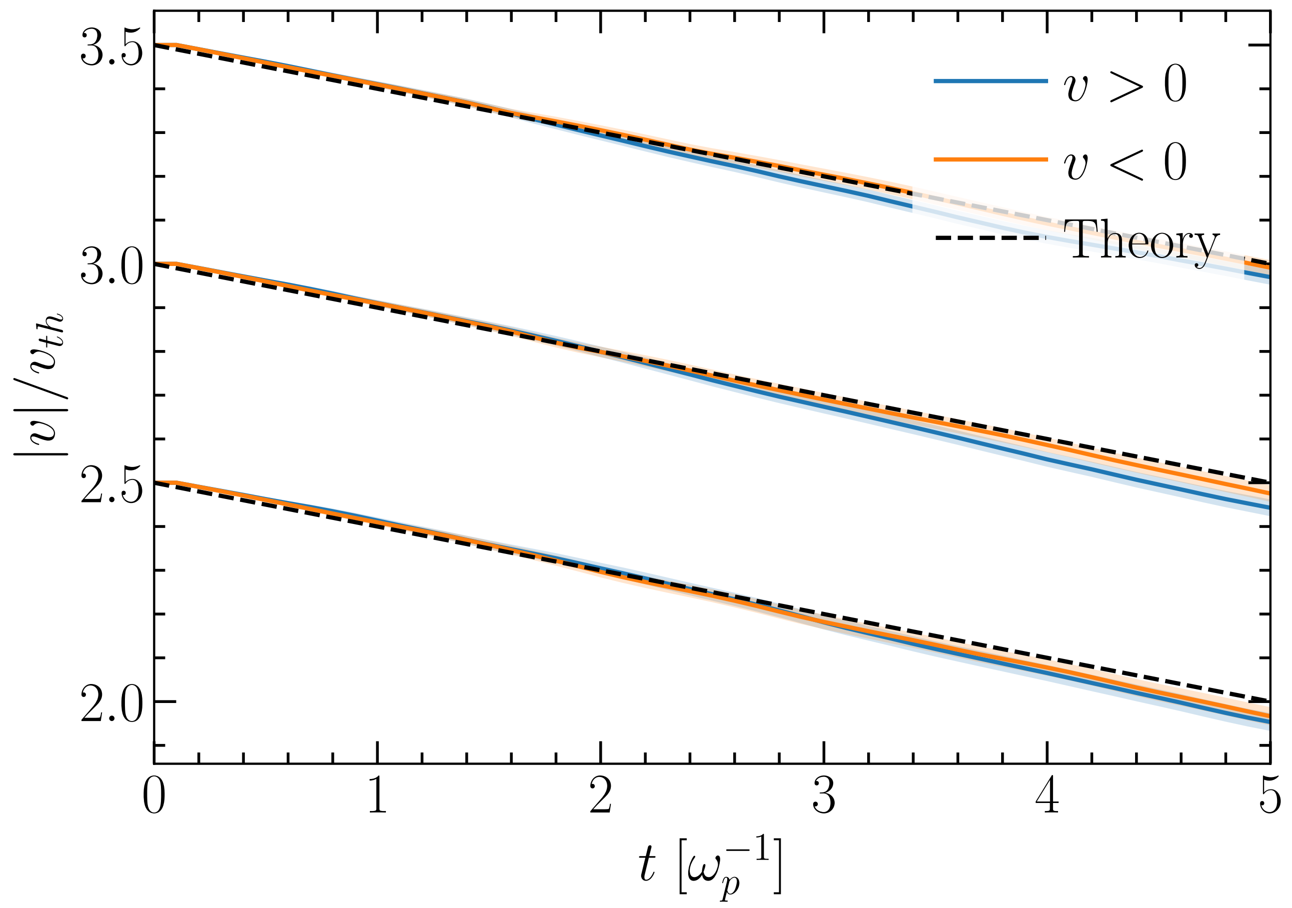
Debye Shielding



Electrostatic Fluctuations



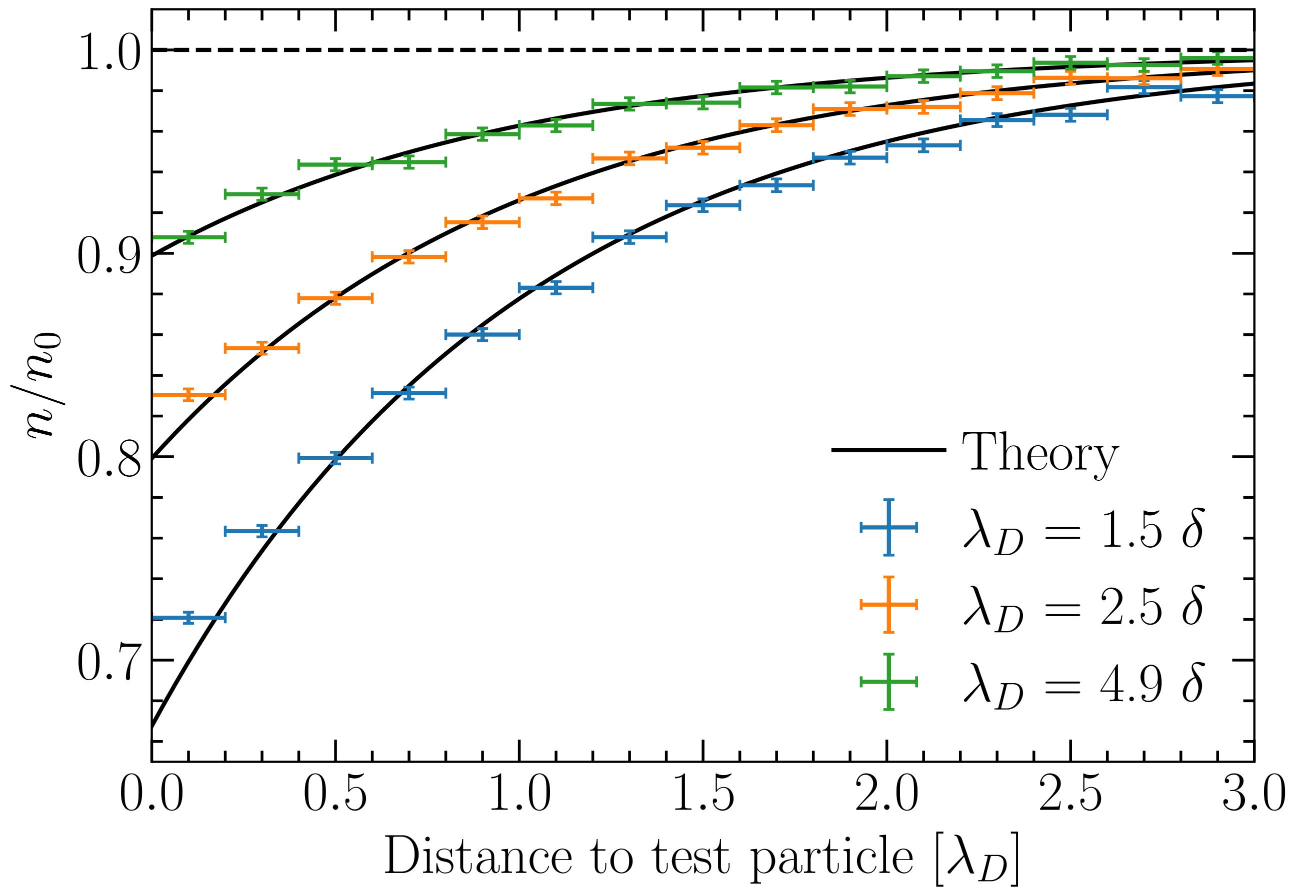
Drag on a Fast Sheet



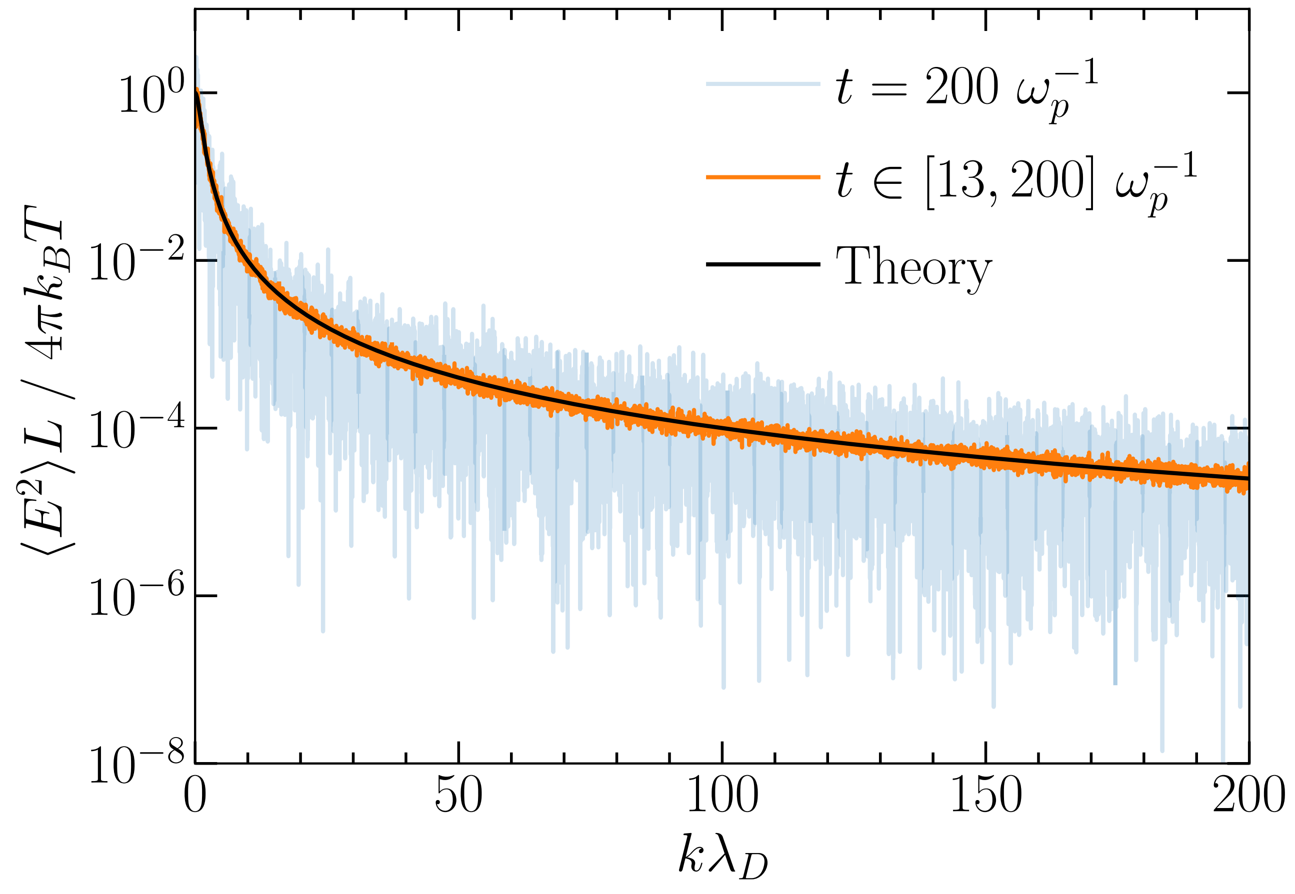
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers a broad range of kinetic plasma processes

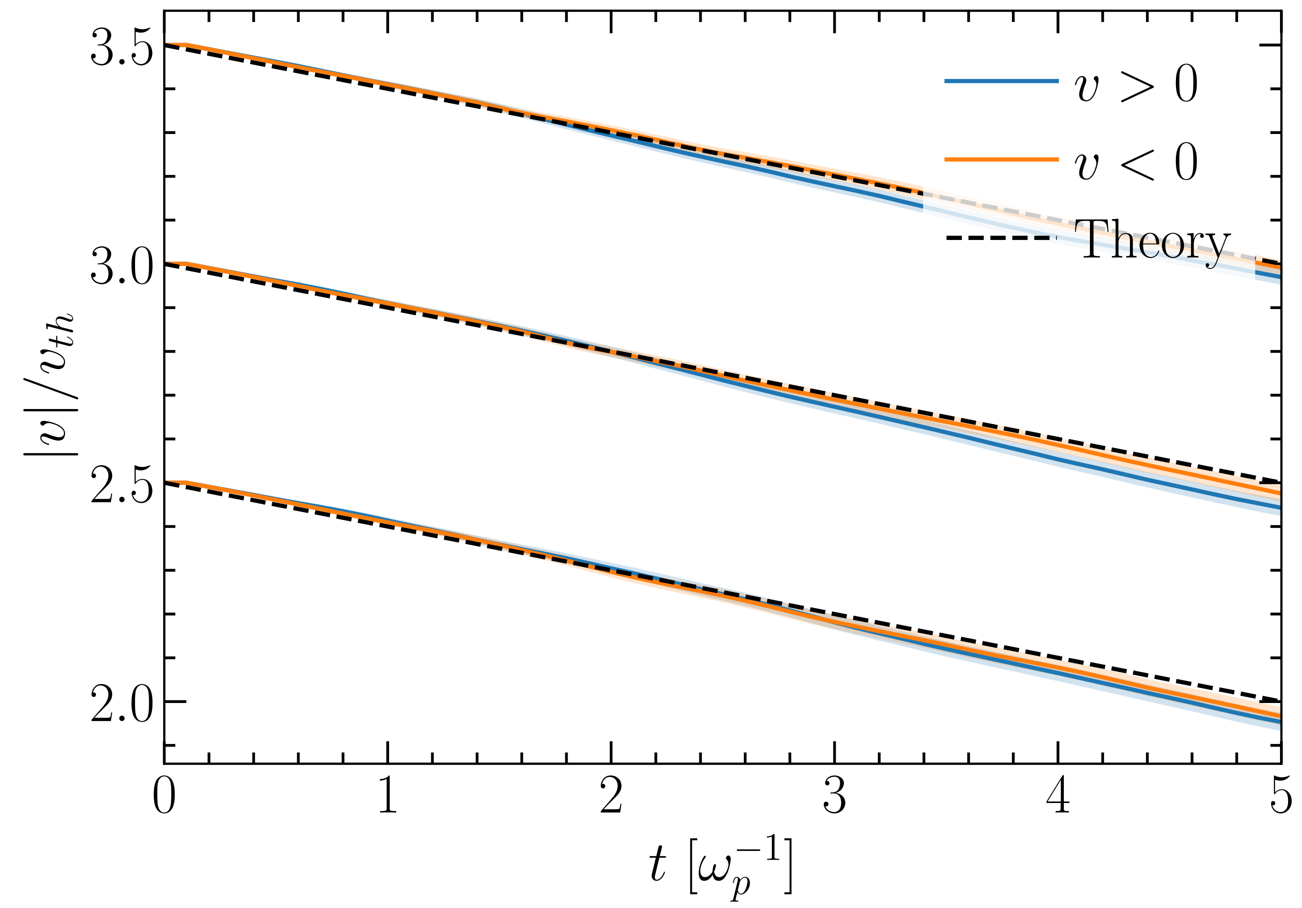
Debye Shielding



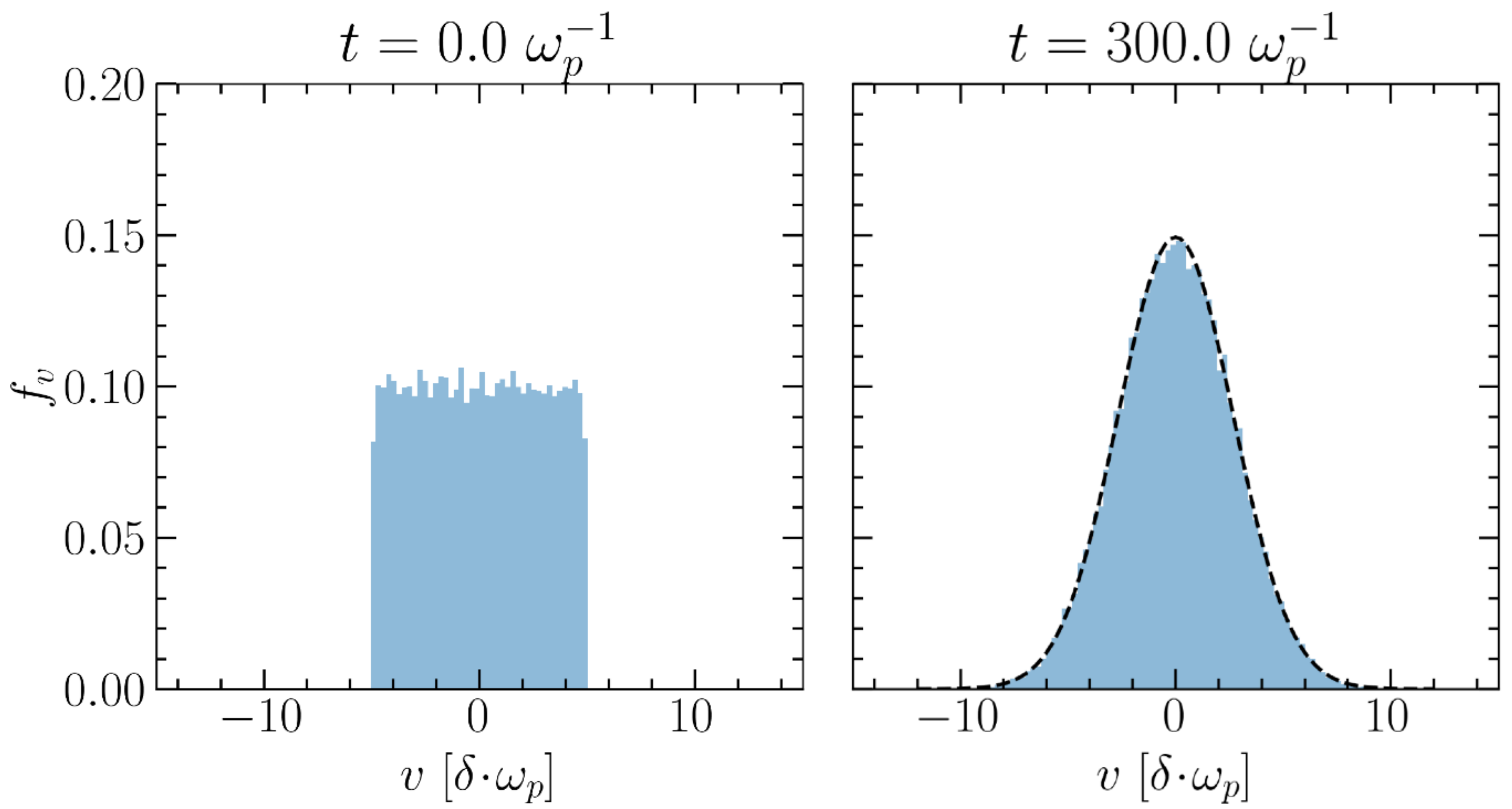
Electrostatic Fluctuations



Drag on a Fast Sheet



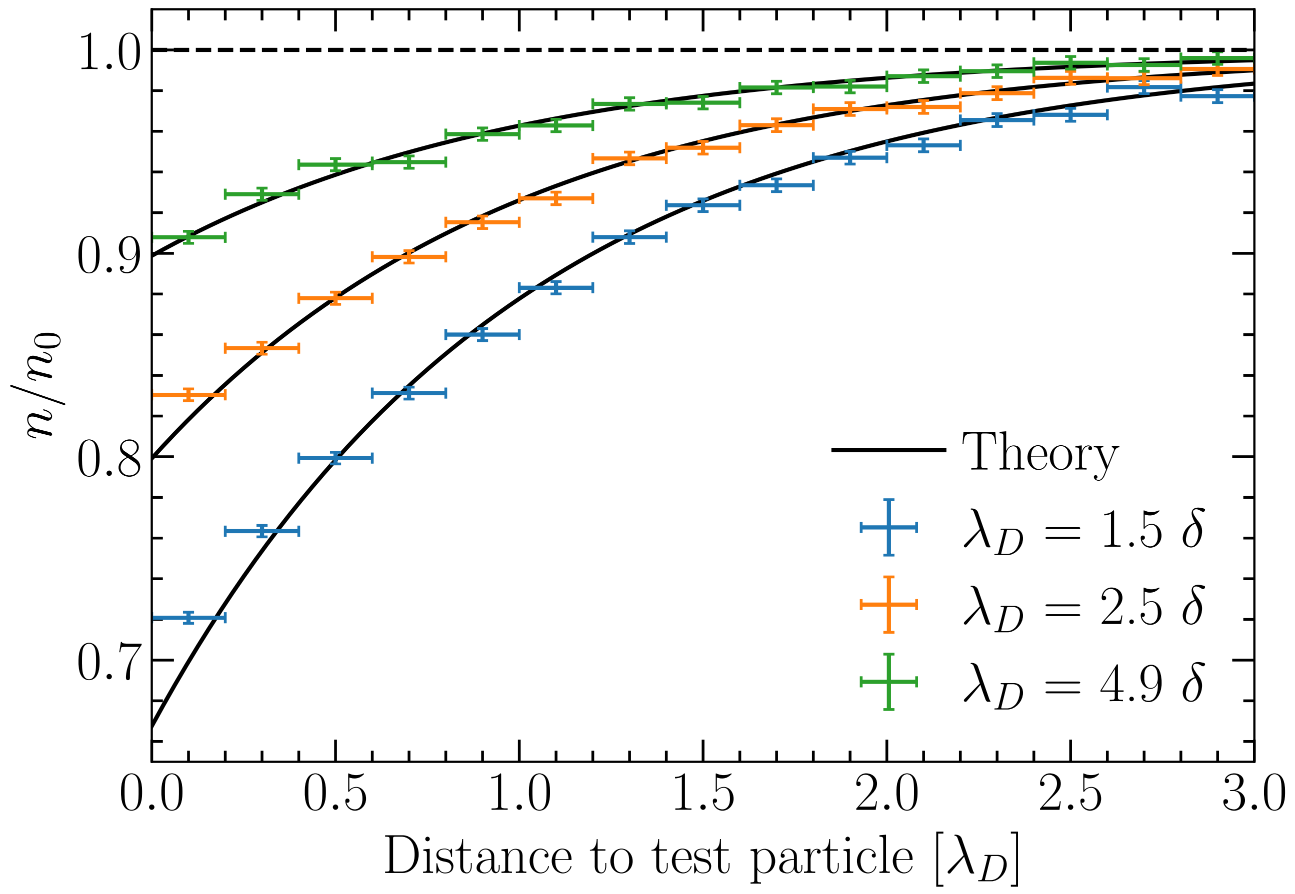
Plasma Thermalization



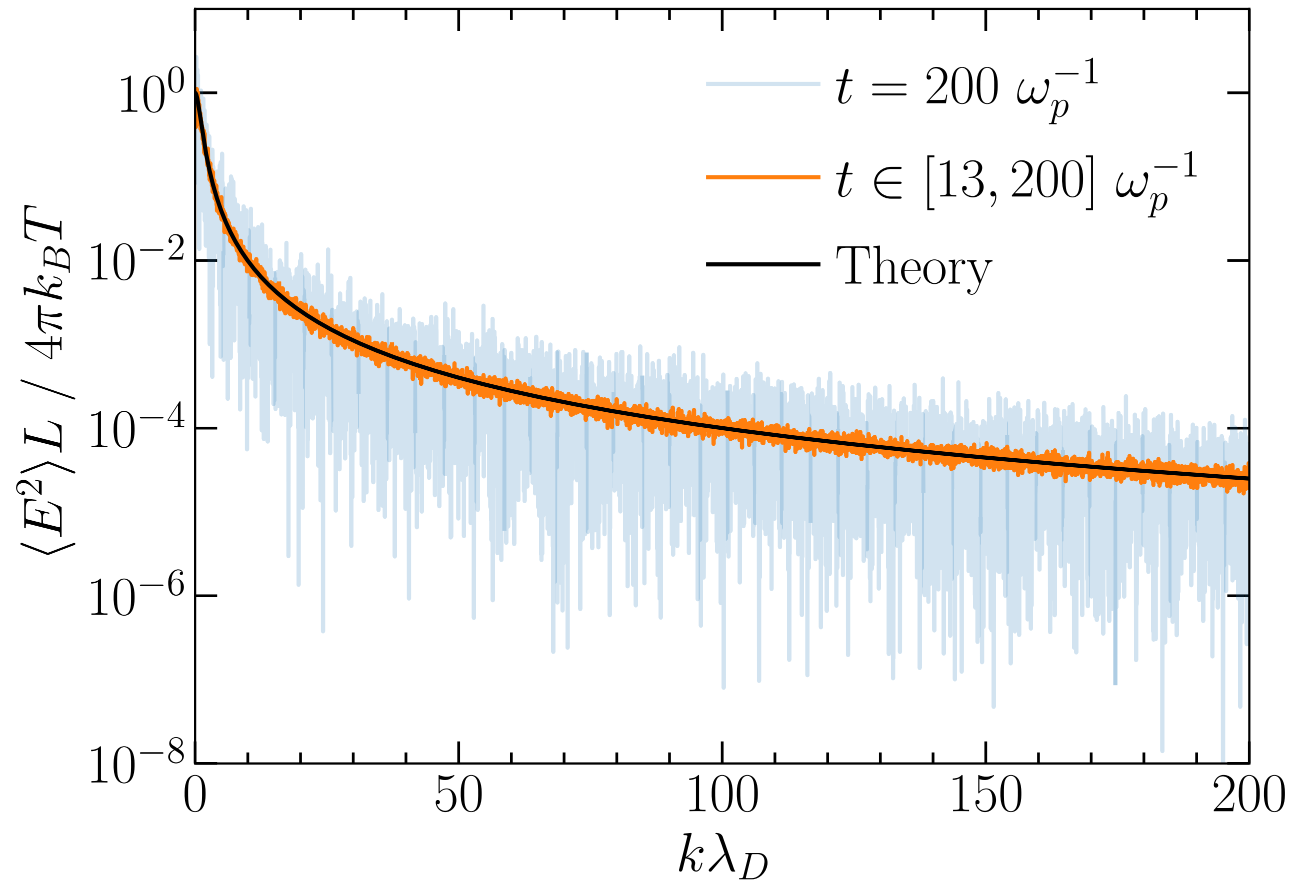
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers a broad range of kinetic plasma processes

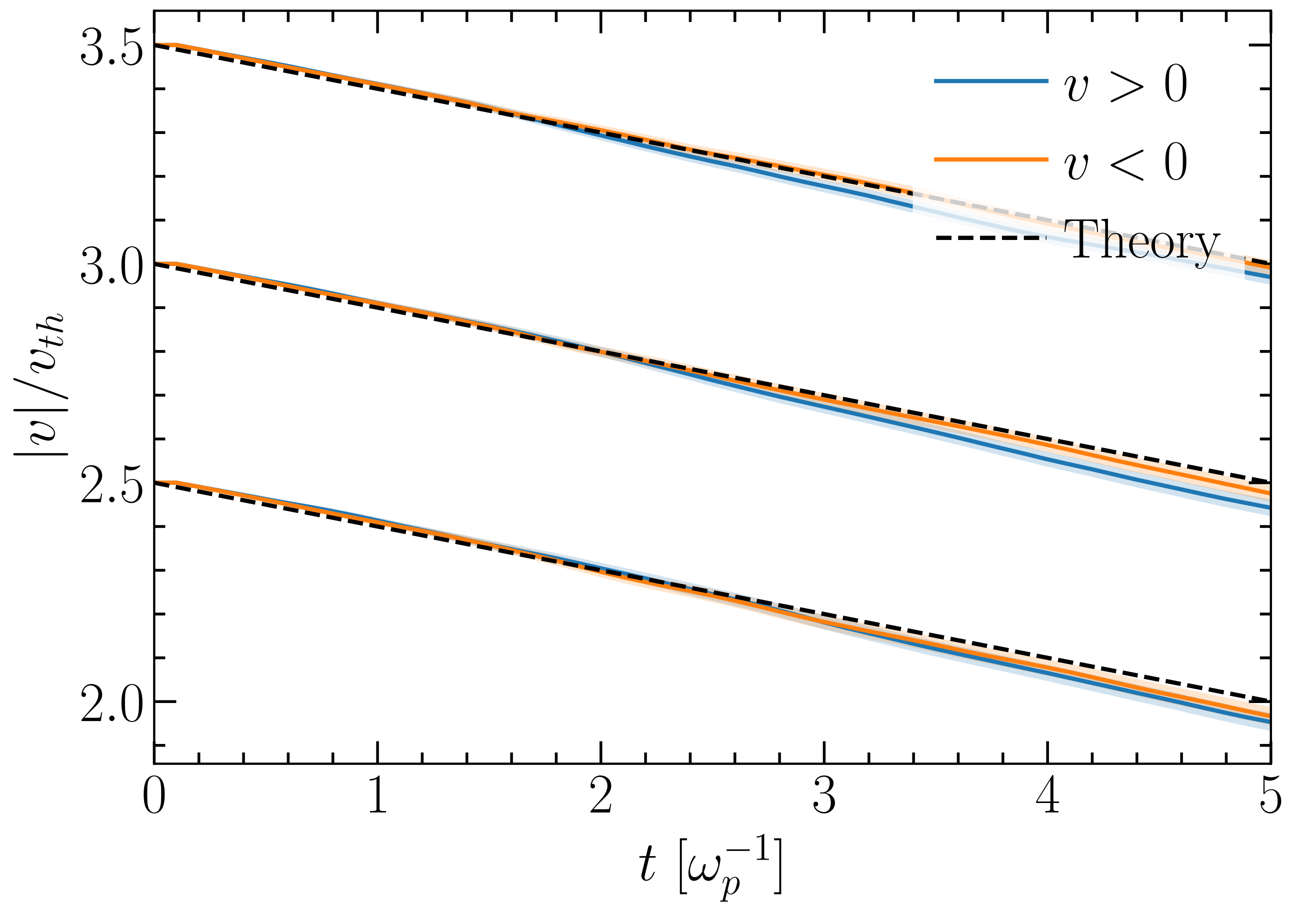
Debye Shielding



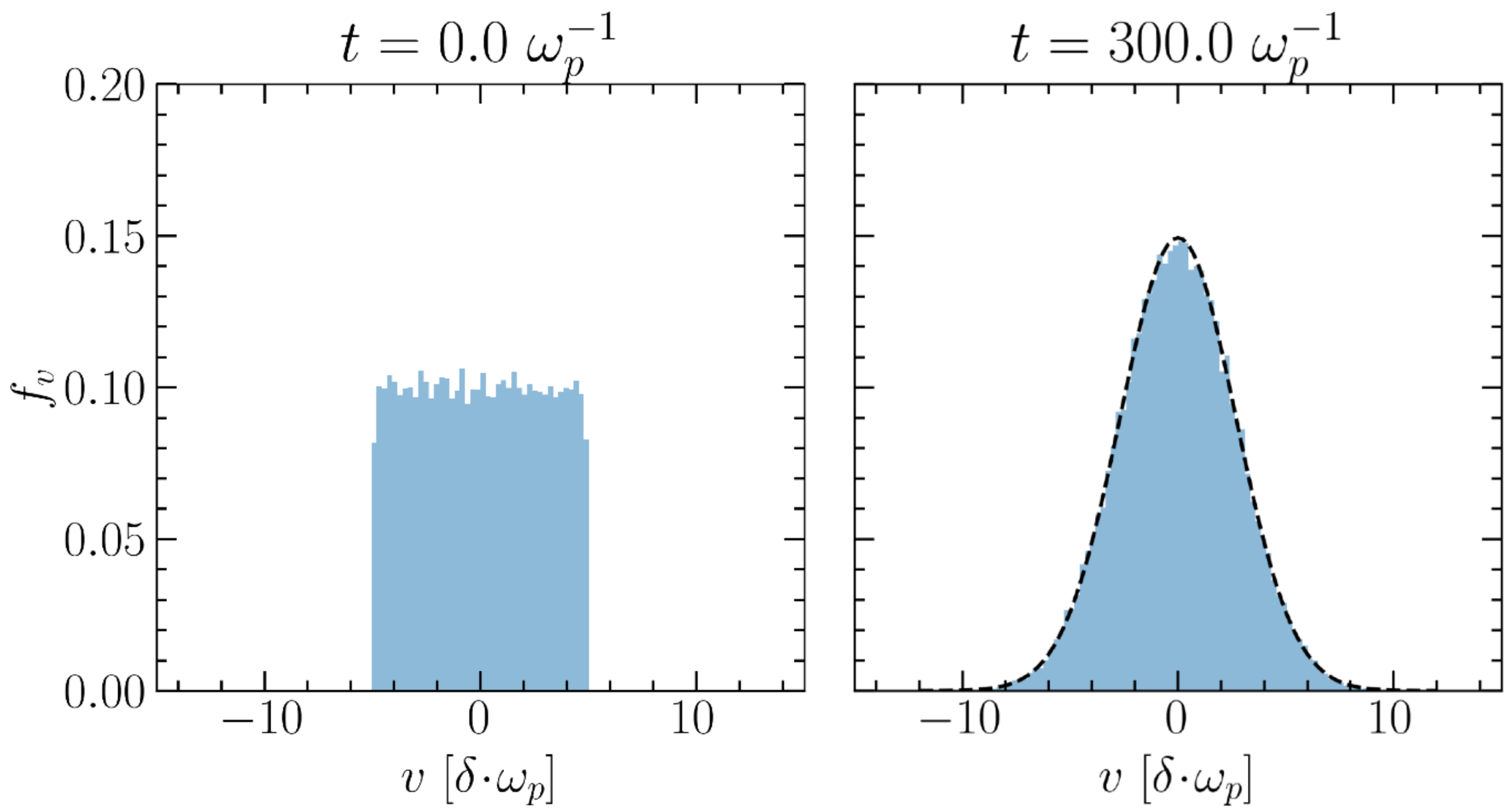
Electrostatic Fluctuations



Drag on a Fast Sheet



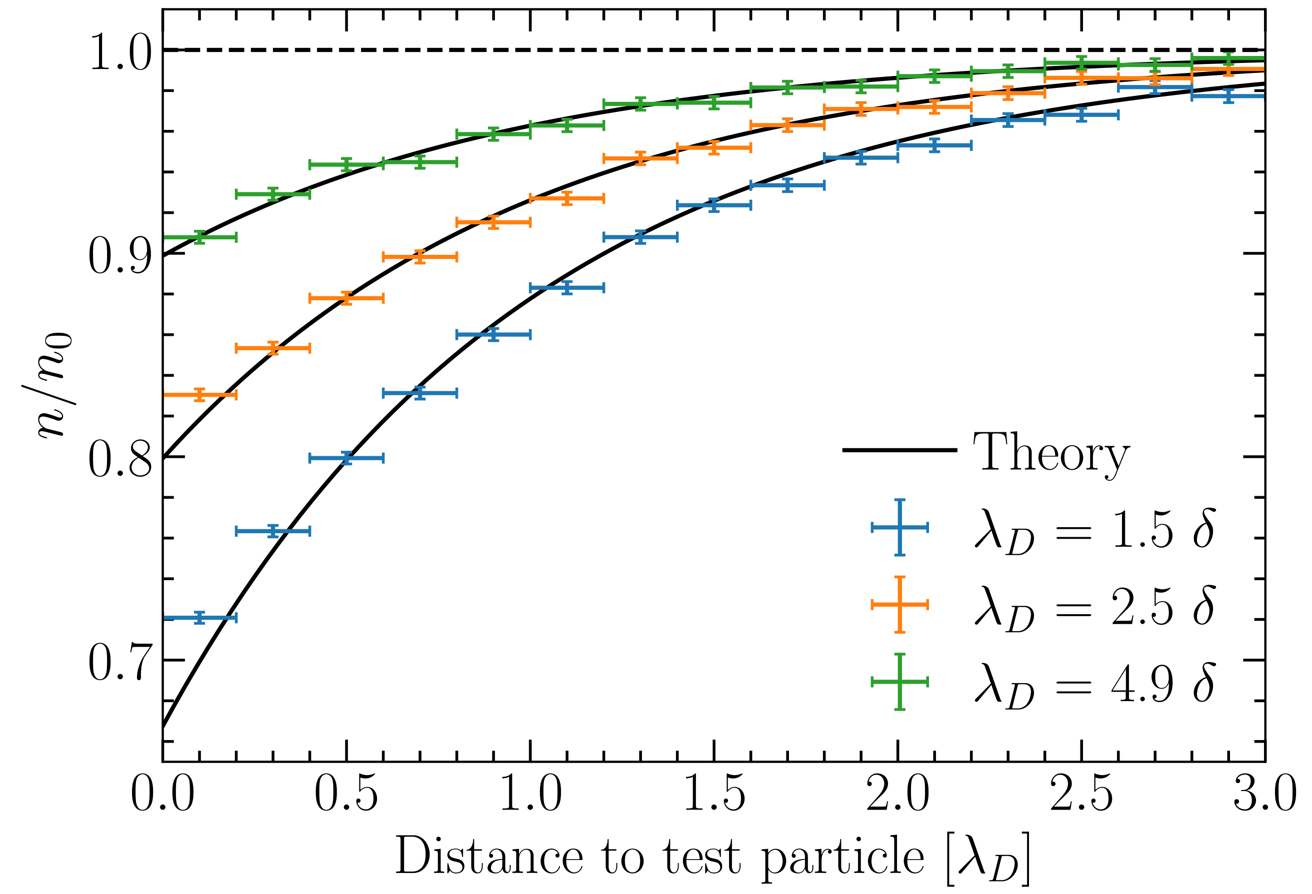
Plasma Thermalization



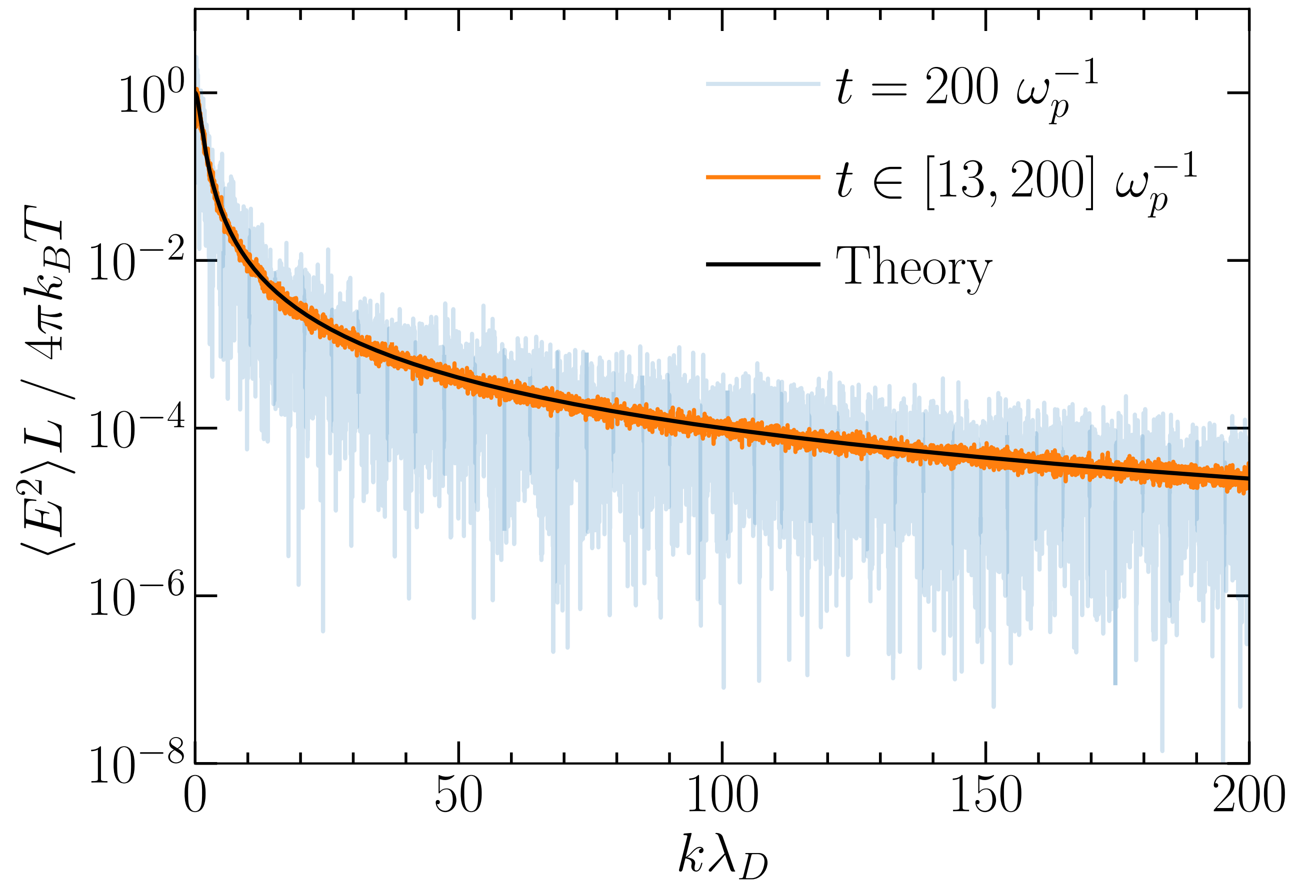
D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers a broad range of kinetic plasma processes

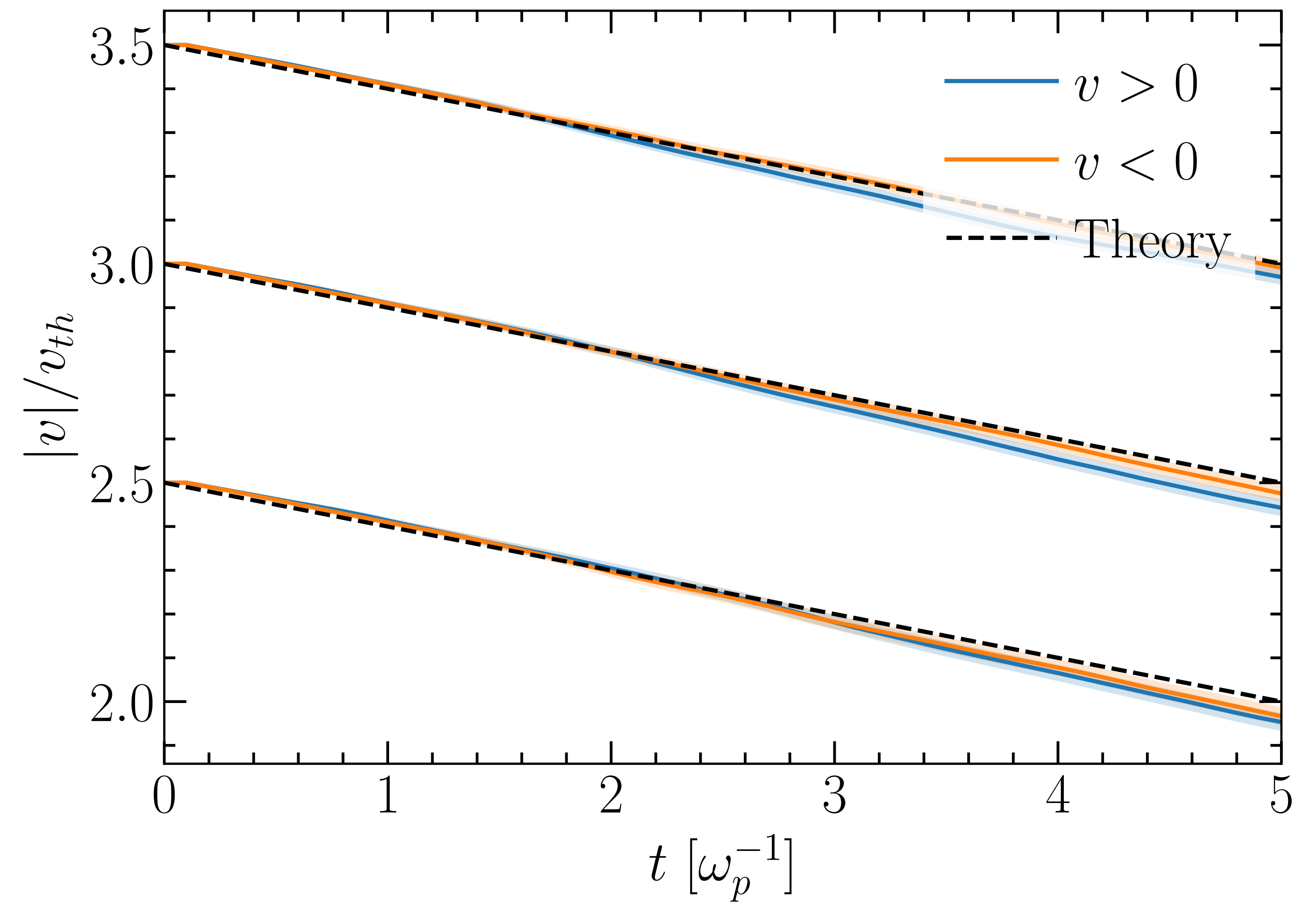
Debye Shielding



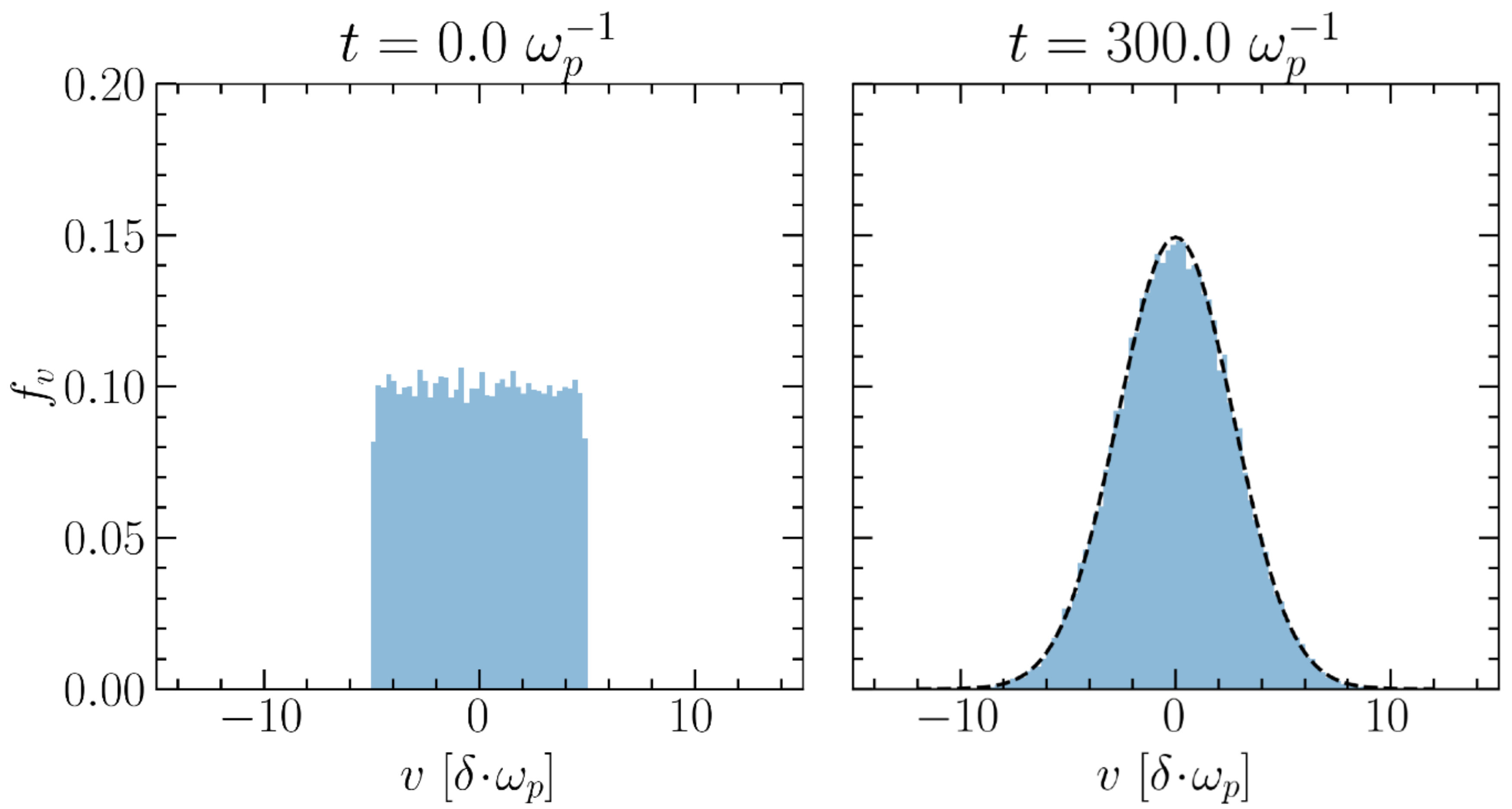
Electrostatic Fluctuations



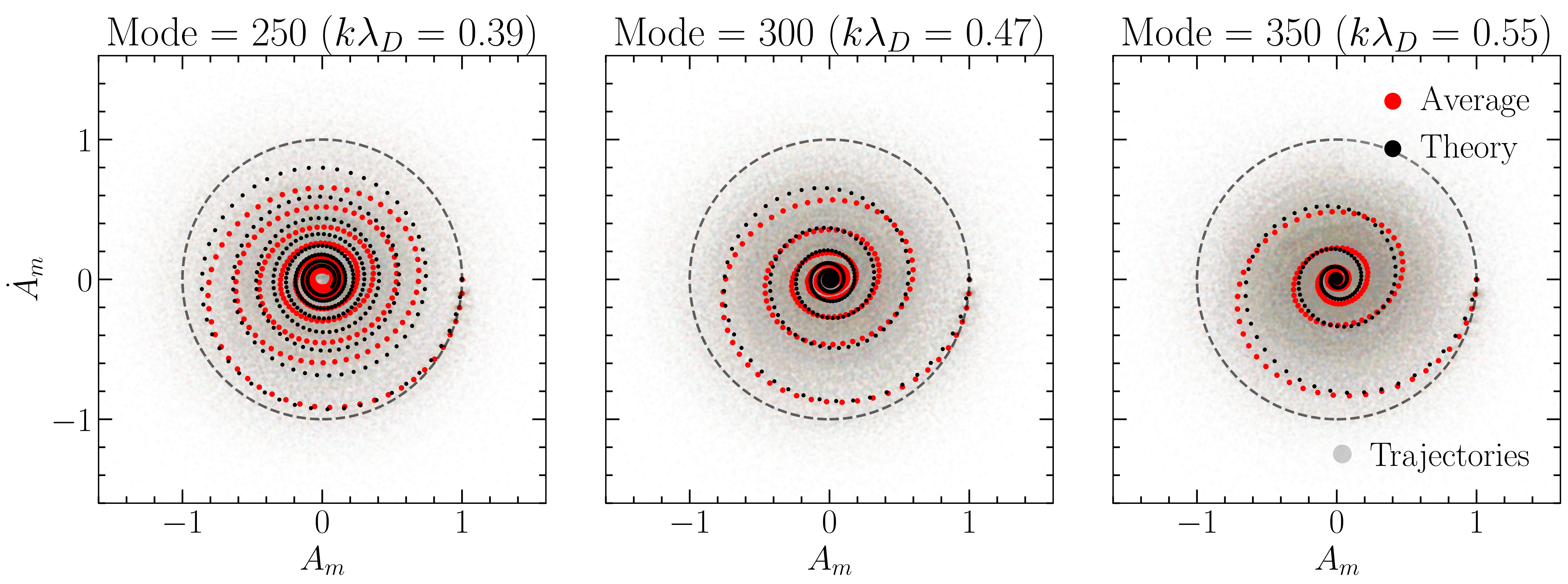
Drag on a Fast Sheet



Plasma Thermalization



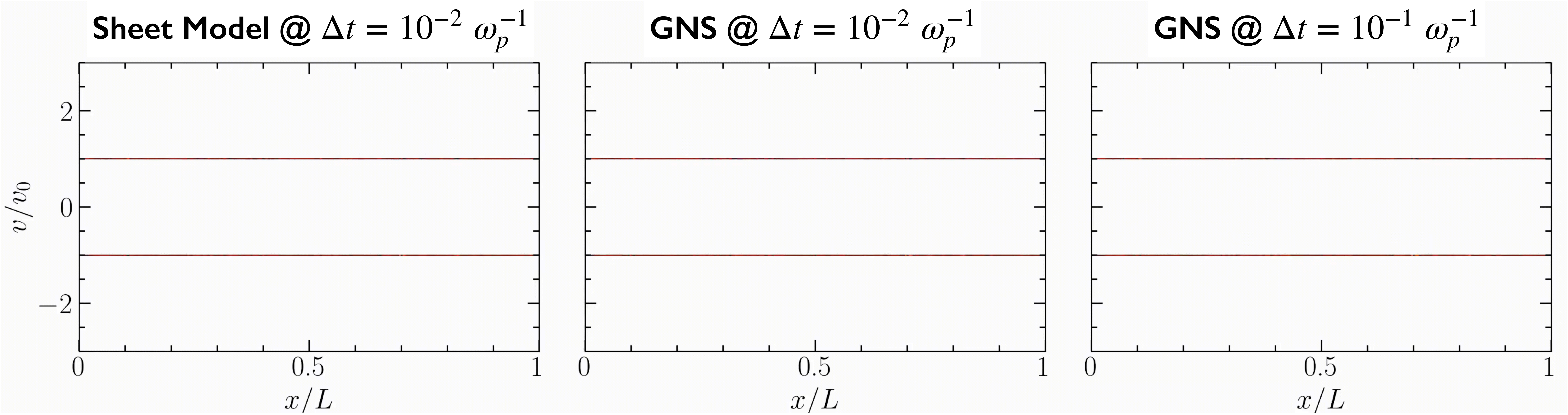
Landau Damping



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

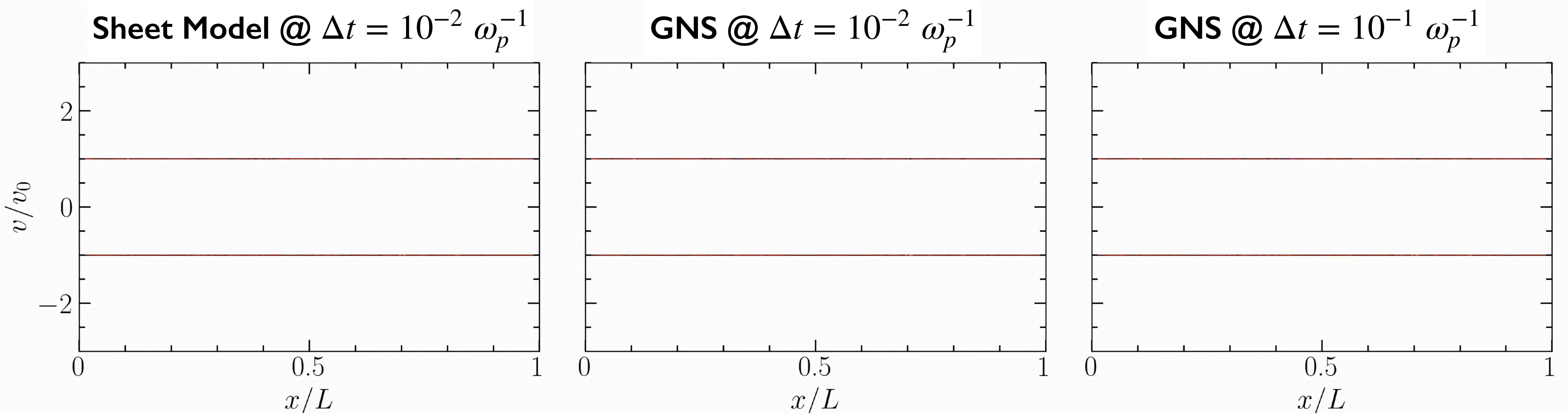
Parameters: $N_{sheets} = 10{,}000 \;\left(\text{vs } N^{train}_{sheets} = 10\right)$ $\quad v_0 \approx 500\ \delta \cdot \omega_p \;\left(\text{vs } v^{train}_{max} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

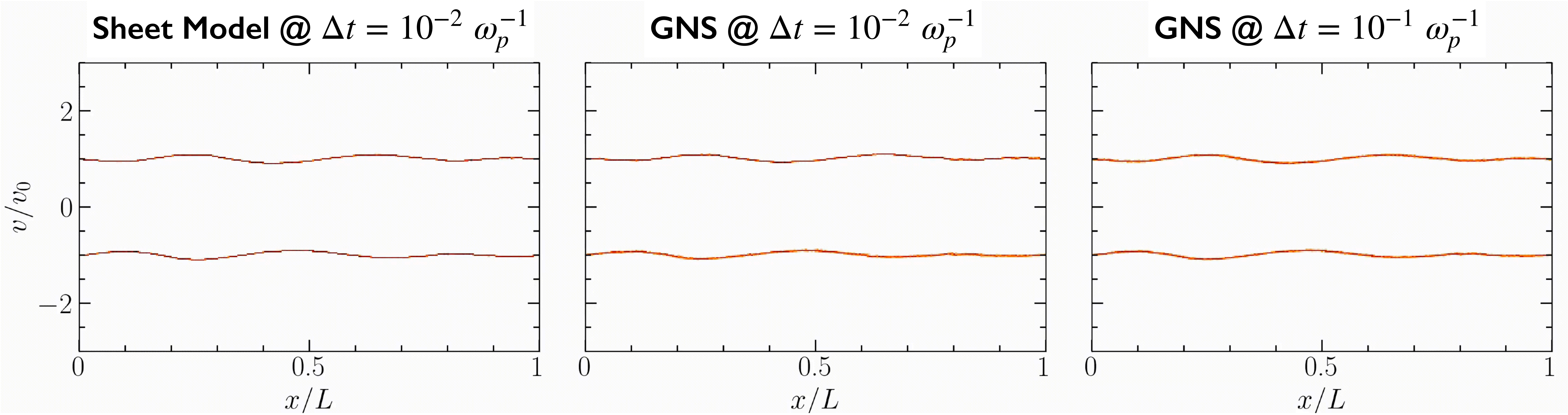
Parameters: $N_{sheets} = 10{,}000 \ \left(\text{vs } N_{sheets}^{train} = 10\right)$ $\quad v_0 \approx 500\ \delta \cdot \omega_p \ \left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

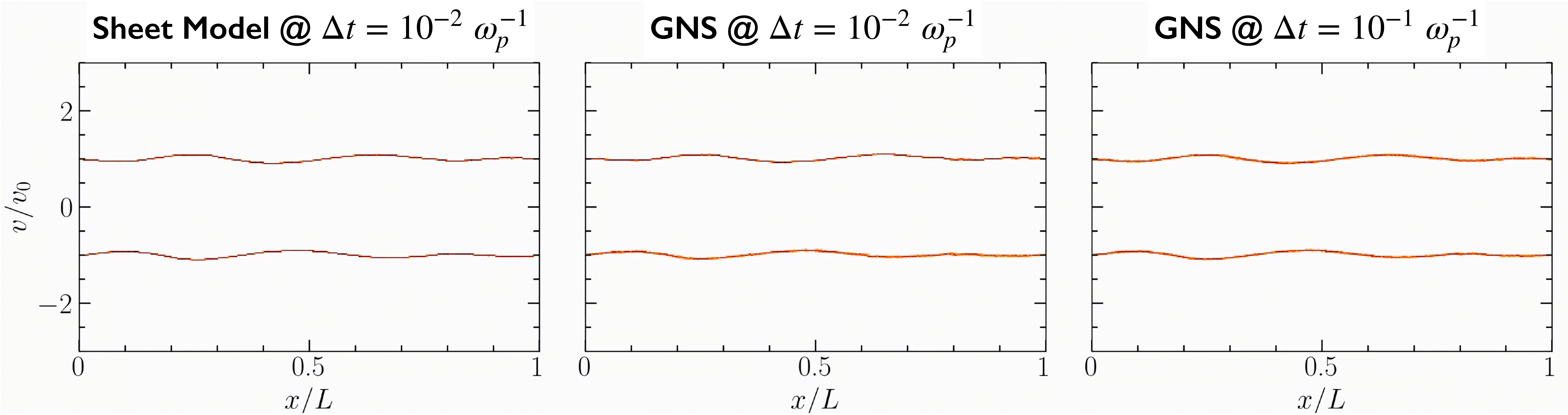
Parameters: $N_{sheets} = 10{,}000 \ \left(\text{vs } N_{sheets}^{train} = 10\right)$ $\quad v_0 \approx 500\ \delta \cdot \omega_p \ \left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

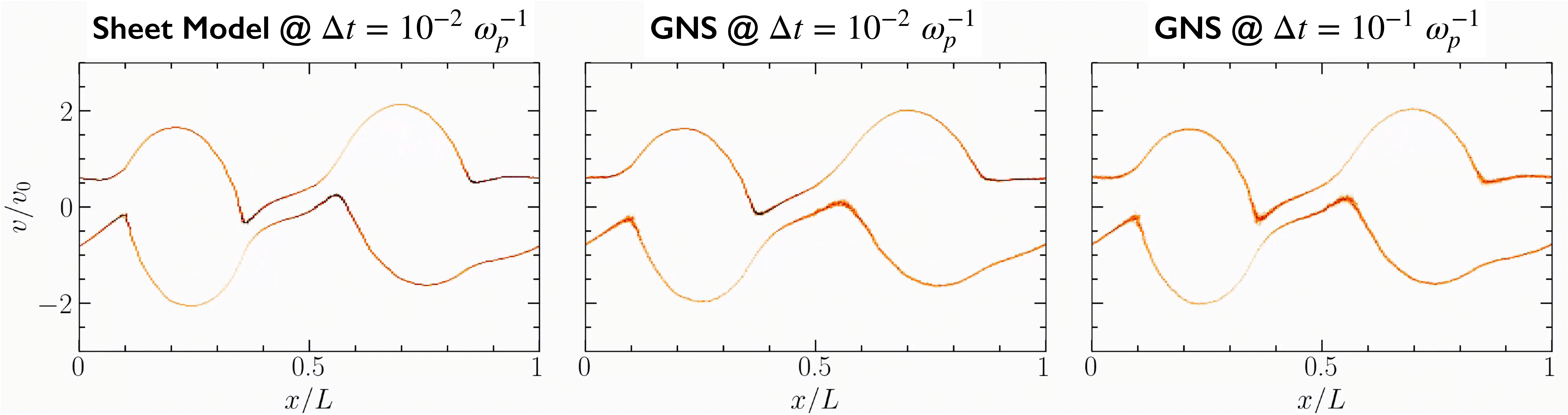
Parameters: $N_{sheets} = 10{,}000$ $\left(\text{vs } N_{sheets}^{train} = 10\right)$ $v_0 \approx 500\ \delta \cdot \omega_p$ $\left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

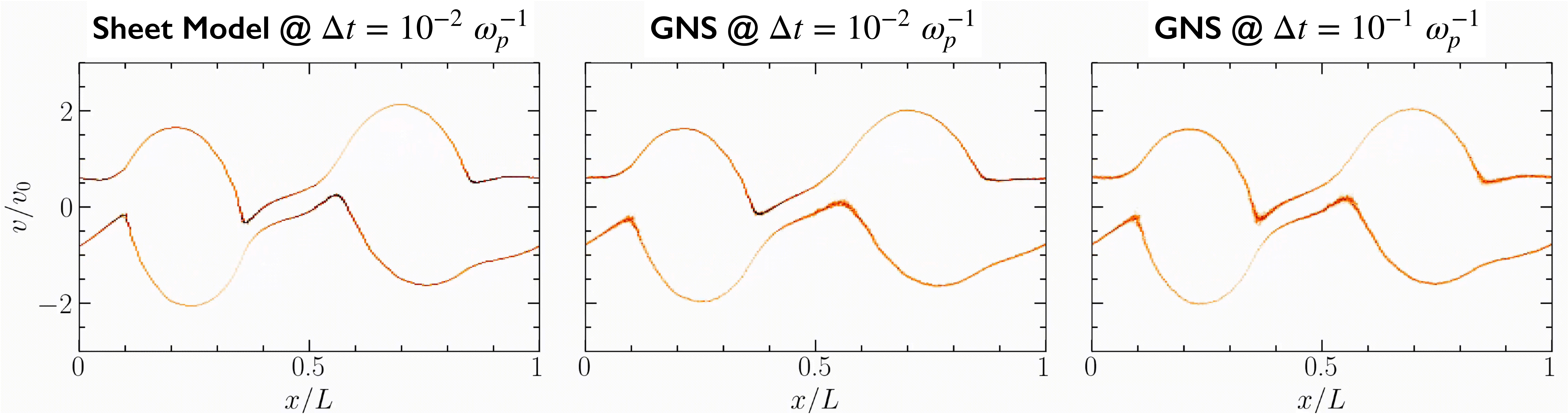
Parameters: $N_{sheets} = 10{,}000$ $\left(\text{vs } N_{sheets}^{train} = 10\right)$ $v_0 \approx 500\ \delta \cdot \omega_p$ $\left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

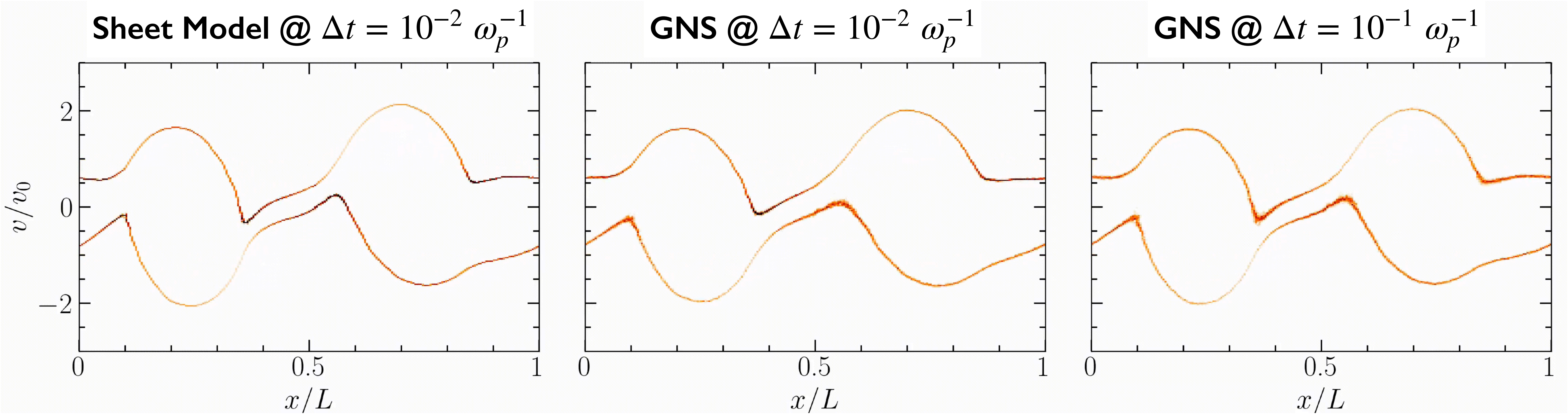
Parameters: $N_{sheets} = 10{,}000 \quad \left(\text{vs } N_{sheets}^{train} = 10\right) \qquad v_0 \approx 500\ \delta \cdot \omega_p \quad \left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

Parameters: $N_{sheets} = 10{,}000 \ \left(\text{vs } N_{sheets}^{train} = 10\right)$ $\quad v_0 \approx 500\ \delta \cdot \omega_p \ \left(\text{vs } v_{max}^{train} = 20\ \delta \cdot \omega_p\right)$



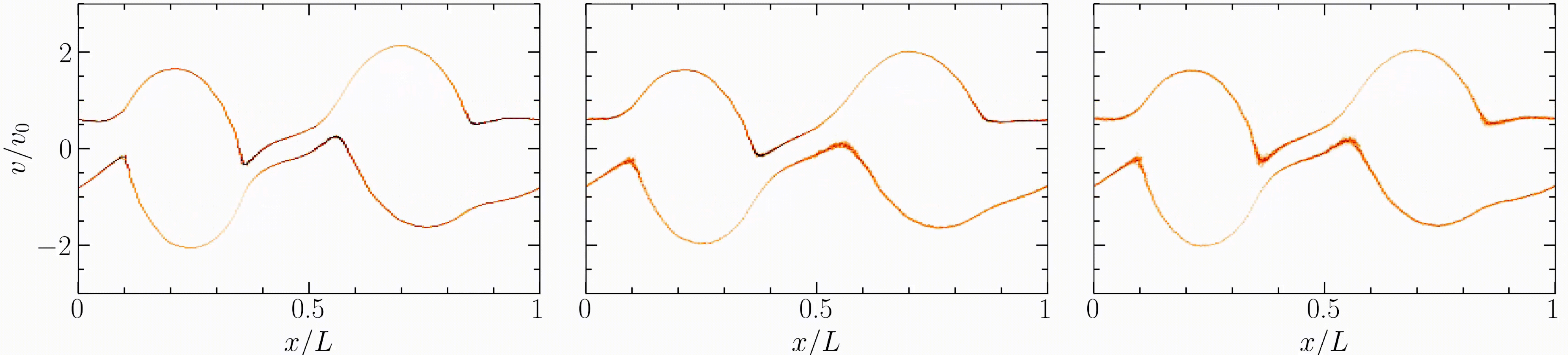
$\Delta\epsilon/\epsilon_0 \approx 10^{-6}$ $\quad \Delta\epsilon/\epsilon_0 \approx 10^{-2}$ $\quad \Delta\epsilon/\epsilon_0 \approx 10^{-2}$

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS recovers the Two Stream Instability

Parameters: $N_{sheets} = 10{,}000 \ \left(\text{vs } N_{sheets}^{train} = 10\right)$ $\quad v_0 \approx 500 \ \delta \cdot \omega_p \ \left(\text{vs } v_{max}^{train} = 20 \ \delta \cdot \omega_p\right)$

Sheet Model @ $\Delta t = 10^{-2} \ \omega_p^{-1}$	**GNS @ $\Delta t = 10^{-2} \ \omega_p^{-1}$**	**GNS @ $\Delta t = 10^{-1} \ \omega_p^{-1}$**

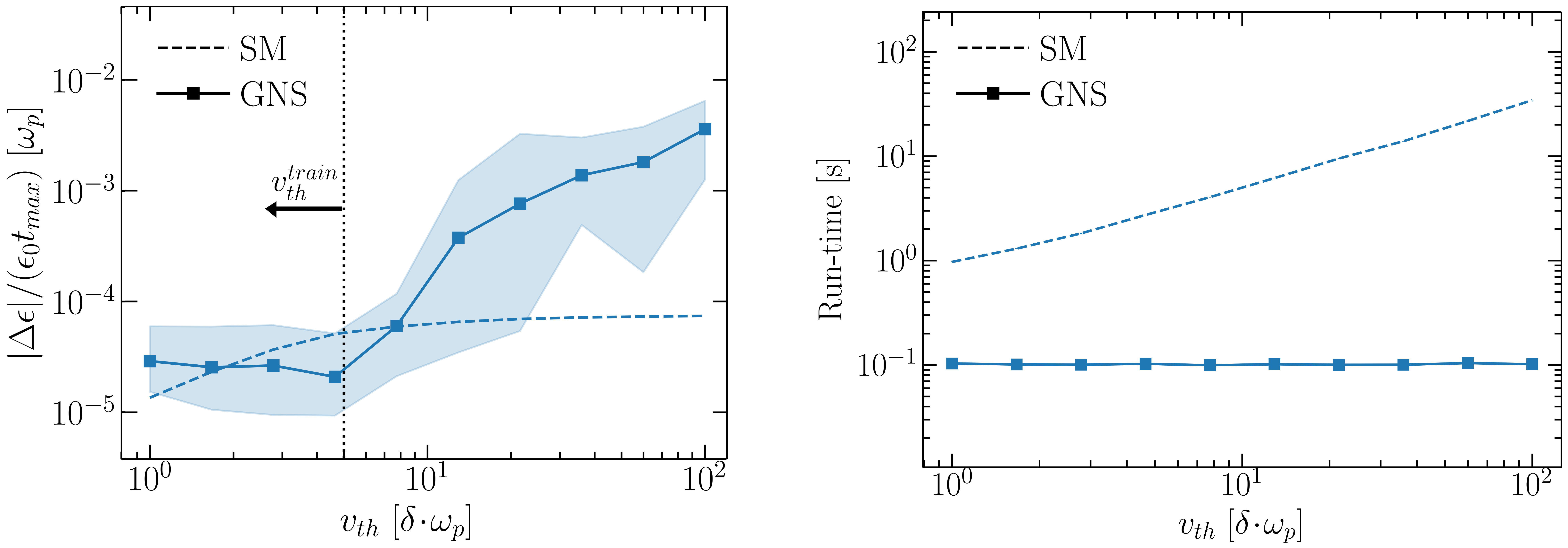


$\Delta\epsilon/\epsilon_0 \approx 10^{-6}$	$\Delta\epsilon/\epsilon_0 \approx 10^{-2}$	$\Delta\epsilon/\epsilon_0 \approx 10^{-2}$
Run-Time $\approx$ 1h	Run-Time $\approx$ 1min	Run-Time $\approx$ 10s

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

GNS conserves energy similarly to Sheet Model while being significantly faster*

Parameters: $N_{sheets} = 1000$, velocities sampled from thermal distribution, $\Delta t_{GNS} = 10^{-1}\ \omega_p^{-1}$



***Note: GNS is implemented in JAX** (GPU), Sheet Model is implemented in NumPy (CPU)

D. Carvalho et al., Mach. Learn.: Sci. Technol. 5 025048 (2024)

MC models in PIC simulations

New simulator models - GNN collisional plasma model

Learning advection and diffusion coefficients

The (ground) truth? - collisions in PIC codes

What is the ground truth in PIC simulations?

Klimontovich + Maxwell's equations

$$\frac{\partial N}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} N - \frac{q}{m} \left(\mathbf{E}^m + \mathbf{v} \times \mathbf{B}^m \right) \cdot \nabla_{\mathbf{v}} N = 0$$

This is the particle-in-cell algorithm (with finite-size particles):

statistical mechanics is well-known (e.g. H. Okuda and C. Birdsall, (1970), R. Hockney (1971), M. Touati et al. (2022), S. Jubin et al., (2024))

Born-Infeld electrodynamics

Numerical collision operator has been derived in previous works: Can this be learned from the simulation data in the weakly collisional regime?

D. Carvalho et al., in preparation for JPP (2025)

Can we describe phase-space dynamics using a Fokker-Planck operator?

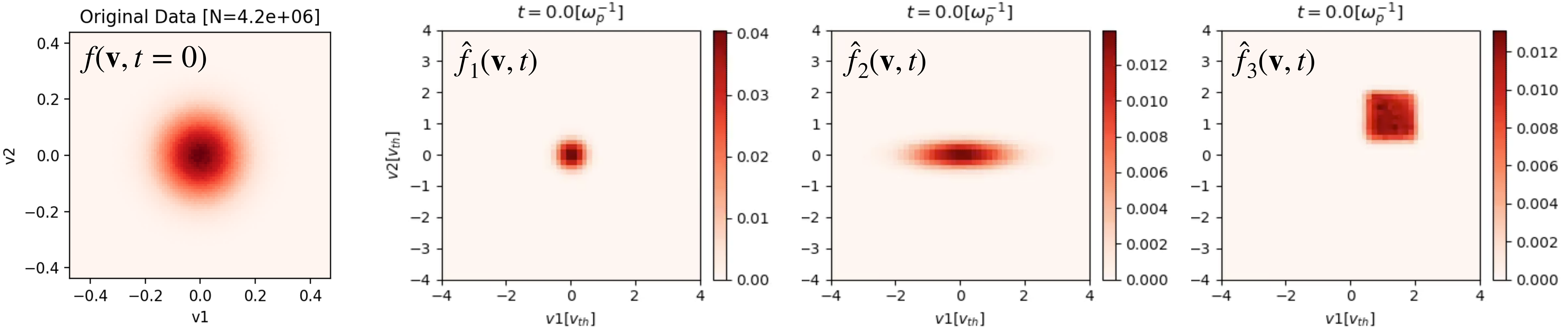
PIC (Klimontovich)

$$\frac{\partial N}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} N - \frac{q}{m} \left(\mathbf{E}^m + \mathbf{v} \times \mathbf{B}^m \right) \cdot \nabla_{\mathbf{v}} N = 0$$

? →

What if we want (Fokker-Planck)?

$$\frac{\partial \hat{f}(\mathbf{v}, t)}{\partial t} = -\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} \hat{f} \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} \hat{f} \right)$$



Thermal Plasma

D. Carvalho et al., in preparation for JPP (2025)

Can we describe phase-space dynamics using a Fokker-Planck operator?

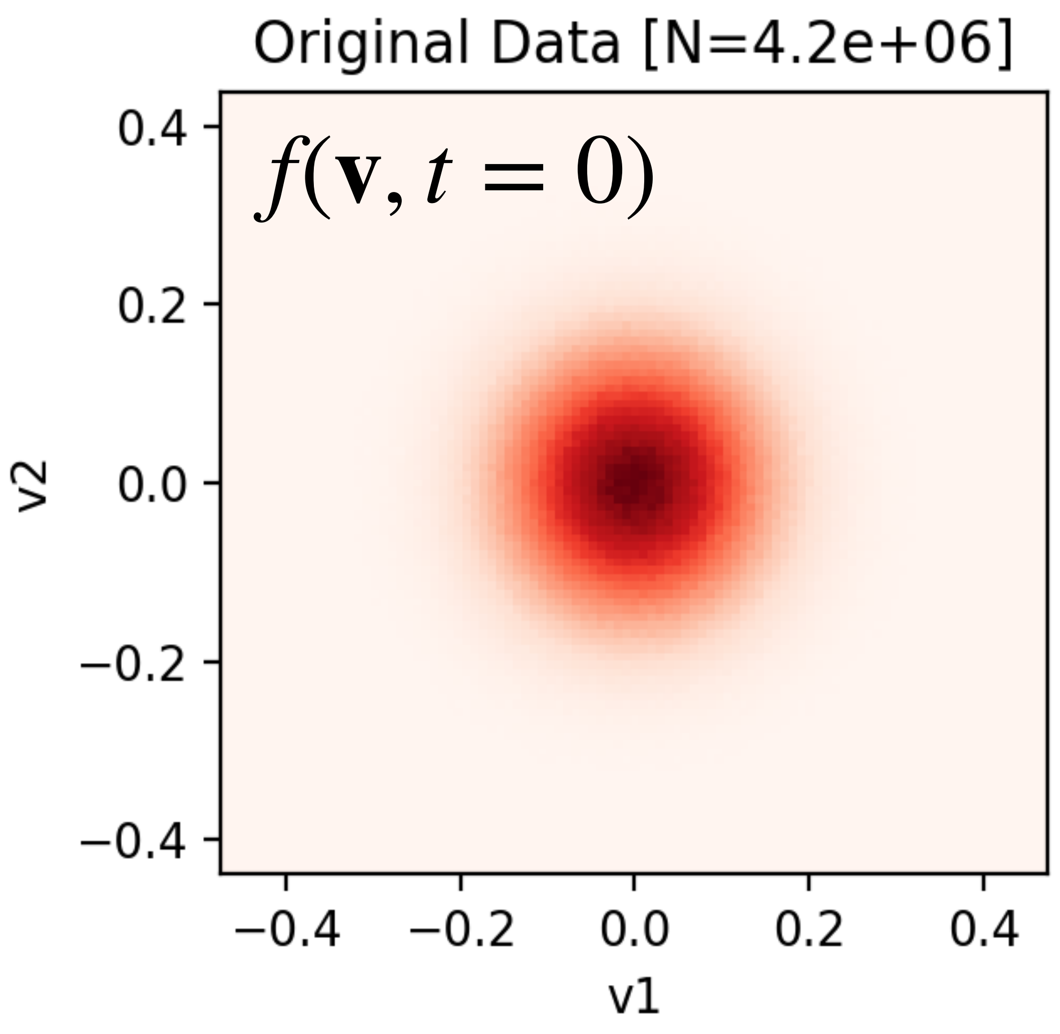
PIC (Klimontovich)

$$\frac{\partial N}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} N - \frac{q}{m} \left(\mathbf{E}^m + \mathbf{v} \times \mathbf{B}^m\right) \cdot \nabla_{\mathbf{v}} N = 0$$

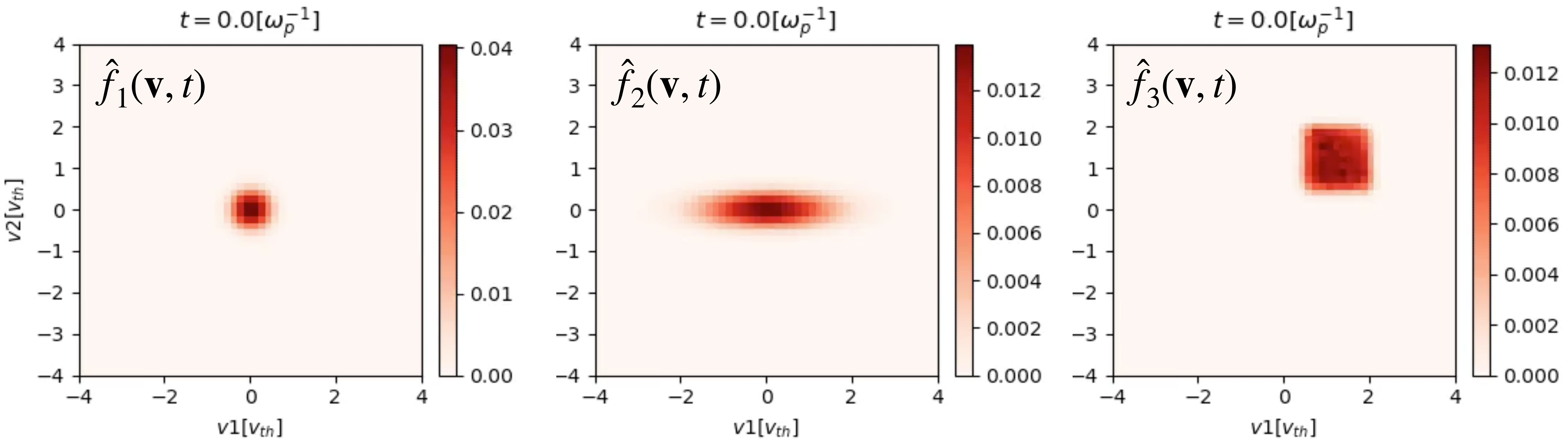
? →

What if we want (Fokker-Planck)?

$$\frac{\partial \hat{f}(\mathbf{v}, t)}{\partial t} = - \nabla_{\mathbf{v}} \cdot \left(\mathbf{A}\hat{f}\right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}}\hat{f}\right)$$



Thermal Plasma



D. Carvalho et al., in preparation for JPP (2025)

Can we describe phase-space dynamics using a Fokker-Planck operator?

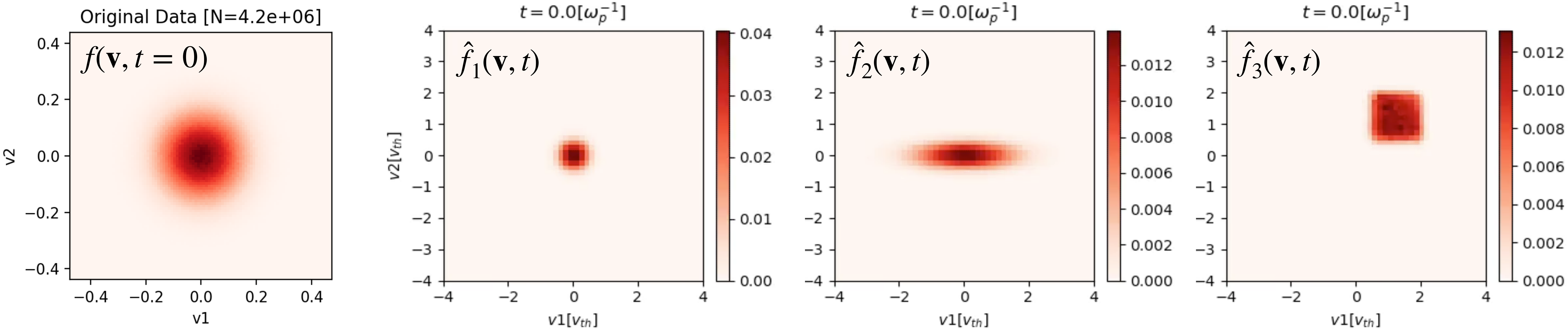
PIC (Klimontovich)

$$\frac{\partial N}{\partial t} + \mathbf{v}\cdot\nabla_{\mathbf{x}} N - \frac{q}{m}\left(\mathbf{E}^m + \mathbf{v}\times\mathbf{B}^m\right)\cdot\nabla_{\mathbf{v}} N = 0$$

?

What if we want (Fokker-Planck)?

$$\frac{\partial \hat{f}(\mathbf{v},t)}{\partial t} = -\nabla_{\mathbf{v}}\cdot\left(\mathbf{A}\hat{f}\right) + \frac{1}{2}\nabla_{\mathbf{v}}\nabla_{\mathbf{v}}\cdot\left(\overleftrightarrow{\mathbf{D}}\hat{f}\right)$$



Thermal Plasma

How do we estimate $\mathbf{A}$ (advection) and $\overleftrightarrow{\mathbf{D}}$ (diffusion)?

D. Carvalho et al., in preparation for JPP (2025)

How do we estimate $\mathbf{A}$ (advection) and $\overleftrightarrow{\mathbf{D}}$ (diffusion)?

Option 1: From raw particle data

The "correct" approach if possible

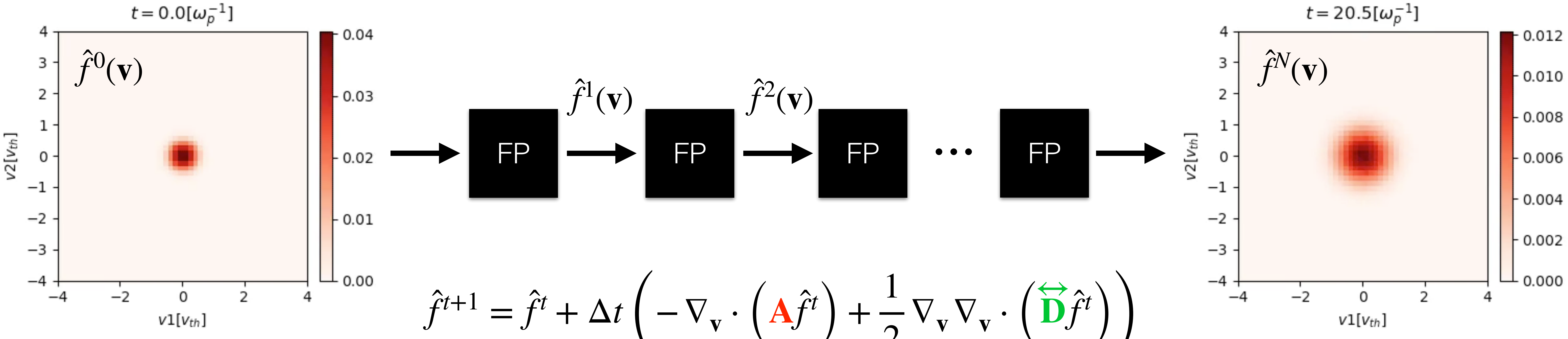
Not feasible for larger systems (memory-wise) unless it is done at run-time

Option 2: From the phase-space evolution of sub-populations

Can be done in post-processing with a differentiable solver

Ill-posed problem: non-unique solution for coefficients

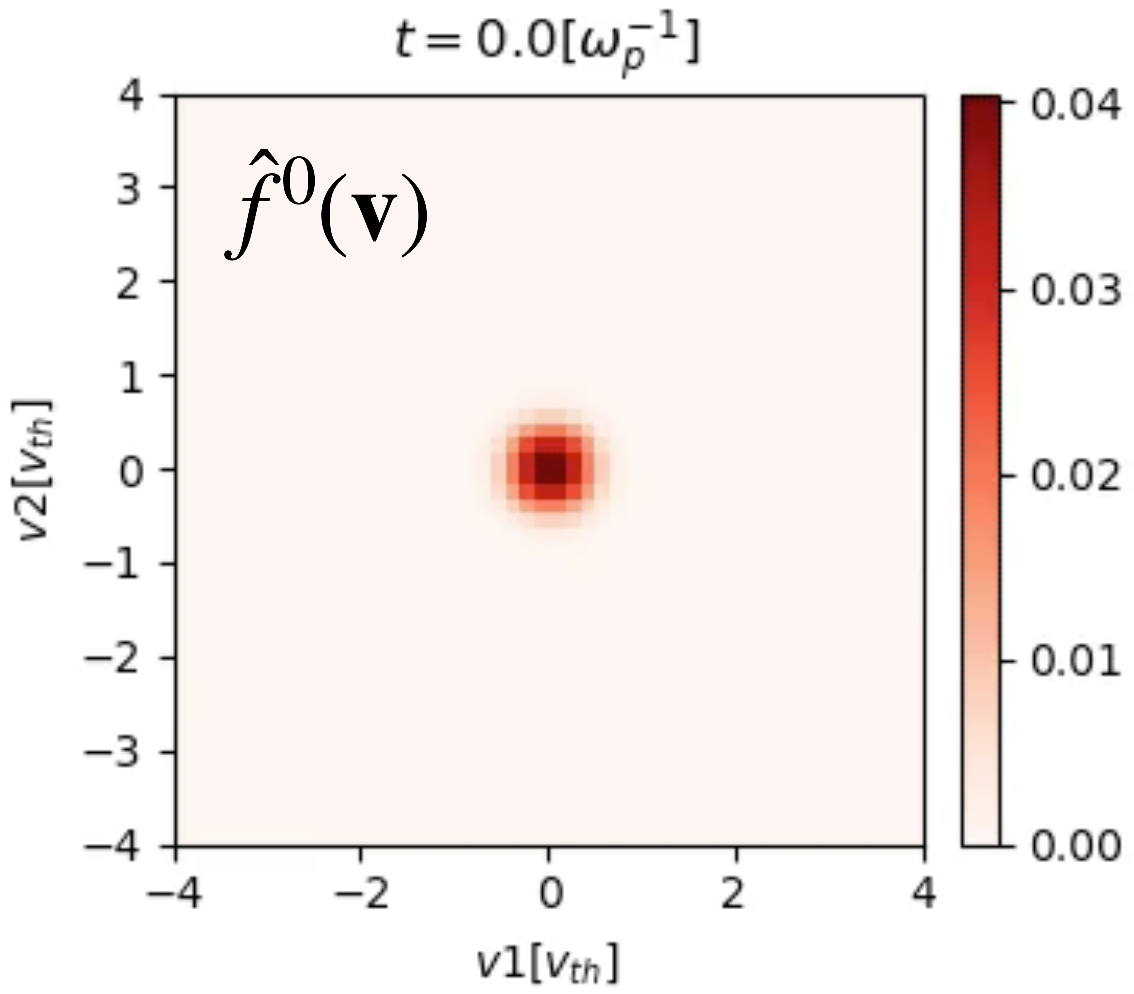
Learning advection / diffusion from evolution of sub-populations

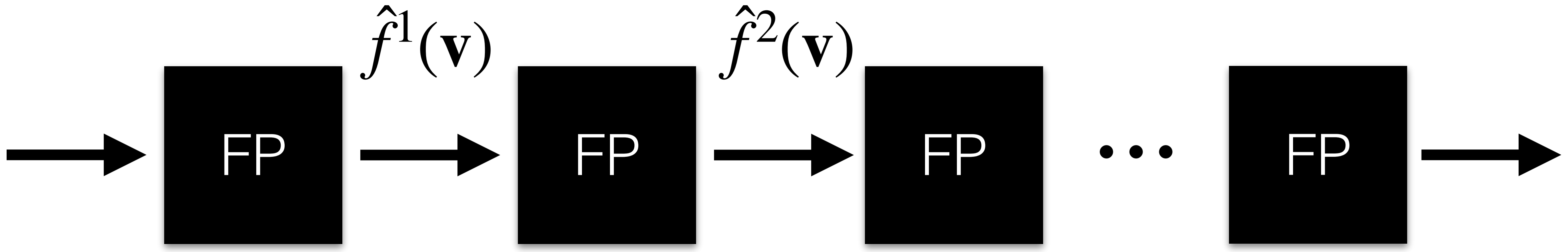


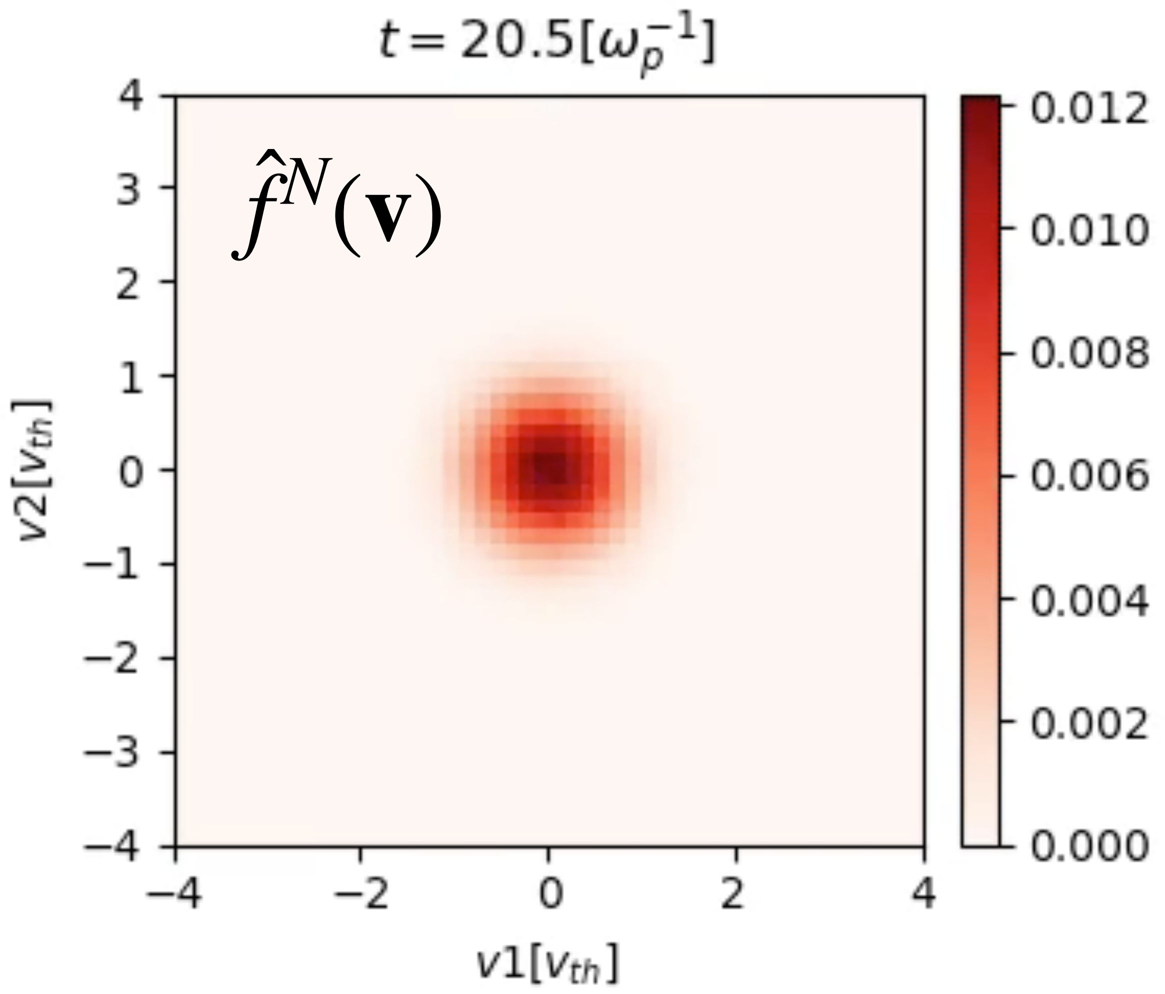
$$\hat{f}^{t+1} = \hat{f}^t + \Delta t \left(-\nabla_\mathbf{v} \cdot \left(\mathbf{A} \hat{f}^t \right) + \frac{1}{2} \nabla_\mathbf{v} \nabla_\mathbf{v} \cdot \left(\overleftrightarrow{\mathbf{D}} \hat{f}^t \right) \right)$$

D. Carvalho et al., in preparation for JPP (2025)

Learning advection / diffusion from evolution of sub-populations



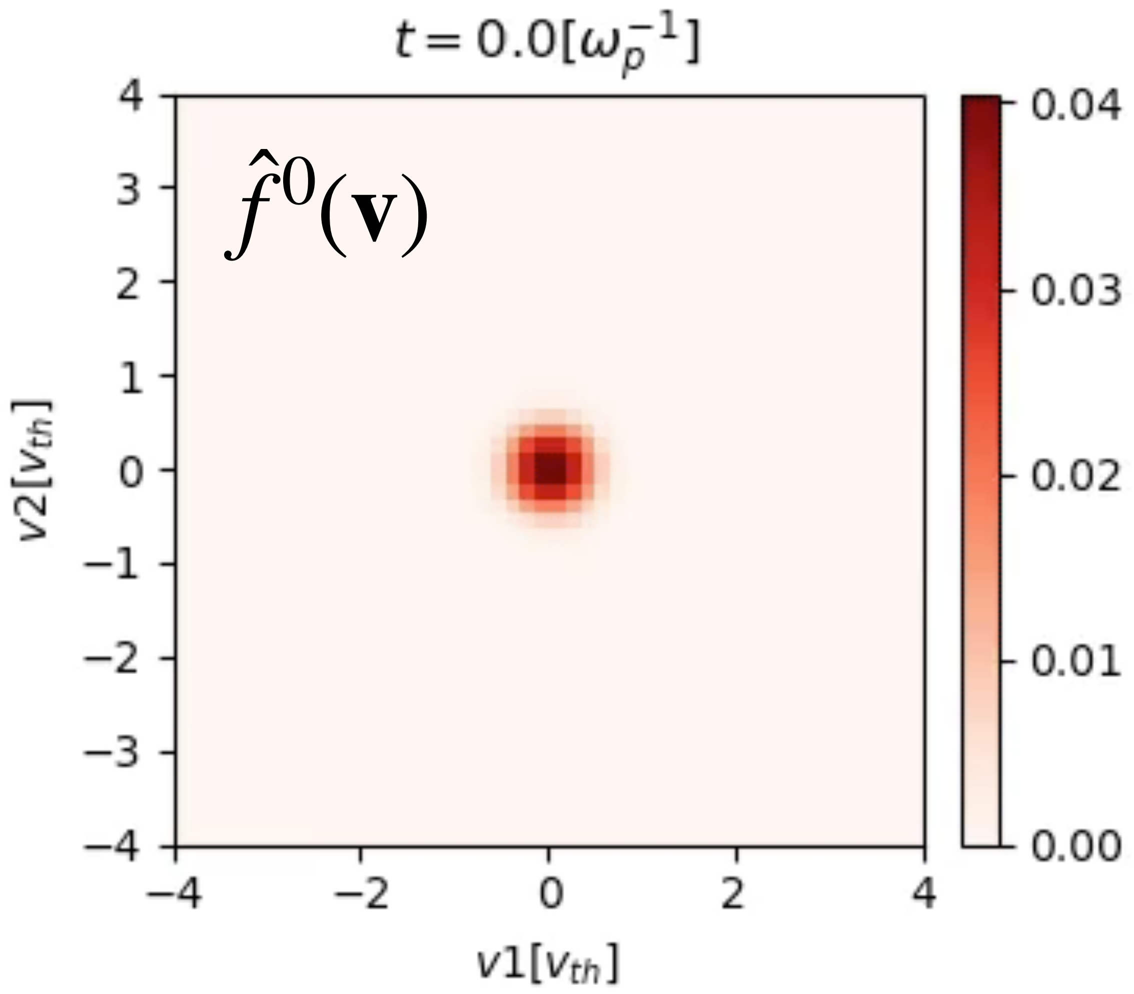


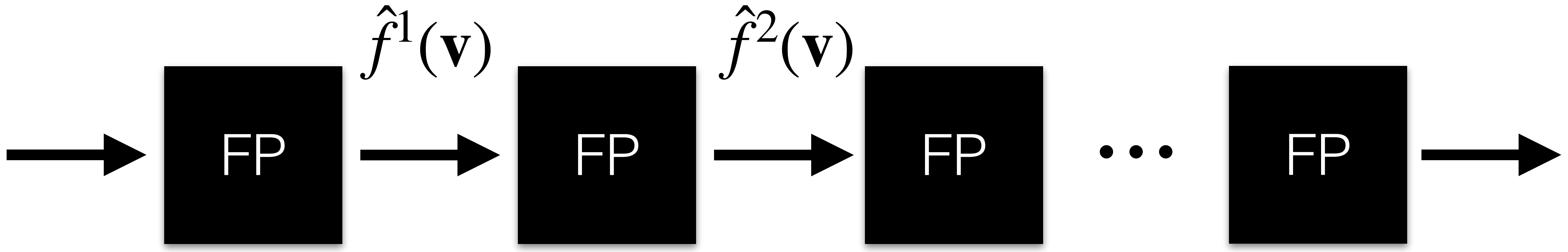


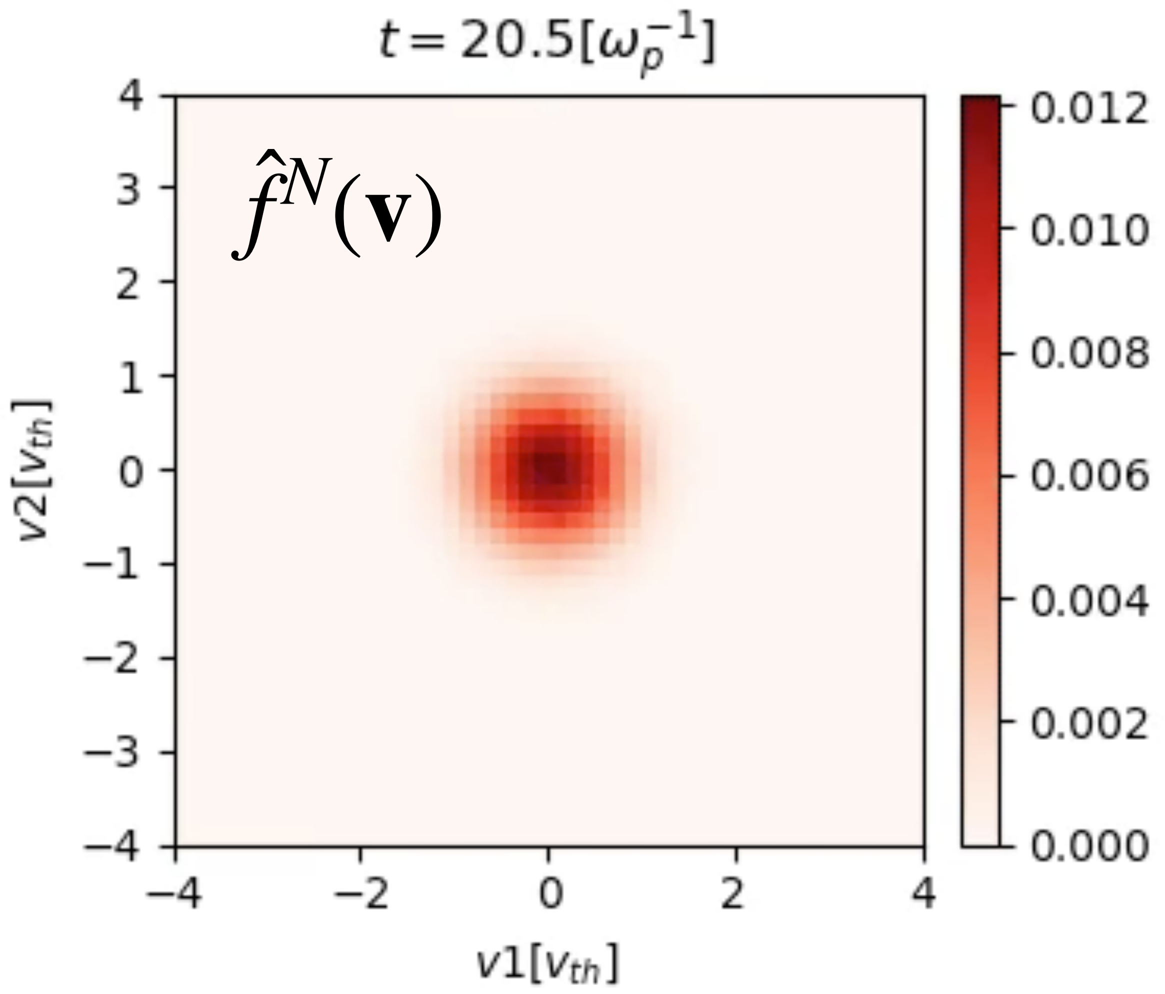
$$\hat{f}^{t+1} = \hat{f}^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} \hat{f}^t \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} \hat{f}^t \right) \right)$$

D. Carvalho et al., in preparation for JPP (2025)

Learning advection / diffusion from evolution of sub-populations







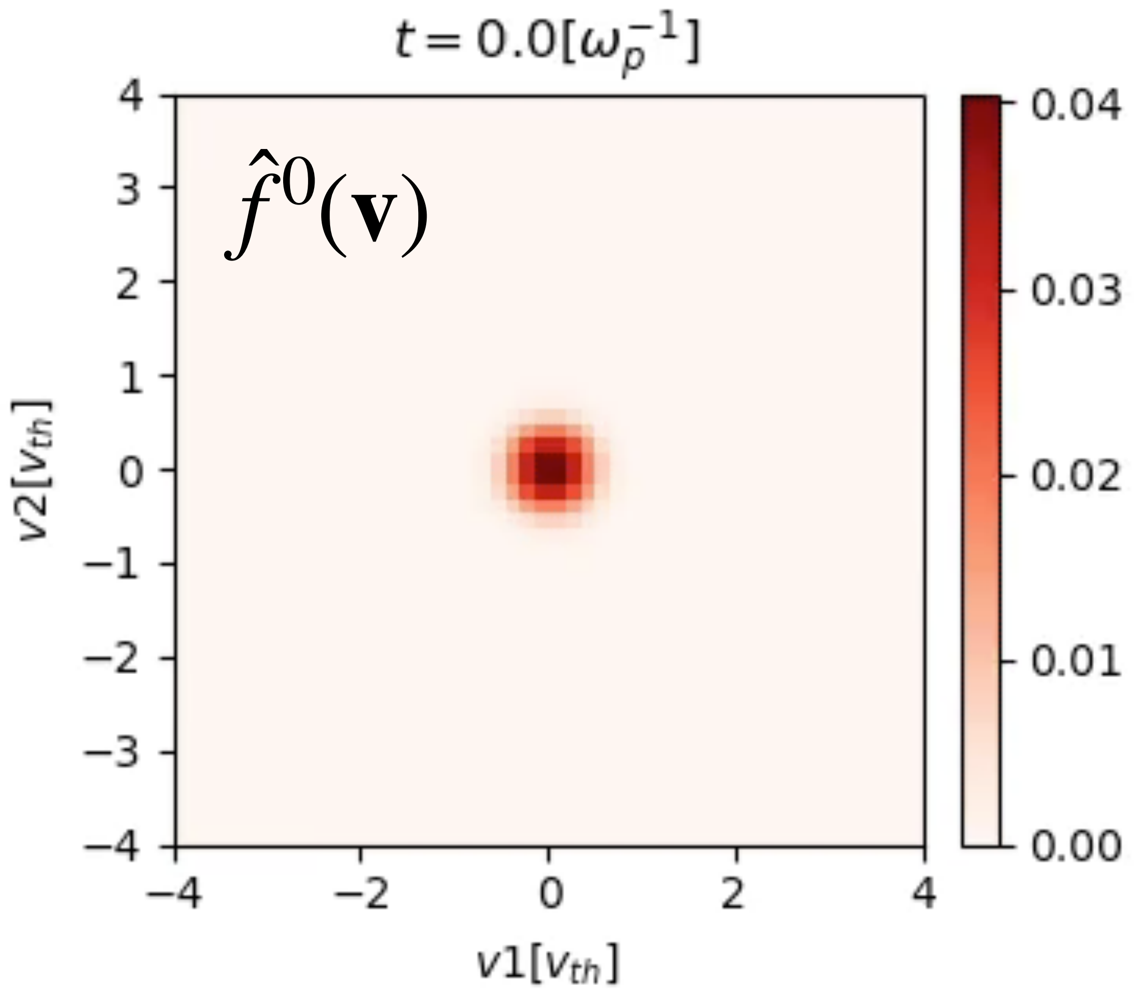
$$\hat{f}^{t+1} = \hat{f}^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} \hat{f}^t \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} \hat{f}^t \right) \right)$$

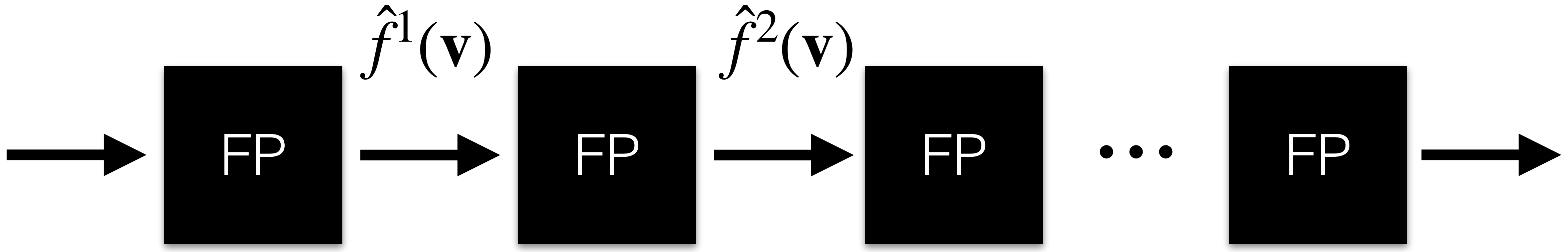
We can make the **Fokker-Planck solver differentiable** and frame this as an **optimisation task**

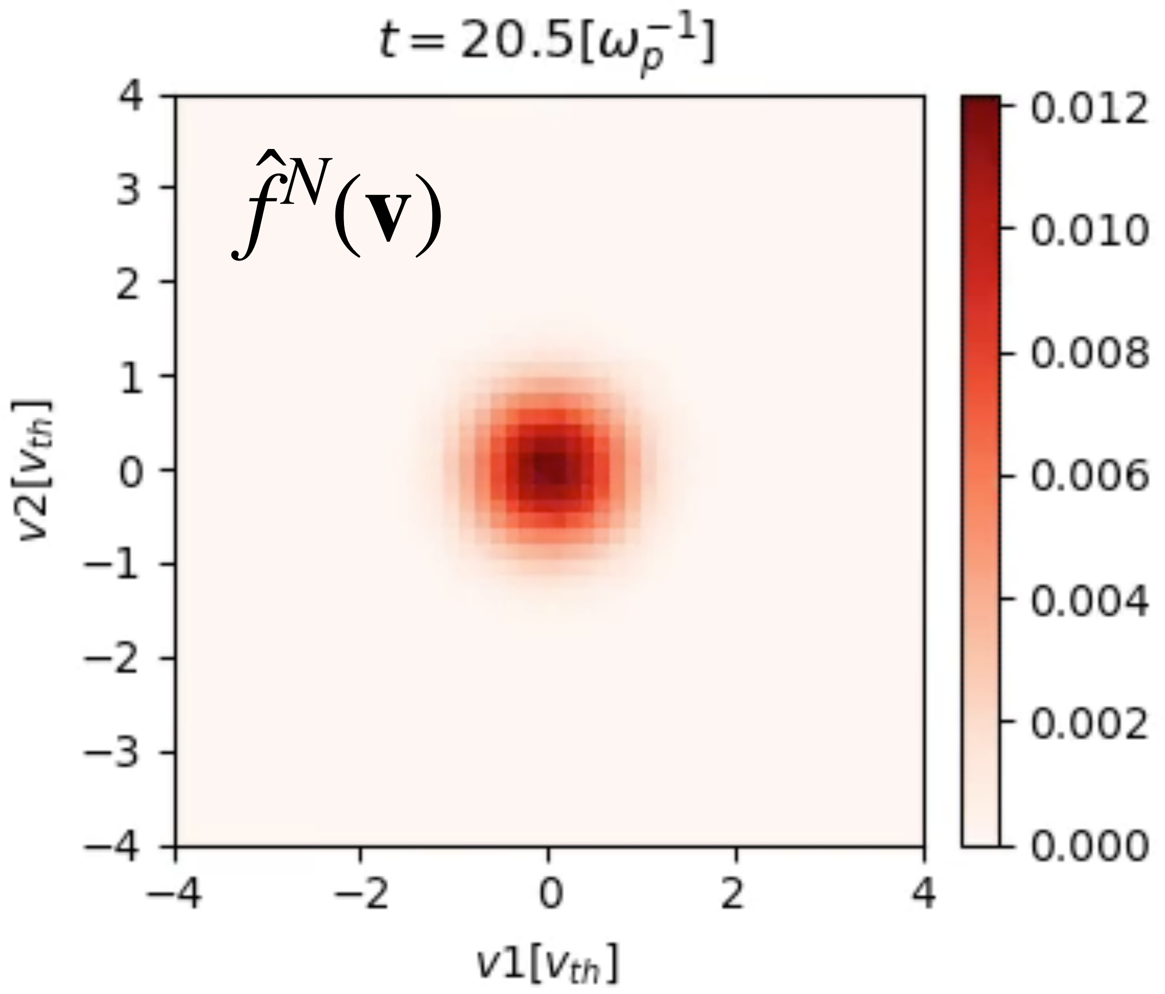
$$\min_{\mathbf{A}, \overleftrightarrow{\mathbf{D}}} \left\| \hat{f}^N_{predicted} - \hat{f}^N_{true} \right\|$$

D. Carvalho et al., in preparation for JPP (2025)

Learning advection / diffusion from evolution of sub-populations







$$\hat{f}^{t+1} = \hat{f}^{t} + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} \hat{f}^{t} \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} \hat{f}^{t} \right) \right)$$

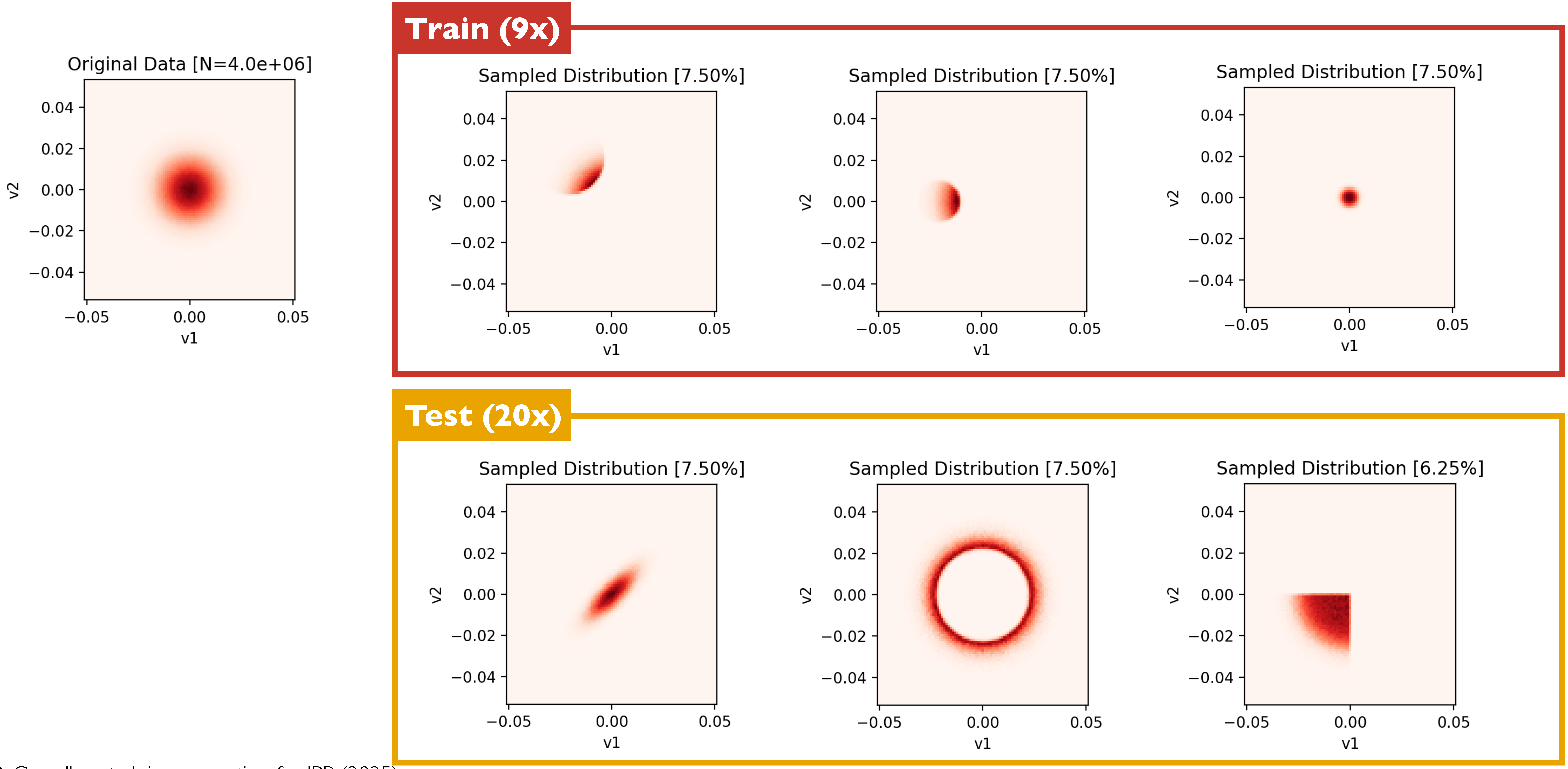
We can make the **Fokker-Planck solver differentiable** and frame this as an **optimisation task**

$$\min_{\mathbf{A}, \overleftrightarrow{\mathbf{D}}} \left\| \hat{f}^{N}_{predicted} - \hat{f}^{N}_{true} \right\|$$

This is an ill-posed problem (there exists a family of solutions) → Train with multiple sub-populations

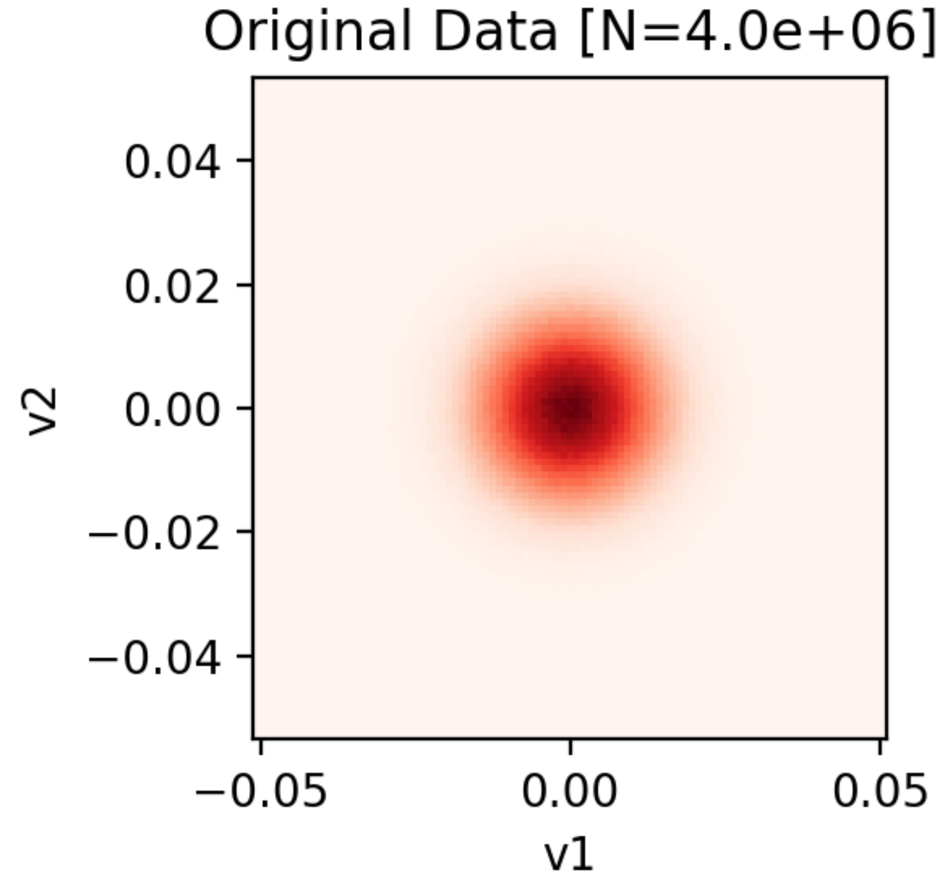
D. Carvalho et al., in preparation for JPP (2025)

We use different sets of training and test sub-populations

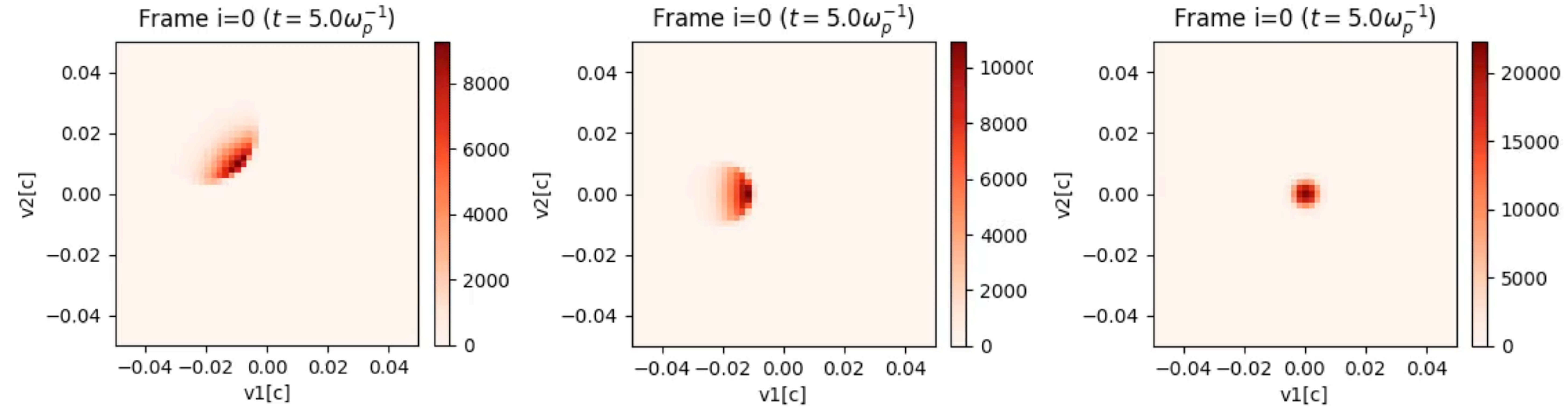


D. Carvalho et al., in preparation for JPP (2025)

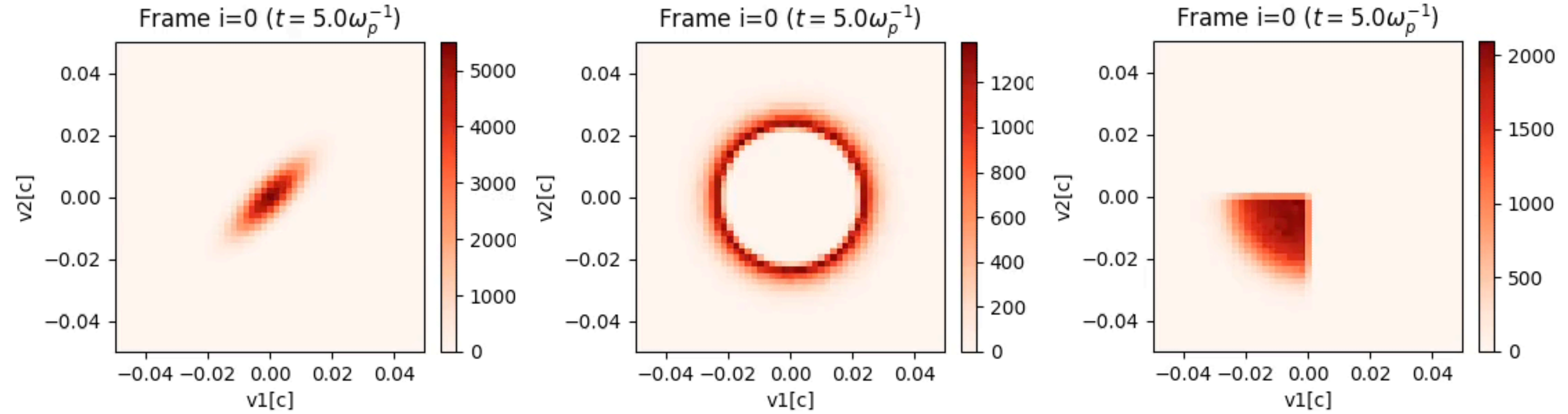
We use different sets of training and test sub-populations



Train (9x)

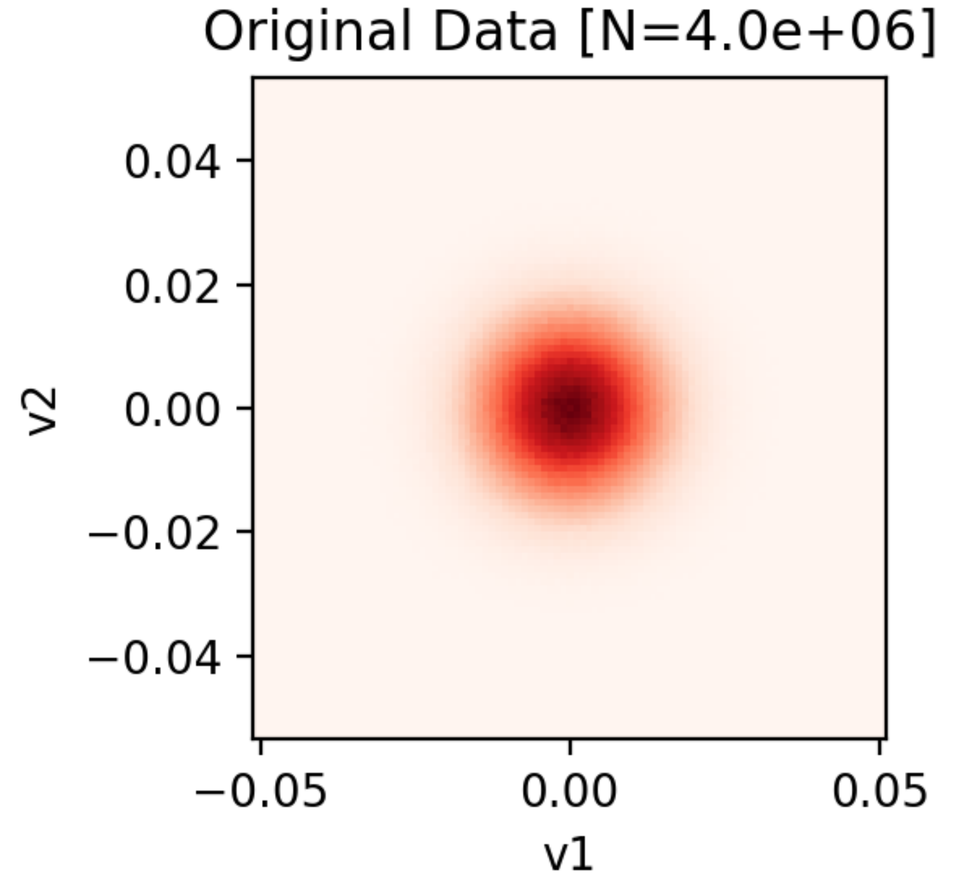


Test (20x)

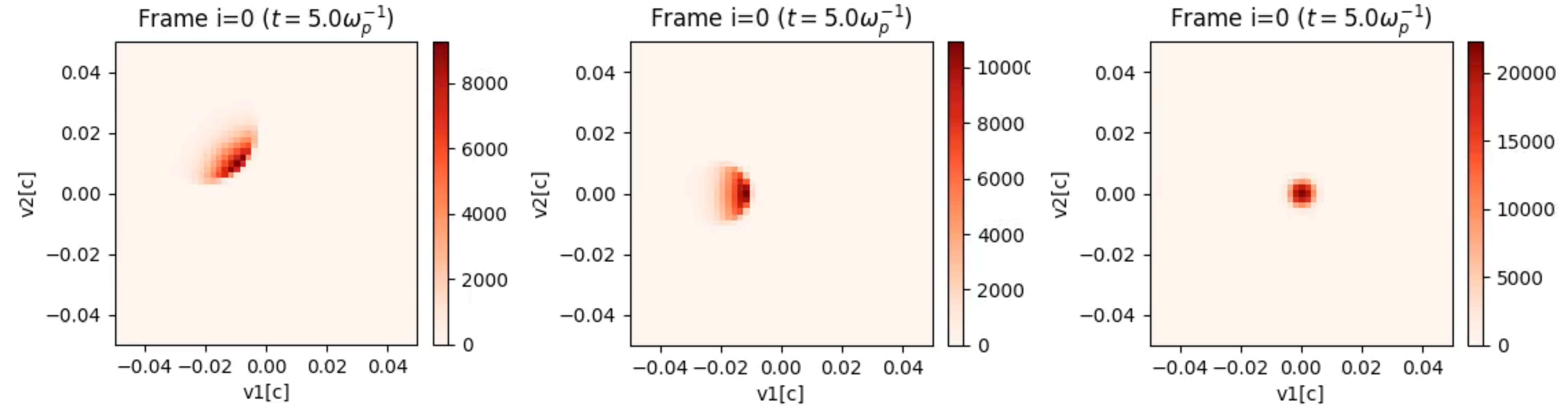


D. Carvalho et al., in preparation for JPP (2025)

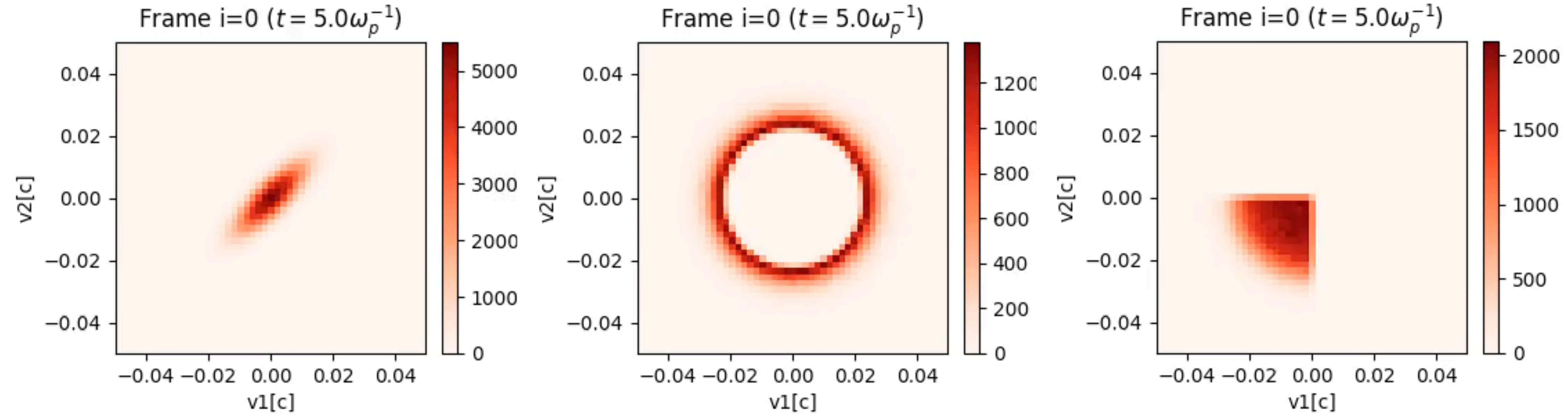
We use different sets of training and test sub-populations



Train (9x)



Test (20x)



D. Carvalho et al., in preparation for JPP (2025)

We can parameterise A/D using a Tensor (discrete) or a NN (continuous)

Tracks

Tensor

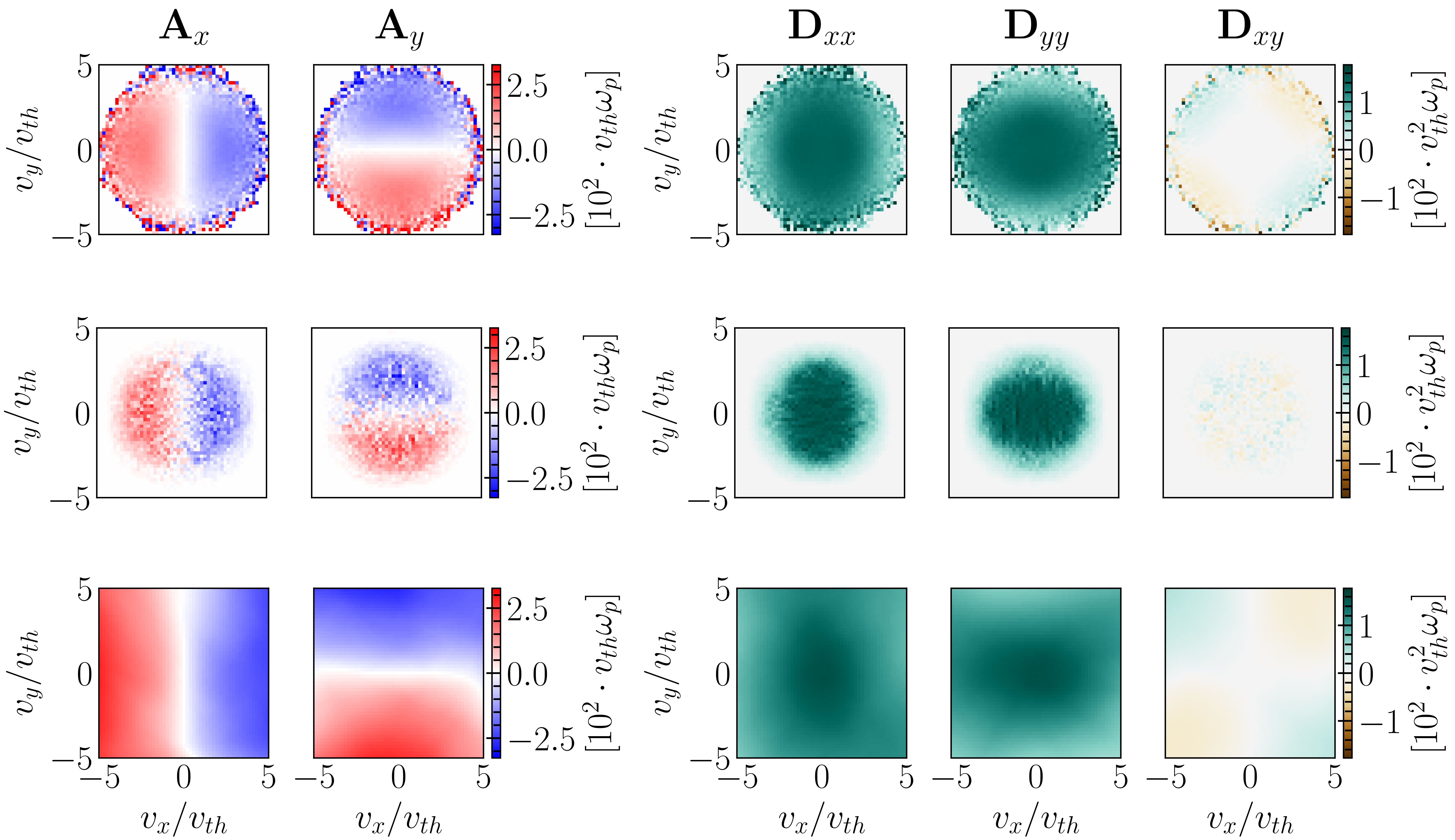
$A_i[v_x, v_y]$

$D_{ij}[v_x, v_y]$

NN

$A_i = MLP(v_x, v_y)$

$D_{ij} = MLP(v_x, v_y)$



D. Carvalho et al., in preparation for JPP (2025)

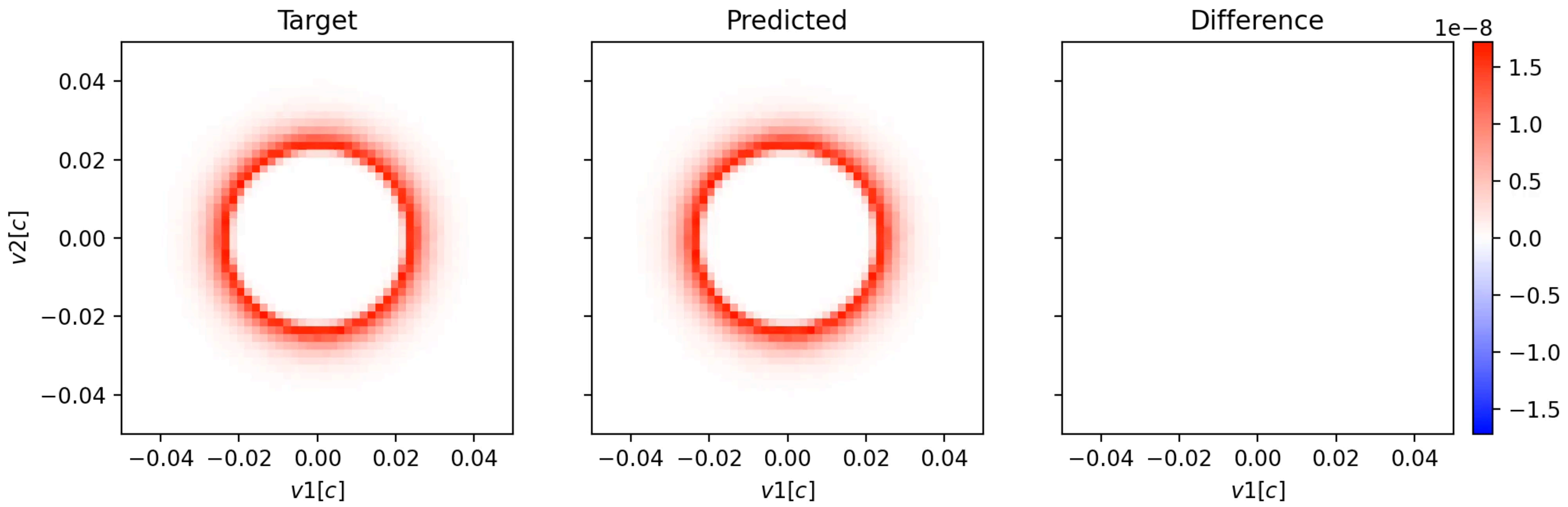
Can these operators reproduce dynamics?

$$f^{t+1} = f^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} f^t\right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} f^t \right) \right)$$

D. Carvalho et al., in preparation for JPP (2025)

Can these operators reproduce dynamics? (Yes)

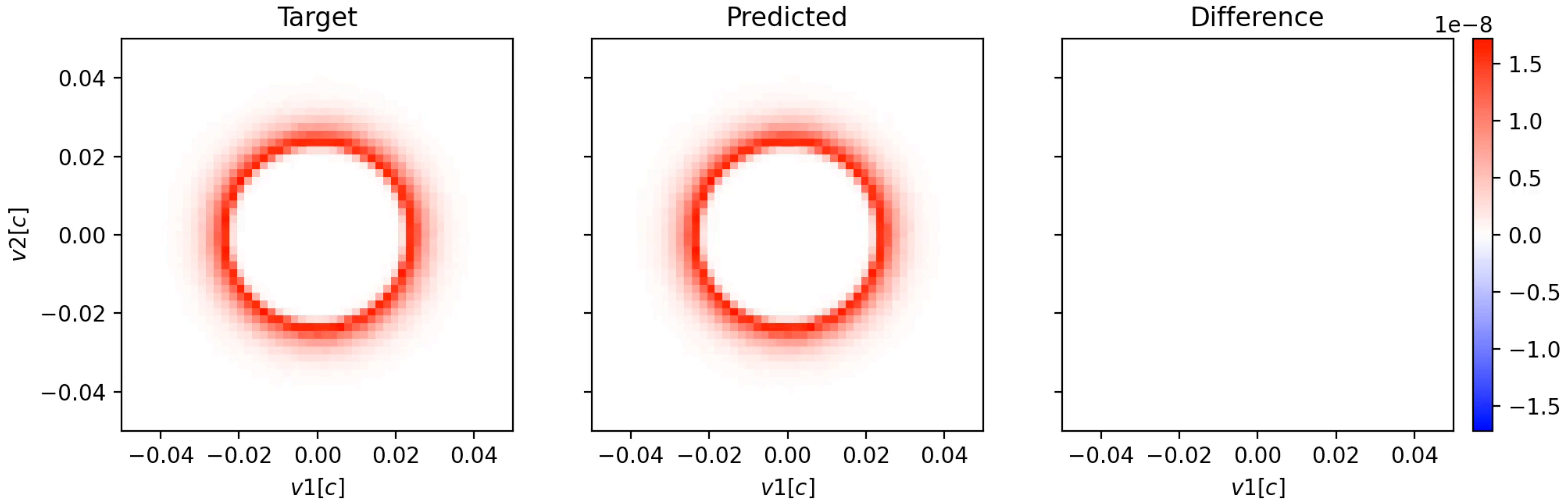
$$f^{t+1} = f^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} f^t\right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} f^t \right) \right)$$



D. Carvalho et al., in preparation for JPP (2025)

Can these operators reproduce dynamics? (Yes)

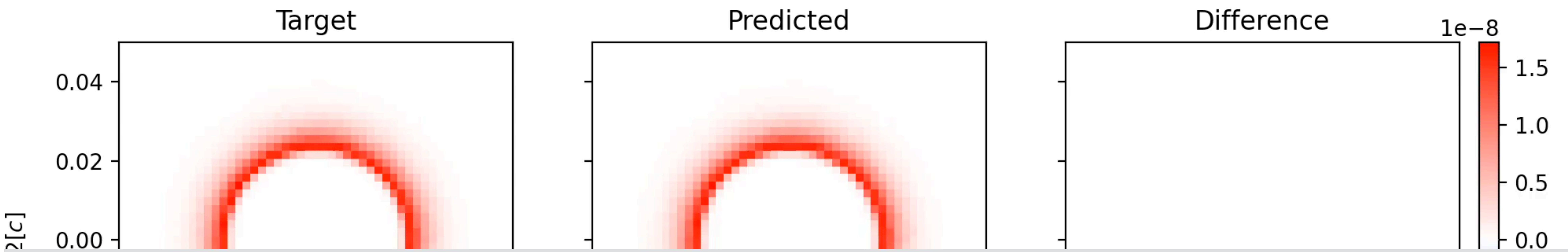
$$f^{t+1} = f^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} f^t \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} f^t \right) \right)$$



D. Carvalho et al., in preparation for JPP (2025)

Can these operators reproduce dynamics? (Yes)

$$f^{t+1} = f^t + \Delta t \left(-\nabla_{\mathbf{v}} \cdot \left(\mathbf{A} f^t \right) + \frac{1}{2} \nabla_{\mathbf{v}} \nabla_{\mathbf{v}} \cdot \left(\overleftrightarrow{\mathbf{D}} f^t \right) \right)$$



Next steps

General purpose library: from phase space data, retrieve **A** and **D** (to be inserted on Fokker-Planck codes) for varying plasma conditions, and from different sources of data

Sub-module to capture (PIC or other) collisions for mesoscale simulations

Meta analysis: use different A and D for different plasma conditions (n, B, T) to learn more general behaviour e.g. via sparse regression

D. Carvalho et al., in preparation for JPP (2025)

MC models in PIC simulations

New simulator models - GNN collisional plasma model

Learning advection and diffusion coefficients

The (ground) truth? - collisions in PIC codes

What is the ground truth?

Klimontovich + Maxwell's equations

$$\frac{\partial N}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} N - \frac{q}{m} \left(\mathbf{E}^m + \mathbf{v} \times \mathbf{B}^m \right) \cdot \nabla_{\mathbf{v}} N = 0$$

This is the particle-in-cell algorithm (with finite-size particles): statistical mechanics is well-known (e.g. H. Okuda and C. Birdsall (1970), R. Hockney (1971), M. Touati et al. (2022), S. Jubin et al. (2024)) + Born-Infeld electrodynamics

What if the cell/particle size is shorter than the classical electron radius?

What are the challenges of running << 1 ppc?

Field initialization becomes critical + Computation determined by grid (N^3 or N^2)

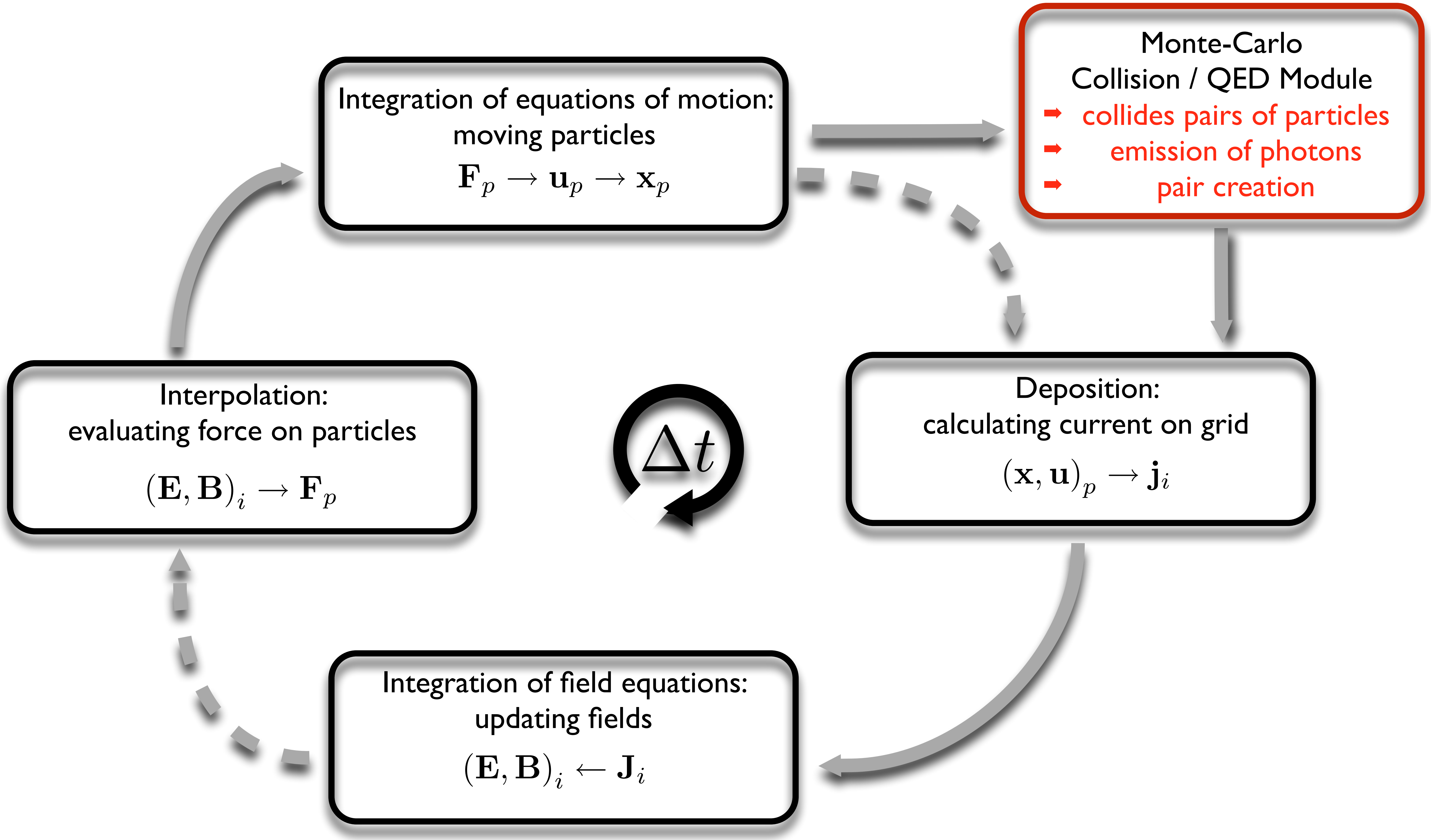
Numerical heating (still need to resolve Debye length) + Very small time steps (CFL) + numerical transition radiation

Validation against theory (but theory is very limited - only 2D) or computational models (MD non relativistic)

Shape functions to capture quantum effects?

D. Carvalho et al., in preparation (2025)

The PIC loop by itself should be able to model collisional dynamics

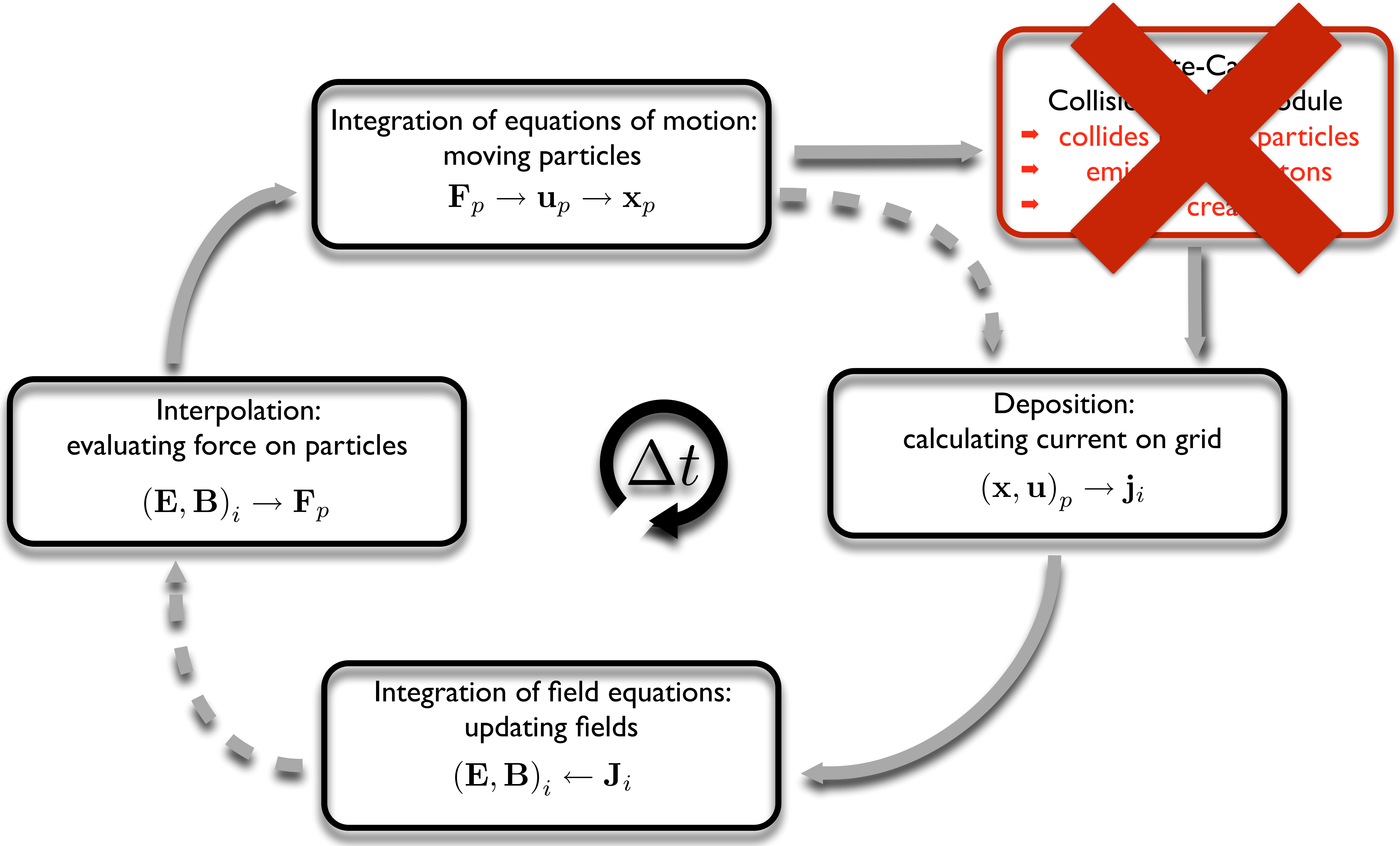


$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}$$

Nonlinear collisional absorption in laser-driven plasmas
C. D. Decker *et al.*, Phys. Plasmas 1, 4043–4049 (1994)
Disorder induced heating
M. D. Acciarri *et al.*, Plasma Sources Sci. Technol. 33 035009 (2024)

The PIC loop by itself should be able to model collisional dynamics



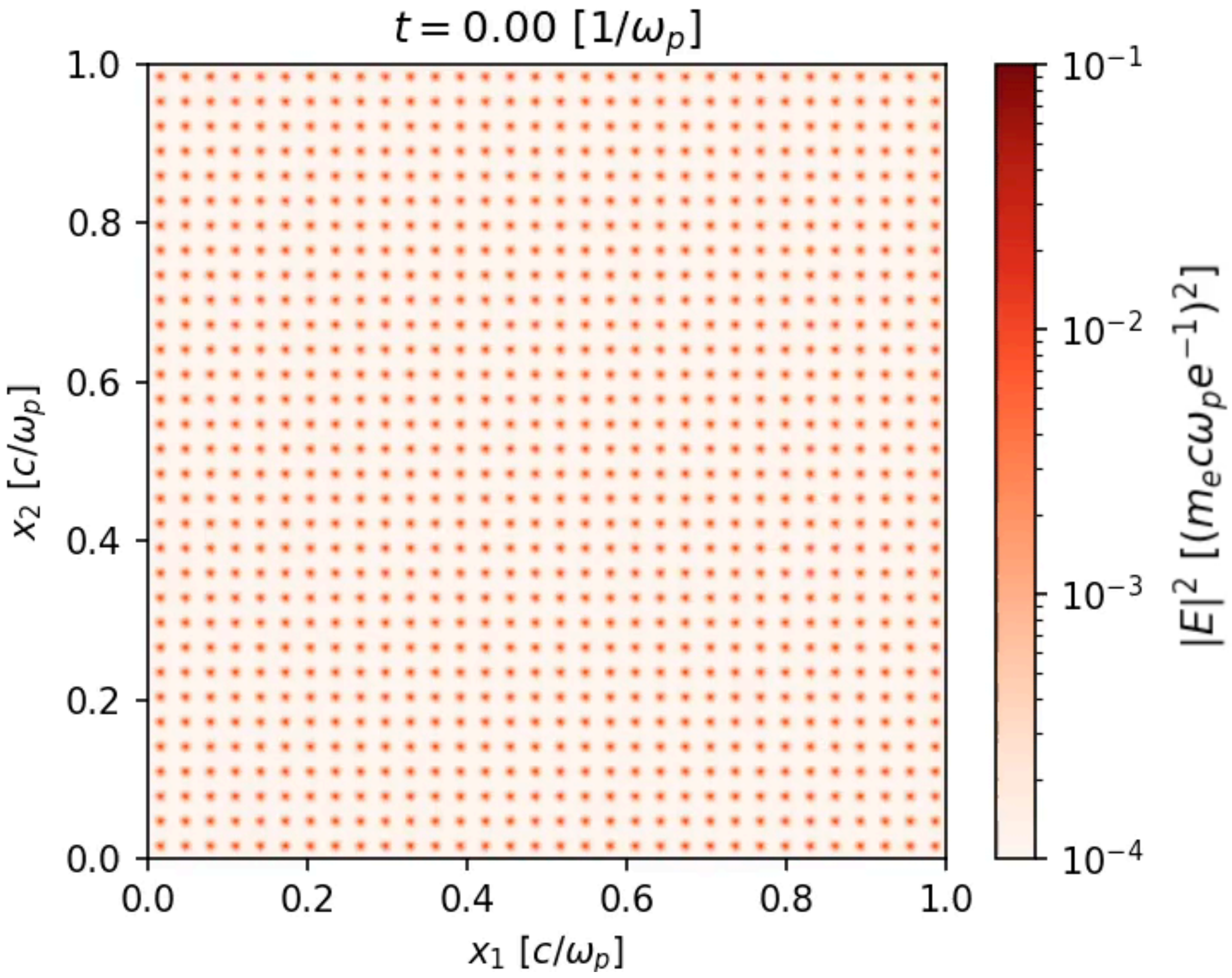
$$\frac{\partial \mathbf{E}}{\partial t} = c\nabla \times \mathbf{B} - 4\pi \mathbf{j}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}$$

Nonlinear collisional absorption in laser-driven plasmas
C. D. Decker *et al.*, Phys. Plasmas 1, 4043–4049 (1994)
Disorder induced heating
M. D. Acciarri *et al.*, Plasma Sources Sci. Technol. 33 035009 (2024)

Simulating self-consistent collisions with PIC

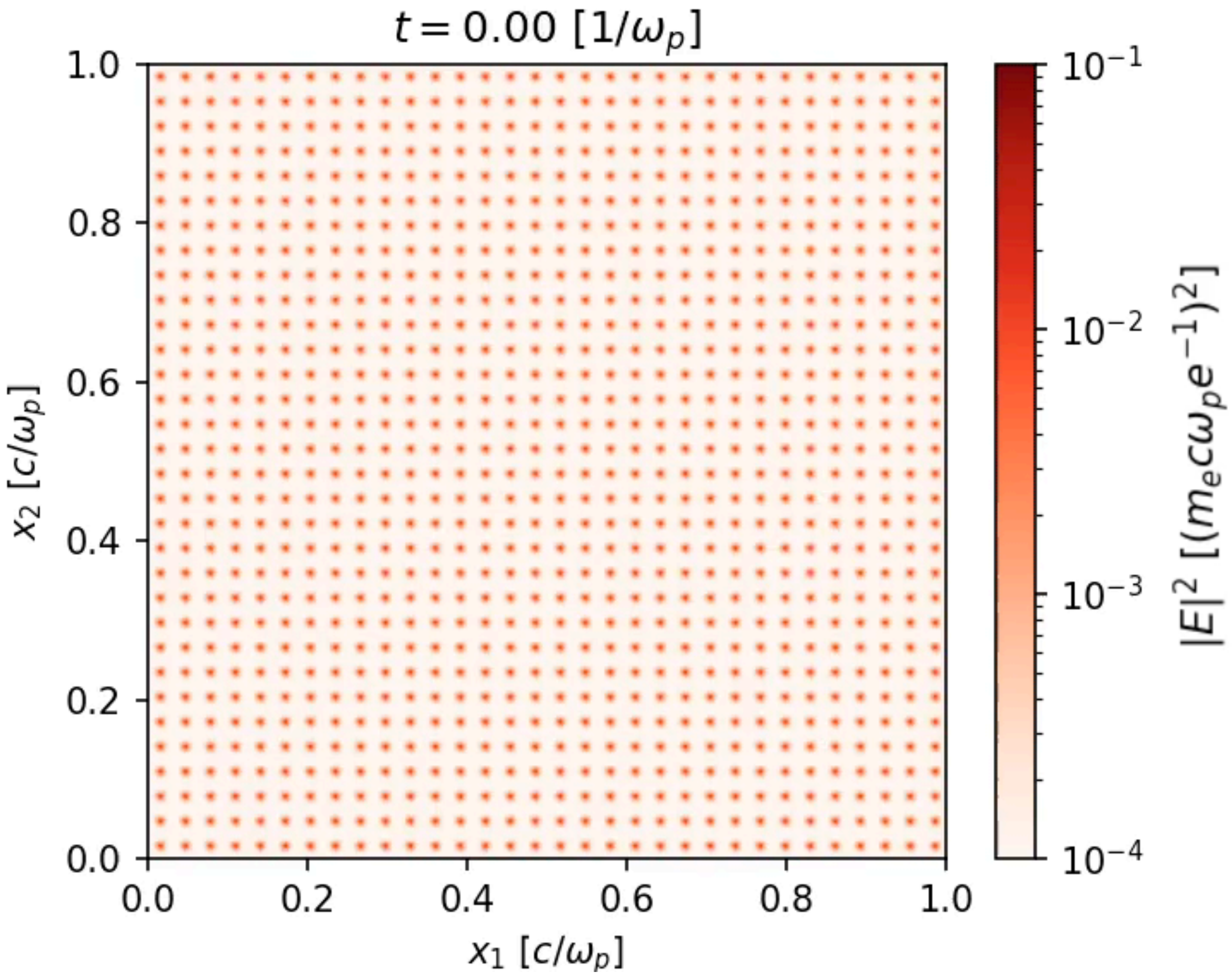
$n\lambda_D^2 \simeq 0.1$ (Average interparticle distance $\gg \lambda_D$)



D. Carvalho et al., in preparation (2025)

Simulating self-consistent collisions with PIC

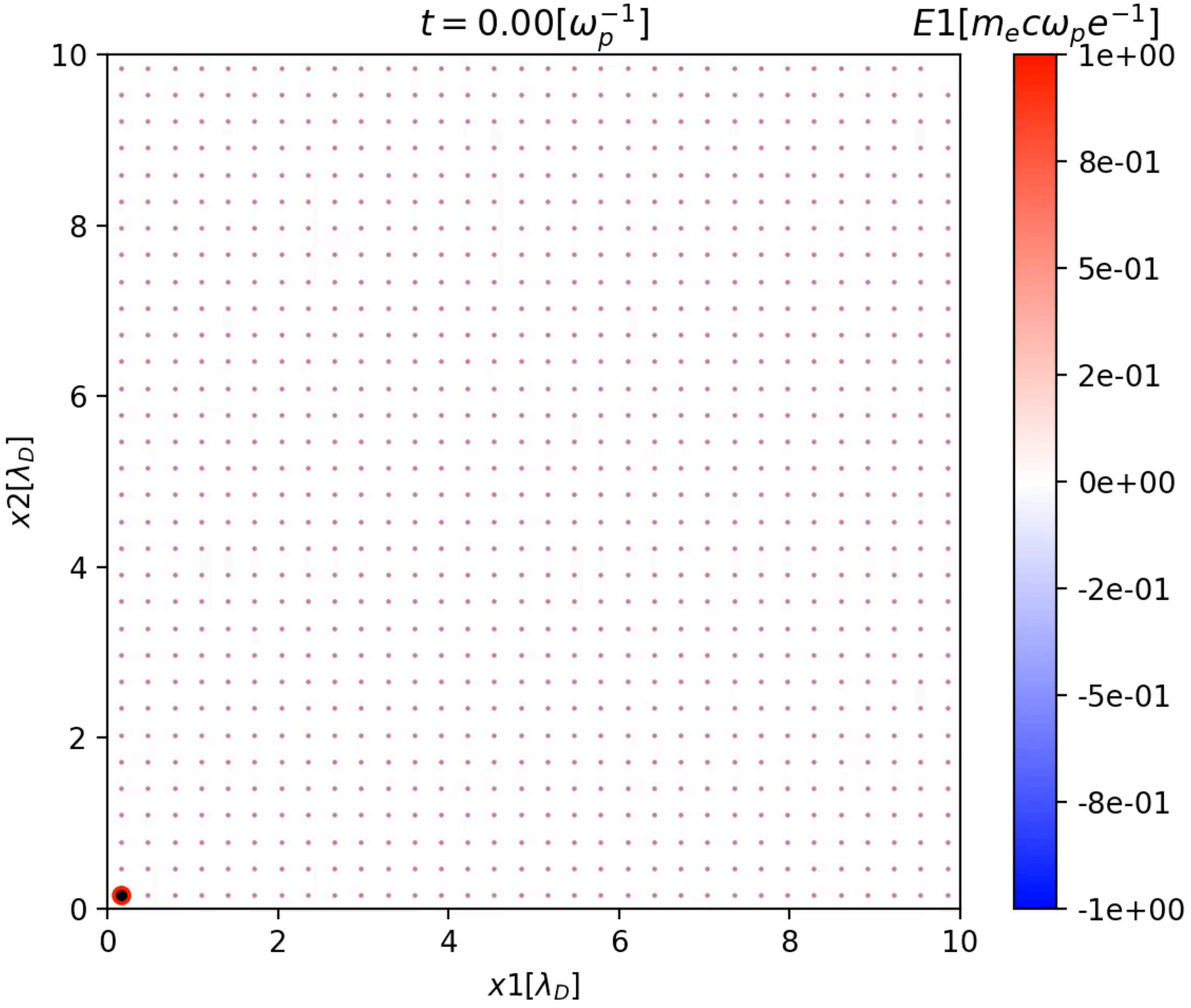
$n\lambda_D^2 \simeq 0.1$ (Average interparticle distance $\gg \lambda_D$)



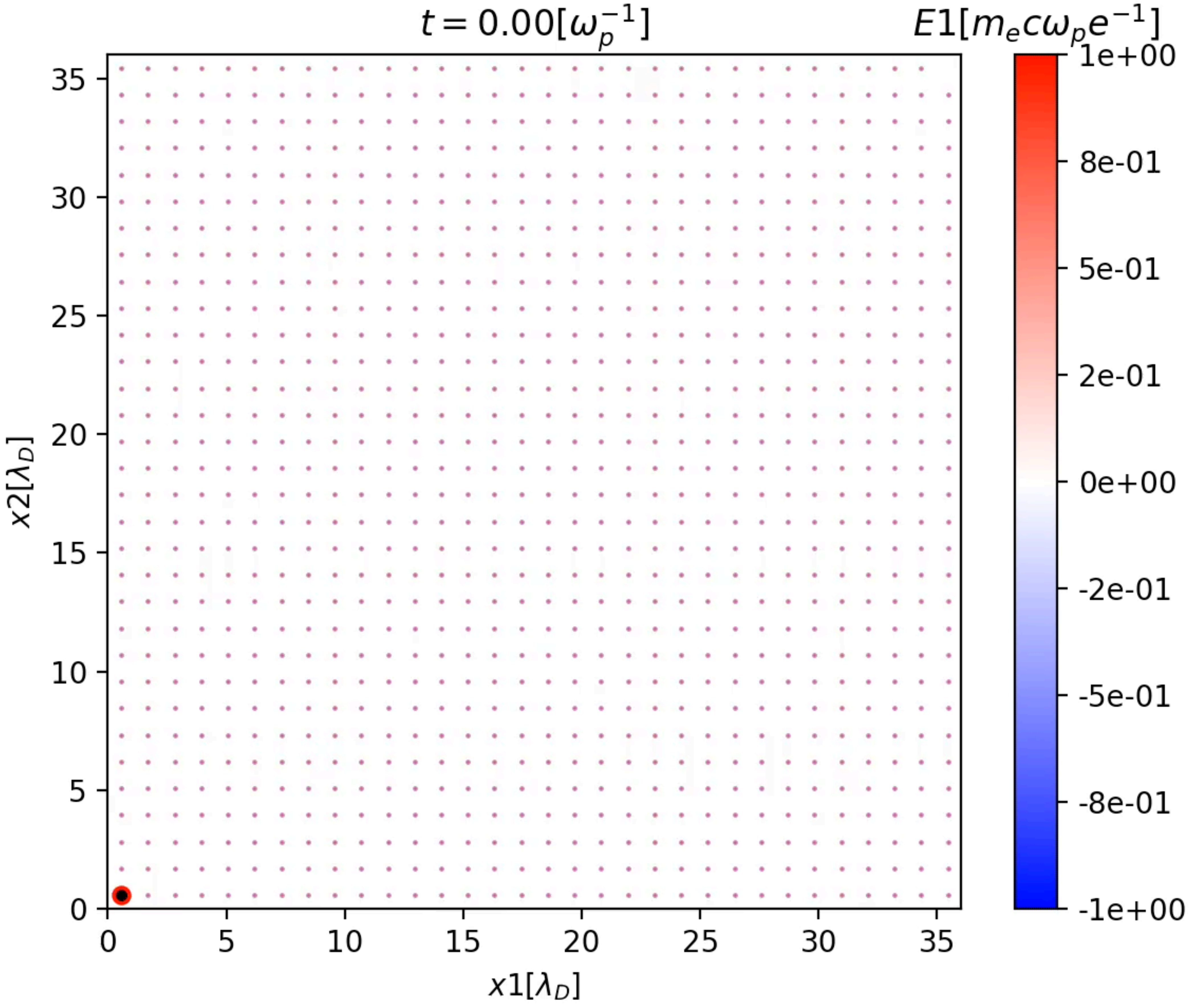
D. Carvalho et al., in preparation (2025)

Can we learn operators that describe the collisional dynamics?

$N_D^{2D} \simeq 32$



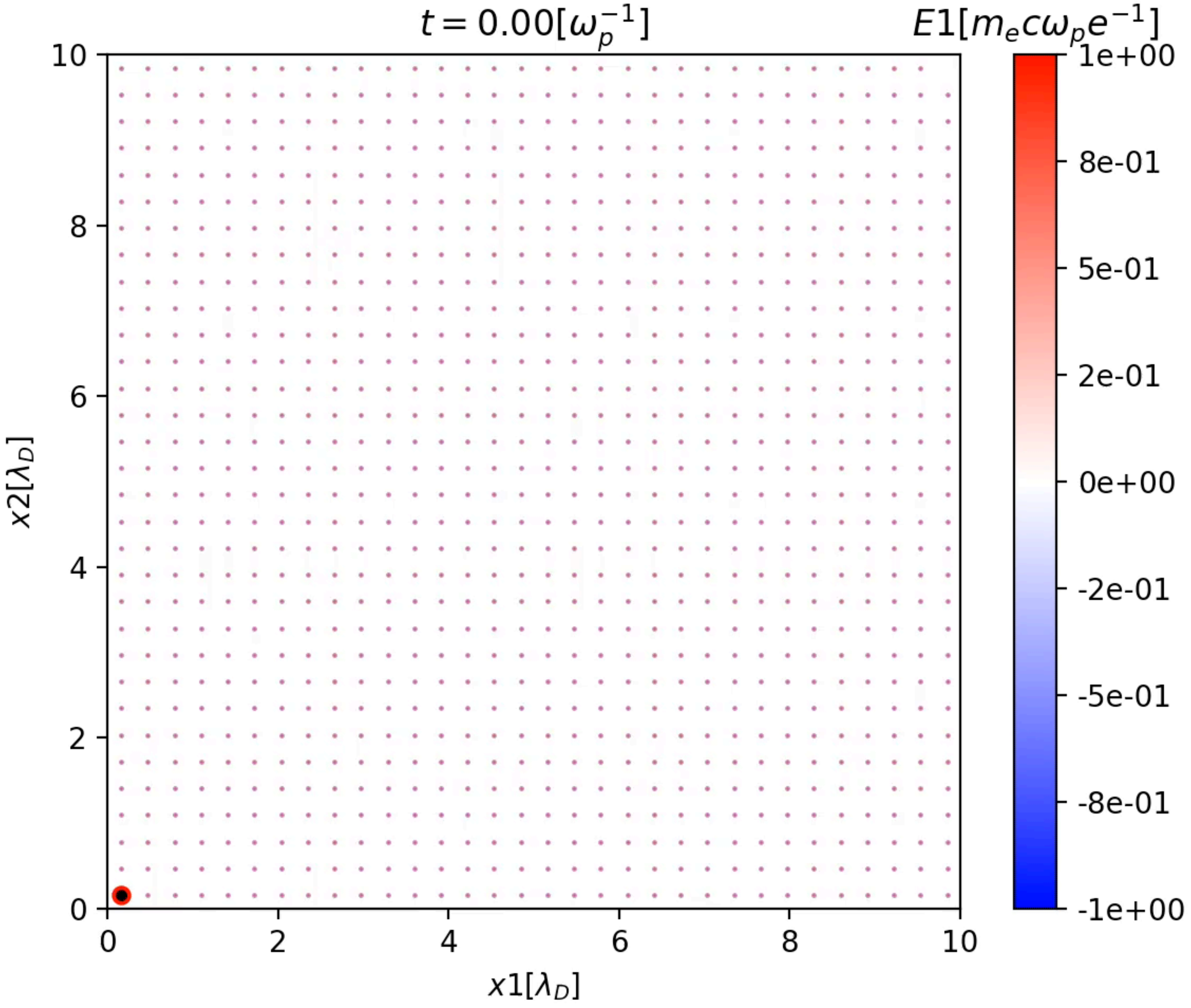
$N_D^{2D} \simeq 2.5$



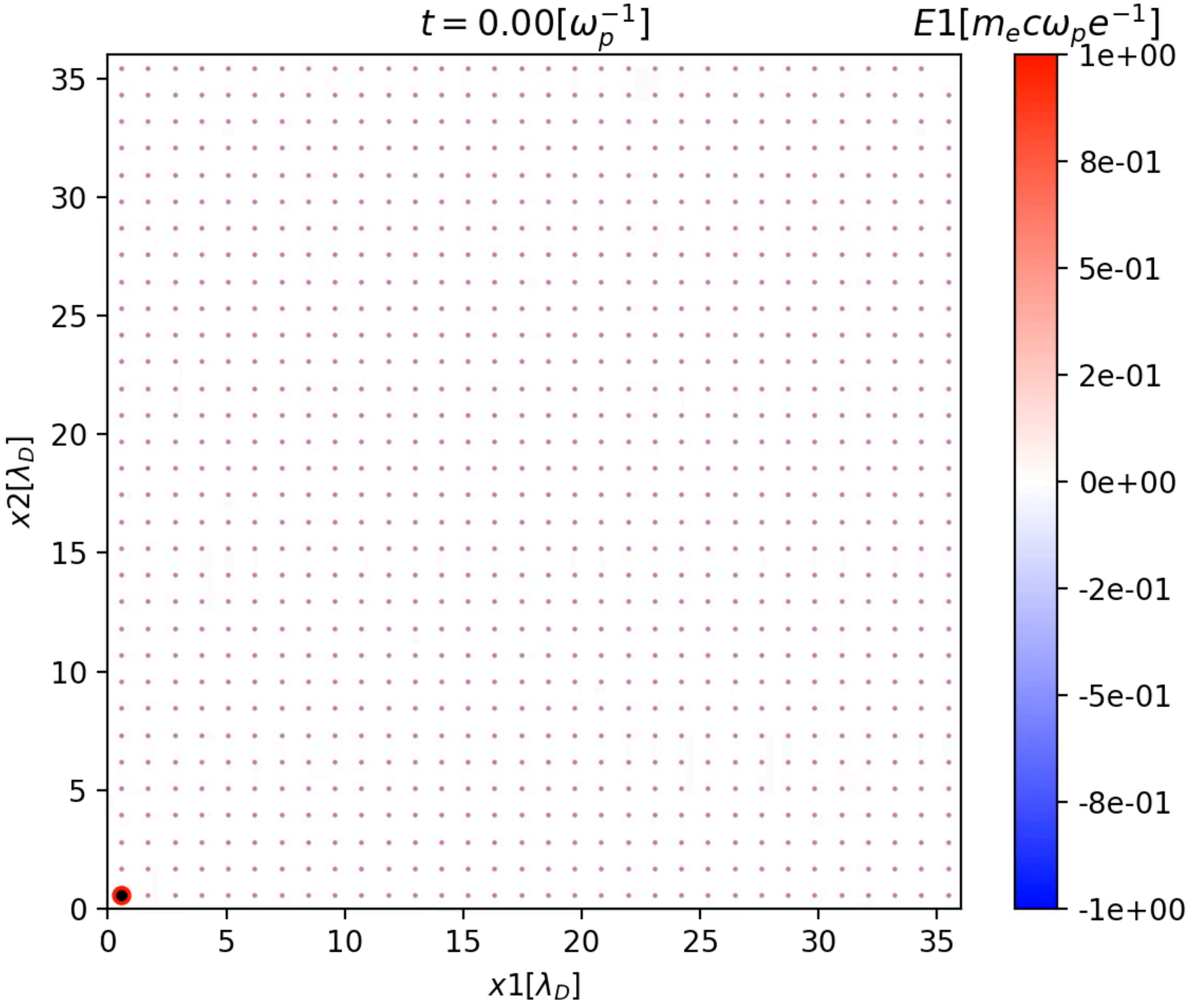
D. Carvalho et al., in preparation (2025)

Can we learn operators that describe the collisional dynamics?

$N_D^{2D} \simeq 32$



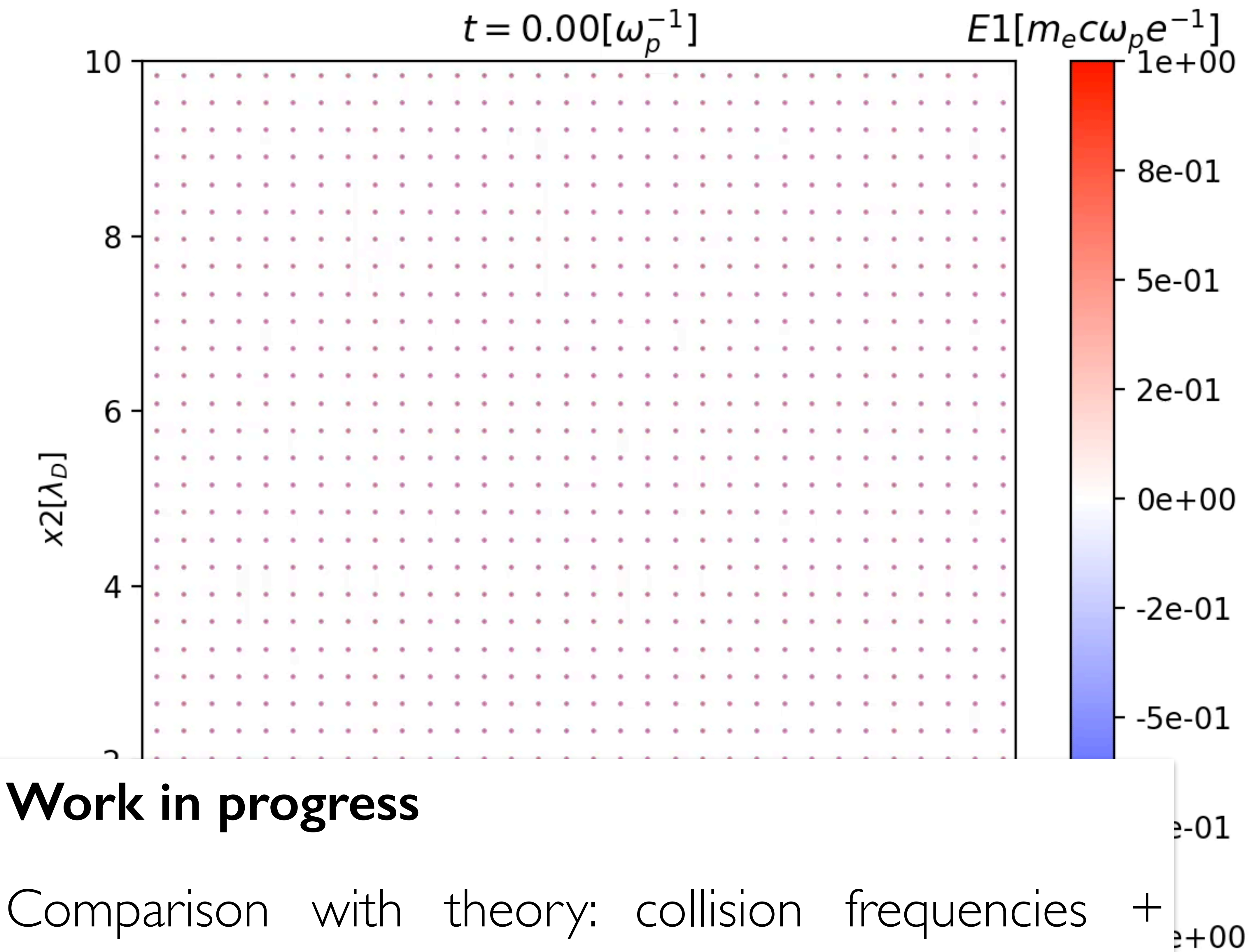
$N_D^{2D} \simeq 2.5$



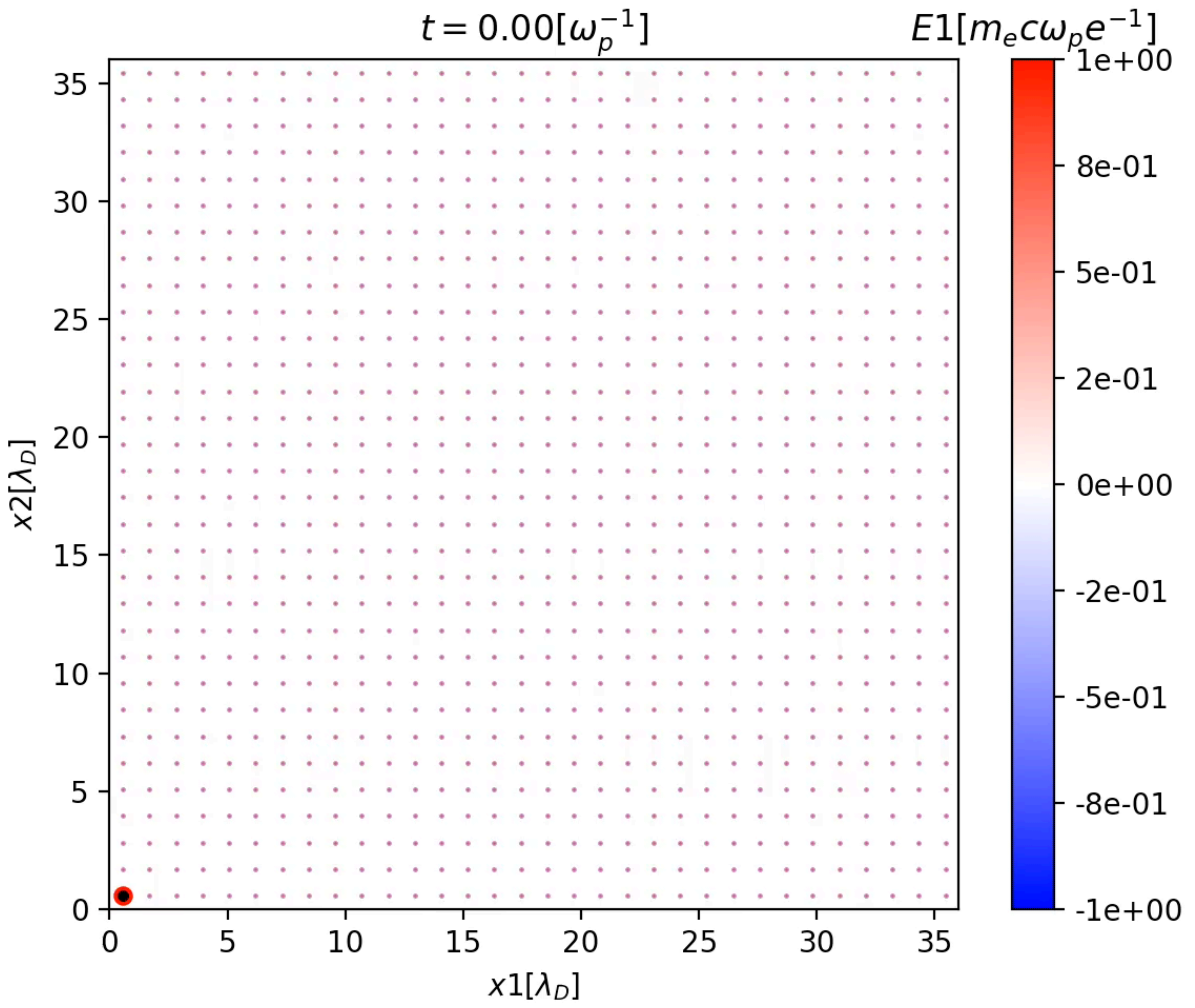
D. Carvalho et al., in preparation (2025)

Can we learn operators that describe the collisional dynamics?

$N_D^{2D} \simeq 32$



$N_D^{2D} \simeq 2.5$



Work in progress

Comparison with theory: collision frequencies + **A** and **D** and 2D Fokker-Planck model (Morales et al.)

What about B fields?

D. Carvalho et al., in preparation (2025)

Can ML help us speed up standard plasma simulators? No speed up but huge memory gains

Can we build faster ML based simulators? Yes, but with significant modifications to the algorithms/structure/philosophy

What can we learn from data-driven approaches + ML? It looks like we can learn a lot (and the community is learning how to do it): e.g. collision operators

Can standard plasma simulators provide "high quality data" for data-driven discovery? Yes, pushing for additional developments in HPC simulations

Interplay between HPC and AI is just starting: "There are (many) unknown unknowns" (which is great for science!)