NEURAL FIELD-BASED CHARACTERISTIC RECONSTRUCTION FOR VLASOV–POISSON DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

The particle distribution in Vlasov–Poisson dynamics develops increasingly fine structures over time, making the full distribution difficult for a neural network to represent directly, whereas the electric field driving particle trajectories depends only on position and time and usually remains comparatively smooth. Motivated by this difference, we introduce Neural Field-Based Characteristic Reconstruction (NFCR), in which the neural network represents only the electric field rather than directly learning the full distribution. A time network outputs a finite set of spatial Fourier coefficients over the training interval, and differentiable backward characteristic integration reconstructs the evolving distribution from the prescribed initial condition. The density and current obtained from this distribution are related to the network output through the Poisson and Ampère equations, keeping the electric field consistent with the reconstructed distribution; conditions on the initial field and its time derivative constrain the dynamics near the beginning of the interval. The neural network therefore needs to learn only a finite number of electric-field coefficients, while fine distribution scales are generated naturally by transport, without an evolving phase-space grid or reference labels. Backward trajectories are integrated with a symmetric drift–acceleration–drift scheme. This scheme locally preserves phase-space area during backward tracing and ensures that a distribution reconstructed from a nonnegative initial condition remains nonnegative. Experiments on Gaussian transport and Landau damping show that NFCR reconstructs the distribution with lower errors than neural methods that directly learn the distribution or the full particle-trajectory map, while accurately capturing the electric-field evolution and maintaining small mass drift.

1 INTRODUCTION

In Vlasov–Poisson dynamics, the quantity that is hardest to represent is not necessarily the quantity most worth learning directly with a neural network. During phase-mixing processes such as Landau damping, a smooth initial distribution can gradually develop fine velocity-space structure. The electric field driving particle trajectories, by contrast, depends only on position and time: integrating the distribution over velocity gives the density, and Poisson's equation then determines the field. Numerical studies have shown that the electric field often remains comparatively smooth even when pronounced filamentation appears in the distribution (Kirchhart & Wilhelm, 2024). This difference in representational complexity between a progressively filamenting distribution and a comparatively smooth field provides a basis for reconsidering what the neural network should learn.

Existing methods incur different representation and computational costs when resolving these fine-scale structures. Eulerian schemes must resolve the increasingly fine structure over the entire phase space. Classical semi-Lagrangian (SL) methods trace characteristics backward, but repeated interpolation from an evolving distribution grid accumulates smoothing and discretization errors (Cheng & Knorr, 1976; Sonnendrücker et al., 1999). A physics-informed neural network (PINN) removes a fixed solution grid, yet still uses a global network to represent the progressively filamenting distribution and learns its evolution by minimizing the Vlasov-equation residual (Raissi et al., 2019; Zhang et al., 2023). These limitations suggest that a more suitable representation should let the network learn the low-dimensional physical quantity that determines transport, while characteristic dynamics generate the phase-space details.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A semi-Lagrangian physics-informed neural network (SL-PINN) exploits the constancy of distribution values along characteristics. It learns the inverse flow map from a query point (t, x, v) to initial coordinates (X_0, V_0) and composes this map with the initial condition (Monti et al., 2026). The network therefore predicts a particle's initial position and velocity, while still taking the query time, position, and velocity as input. Numerical Flow Iteration (NuFI), on the other hand, stores an electric-field history and uses it to reconstruct the current distribution from the initial distribution and compute a self-consistent field (Kirchhart & Wilhelm, 2024). The former shows that a distribution can be obtained by composing the inverse flow map with the initial condition, whereas the latter shows that this map can be determined by field-driven characteristic dynamics. Together, these two approaches provide a basis for further reducing what the neural network must learn.

Based on these ideas, we introduce Neural Field-Based Characteristic Reconstruction (NFCR). NFCR uses a time network to output a finite set of spatial Fourier coefficients, from which it obtains a continuously queryable electric field. This field drives differentiable numerical backward characteristics that map any query point to the prescribed initial condition, thereby generating the distribution, density, and current. These velocity moments then constrain the electric field represented by the network through the Poisson and Ampère relations. Training thus forms a loop from field to transport, from transport to velocity moments, and from velocity moments back to field, without reference trajectories or evolved-distribution labels.

This construction makes three contributions. First, NFCR restricts its only neural state to a finite set of electric-field Fourier coefficients that vary with time, avoiding a global representation of the filamented full phase-space distribution. Distribution values are obtained by composing the initial condition with backward characteristic integration, so training and queries do not maintain an evolving phase-space distribution grid.

Second, NFCR uses a symmetric drift–acceleration–drift characteristic integrator. Drift updates position, acceleration updates velocity, and each substep has unit Jacobian determinant. Their composition therefore preserves phase-space area locally. If the initial distribution is nonnegative, evaluating it along the backward trajectories also preserves nonnegativity; positions outside the periodic domain are mapped to equivalent coordinates within the same period.

Third, the Poisson and Ampère relations make the density and current of the reconstructed distribution consistent with the field represented by the network, while initial-field and initial-derivative conditions constrain the dynamics near the beginning of the interval. Experiments on Gaussian transport and Landau damping compare distribution reconstruction error, electric-field evolution, velocity moments, and mass drift across methods.

2 Background and related work

2.1 Vlasov–Poisson coupling and representation complexity

We consider time $t \geq 0$, a periodic position interval $[x_{\min}, x_{\min} + L)$ of length L, and a truncated velocity interval $[v_{\min}, v_{\max}]$. Here $x_{\min}$ is the left spatial endpoint and $v_{\min}, v_{\max}$ are the velocity boundaries. The distribution, particle density, and electric field are $f(t, x, v)$, $\rho(t, x)$, and $E(t, x)$; ρ_{b} is the constant neutralizing density and $\kappa > 0$ is the coupling coefficient. We use $\partial_t, \partial_x, \partial_v$ for partial derivatives and $\langle u \rangle_x = L^{-1} \int_{x_{\min}}^{x_{\min}+L} u(x)\,\mathrm{d}x$ for spatial averaging. All variables are nondimensional, with the particle charge-to-mass ratio and its sign absorbed into the definition of E; the field therefore directly gives the rate of change of velocity with respect to nondimensional time in the characteristic equations. Under this convention, the classical Vlasov–Poisson system is (Glassey, 1996)

$$\partial_t f + v\partial_x f + E\partial_v f = 0, \qquad \partial_x E = \kappa(\rho - \rho_{\mathrm{b}}), \qquad \rho = \int_{v_{\min}}^{v_{\max}} f\,\mathrm{d}v, \qquad \langle E \rangle_x = 0. \tag{1}$$

Equation (1) gives the Vlasov equation, Poisson relation, density definition, and zero-mean field condition. Velocity integration gives the density, density determines spatial field variation, and the field accelerates the distribution.

Transport can generate fine scales from smooth nonuniform distributions, whereas density and field are obtained through velocity integration and Poisson's relation (Mouhot & Villani, 2011; Kirchhart

& Wilhelm, 2024). Microscopic complexity need not transfer fully to the macroscopic field. Approximating the distribution, transport map, or field therefore defines three distinct learning problems.

2.2 Characteristic reconstruction and semi-Lagrangian representations

Given a sufficiently smooth field E, let $s \in [0,t]$ denote trajectory time and $(X(s),V(s))$ the trajectory ending at query point (x,v) at time t. Dots denote derivatives with respect to s. Let $f_0(x,v) = f(0,x,v)$ be the prescribed initial distribution and $\Psi^E_{0\leftarrow t}(x,v) = (X(0),V(0))$ the inverse flow to time zero. Constancy along characteristics gives (Glassey, 1996; Sonnendrücker et al., 1999)

$$\dot{X} = V, \quad \dot{V} = E(s,X), \quad (X(t),V(t)) = (x,v), \qquad f(t,x,v) = f_0\big(\Psi^E_{0\leftarrow t}(x,v)\big). \tag{2}$$

PINN approximates $f(t,x,v)$ directly; SL-PINN learns the inverse flow and composes it with f_0 (Monti et al., 2026); NuFI reconstructs the distribution through numerical characteristics (Kirchhart & Wilhelm, 2024); NFCR uses the same reconstruction while learning only time-dependent field coefficients. Classical SL additionally interpolates an evolving distribution grid (Cheng & Knorr, 1976; Sonnendrücker et al., 1999).

2.3 Field representation and the choice of learning target

Equation (2) shows that the electric field over a time interval generates the distribution from its initial condition. The key issue in choosing the electric field as the learning target is whether fine scales in the distribution require an equally complex field representation. Fixing t and omitting the time argument for brevity, let $\widehat{f}_k(v)$, $\widehat{\rho}_k$, and $\widehat{E}_k$ be the spatial Fourier coefficients of distribution, density, and field. Here $k \in \mathbb{Z}\setminus\{0\}$ is the mode index, $\omega_k = 2\pi k/L$ the wavenumber, and i the imaginary unit. Spatial Fourier expansion of Poisson's relation in Equation (1) gives

$$\widehat{E}_k = \frac{\kappa\widehat{\rho}_k}{\mathrm{i}\omega_k}, \qquad \widehat{\rho}_k = \int_{v_{\min}}^{v_{\max}} \widehat{f}_k(v)\,\mathrm{d}v, \qquad \omega_k = 2\pi k/L, \quad k \neq 0. \tag{3}$$

Equation (3) separates two steps. Velocity integration adds contributions across velocities, allowing oscillations with different phases to cancel partially. Poisson's relation then divides each density mode by its wavenumber. Fine velocity oscillations need not therefore produce equally strong field oscillations; at comparable density amplitudes, higher spatial frequencies also yield weaker field modes. NuFI provides numerical examples in which the distribution develops filaments while the field remains relatively smooth (Kirchhart & Wilhelm, 2024).

This motivates learning time-dependent field coefficients and generating the inverse flow and fine distribution scales through characteristic integration. Temporal Fourier features and expanding training intervals follow established ideas (Wang et al., 2021; 2024).

3 Neural Field-Based Characteristic Reconstruction (NFCR)

The only neural state in NFCR is a time-dependent vector of field coefficients. Let $\mathcal{N}_\theta$ be the time network, with all trainable parameters denoted by θ. The field expansion uses the first K nonzero spatial Fourier modes, with sine and cosine coefficients $a_{k,\theta}(t)$ and $b_{k,\theta}(t)$ collected in $\mathbf{e}_\theta(t)$. The field $E_\theta(t,x)$ represented by these coefficients drives backward integration with characteristic step $h > 0$, producing the discrete inverse flow $\Psi^{\theta,h}_{0\leftarrow t}$, reconstructed distribution $f_{\theta,h}$, density ρ_θ, and current J_θ:

$$\mathbf{e}_\theta(t) \longmapsto E_\theta(t,x) \longmapsto \Psi^{\theta,h}_{0\leftarrow t}(x,v) \longmapsto f_{\theta,h}(t,x,v) \longmapsto \big(\rho_\theta(t,x), J_\theta(t,x)\big). \tag{4}$$

Training requires this field to agree with the density and current it generates. We next define the field coefficients, reconstruction, and consistency constraints.

3.1 Parameterizing the field coefficients

Let $T_{\mathrm{norm}} > 0$ be the time normalization scale, $\tau = t/T_{\mathrm{norm}}$ the normalized time, γ a prescribed time-feature map, and $\mathbf{e}_0$ the initial-field coefficients in these K modes. Because position is periodic

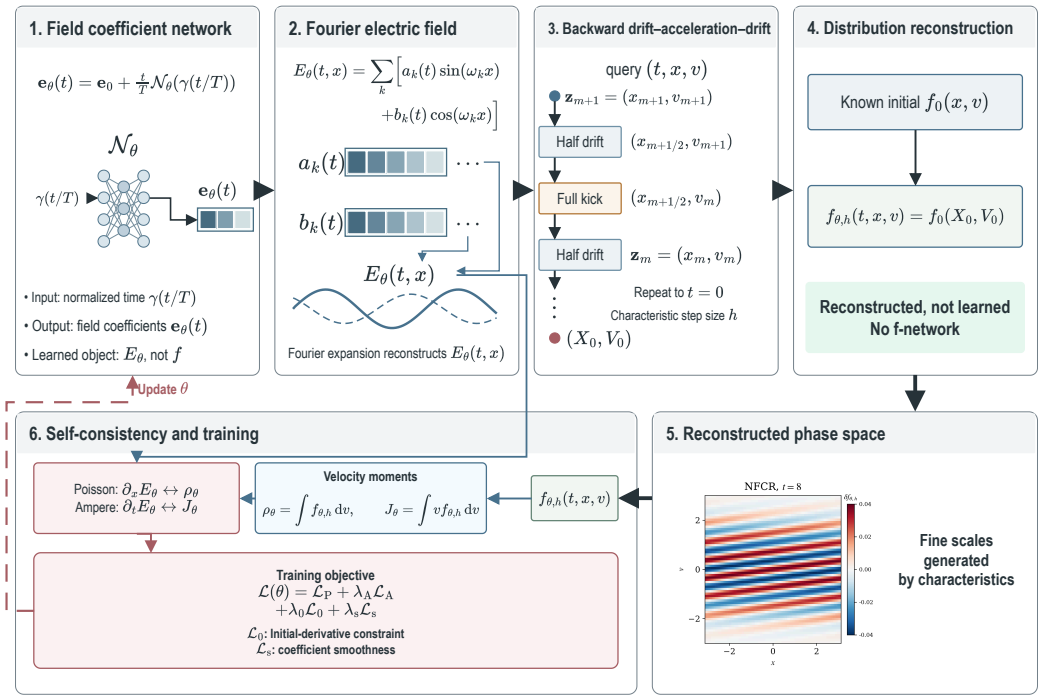

Figure 1: NFCR workflow. A time network outputs field Fourier coefficients; backward drift–acceleration–drift reconstructs the distribution from f_0, whose moments define the self-consistency objectives.

with period L, NFCR represents the field in real sine and cosine Fourier modes with wavenumbers $\omega_k = 2\pi k/L$; this expansion satisfies spatial periodicity by construction and is anchored at the initial field:

$$\begin{aligned} E_\theta(t,x) &= \sum_{k=1}^{K} \left[a_{k,\theta}(t)\sin(\omega_k x) + b_{k,\theta}(t)\cos(\omega_k x)\right], \\ \mathbf{e}_\theta(t) &= \mathbf{e}_0 + \tau\mathcal{N}_\theta(\gamma(\tau)), \qquad \tau = t/T_\mathrm{norm}, \\ \mathbf{e}_\theta(t) &= (a_{1,\theta}(t), b_{1,\theta}(t), \ldots, a_{K,\theta}(t), b_{K,\theta}(t)). \end{aligned} \tag{5}$$

The network returns $2K$ coefficient corrections without position or velocity inputs; training does not repeatedly Fourier-transform distributions or network outputs. The zero-mean condition in Equation (1) intentionally removes the constant term a_0, leaving $k = 1, \ldots, K$. The spatial derivative required by the Poisson residual is obtained analytically from the Fourier basis, while the temporal derivative required by the Ampère residual is computed by automatic differentiation through the time network; neither requires numerical differentiation. The initial coefficients $\mathbf{e}_0$ are computed from the initial density through Poisson's equation and projected onto these K modes. The factor τ enforces $\mathbf{e}_\theta(0) = \mathbf{e}_0$ exactly within this modal space; higher modes in the initial field still incur truncation error.

3.2 Reconstructing the distribution with field-driven backward characteristics

Given E_θ, the continuous characteristic equations in Equation (2) determine the inverse flow and distribution. We approximate this flow with a symmetric drift–acceleration–drift step. Let m index time steps, (x_m, v_m) denote the state at $t_m = t_{m+1} - h$, and $x_{m+1/2}$ the intermediate drift position.

Starting at $(t_{m+1}, x_{m+1}, v_{m+1})$, we compute

$$\begin{aligned} x_{m+1/2} &= \mathrm{wrap}(x_{m+1} - \tfrac{h}{2} v_{m+1}), \\ v_m &= v_{m+1} - h E_\theta(t_{m+1} - \tfrac{h}{2}, x_{m+1/2}), \\ x_m &= \mathrm{wrap}(x_{m+1/2} - \tfrac{h}{2} v_m). \end{aligned} \tag{6}$$

The negative increments move toward earlier times. For a prescribed smooth electric field, the symmetric scheme is second-order accurate in the traced position and velocity over a fixed time interval. The final step is shortened to reach $t = 0$ exactly. For position y, $\mathrm{wrap}(y) = x_{\min} + ((y - x_{\min}) \bmod L)$ returns an out-of-domain position to its representative in one period. It acts only on position and involves no interpolation. With (v_j, w_j) denoting velocity quadrature node j and its weight, the discrete inverse flow defines the distribution, density, and current:

$$\begin{aligned} f_{\theta,h}(t, x, v) &= f_0\left(\Psi^{\theta,h}_{0 \leftarrow t}(x, v)\right), \\ \rho_\theta = \sum_j w_j f_{\theta,h}(t, x, v_j), &\qquad J_\theta = \sum_j w_j v_j f_{\theta,h}(t, x, v_j). \end{aligned} \tag{7}$$

NFCR applies backward characteristic integration directly to each query point and composes it with the known initial condition f_0, so it does not store or repeatedly interpolate an evolving distribution grid. Field evaluation, trajectories, initial-condition composition, and quadrature remain in the automatic-differentiation graph, allowing moment residuals to update the field network. Velocity moments nevertheless use finite quadrature nodes, and reported errors are evaluated on a prescribed grid.

For a fixed smooth field, backward drift updates only position and backward acceleration updates only velocity. Both substeps have unit Jacobian determinant, so their symmetric composition is locally area preserving. Composition with a nonnegative initial distribution preserves nonnegativity, and `wrap` enforces spatial periodicity. These statements do not imply exact mass or energy conservation under velocity truncation, finite quadrature, or finite time steps. Appendix C states the assumptions behind this accuracy result and provides the Jacobian derivation. Section 4 reports mass drift directly.

3.3 Constraining electric-field self-consistency with density and current

The reconstructed density and current must be consistent with the field that generated them. The Poisson residual first tests spatial consistency at each sampled time:

$$R_{\mathrm{P}}(t, x) = \partial_x E_\theta(t, x) - \kappa\big(\rho_\theta(t, x) - \rho_{\mathrm{b}}\big). \tag{8}$$

This condition constrains the spatial variation of E_θ but does not directly constrain the time derivative produced by the network. To obtain a temporal relation, integrating Vlasov in Equation (1) over velocity gives continuity, where J is current and $[Ef]_{v_{\min}}^{v_{\max}}$ is the upper-minus-lower velocity-boundary flux:

$$\partial_t \rho + \partial_x J + [Ef]_{v_{\min}}^{v_{\max}} = 0, \qquad J = \int_{v_{\min}}^{v_{\max}} v f \, \mathrm{d}v. \tag{9}$$

With negligible velocity-boundary flux, differentiate Poisson in Equation (1) in time and substitute continuity, Equation (9). The zero-mean condition $\langle E \rangle_x = 0$ fixes the spatial integration constant $c(t)$ to $\kappa \langle J \rangle_x$, giving

$$\partial_x \partial_t E = -\kappa \partial_x J \implies \partial_t E + \kappa J = c(t) \implies \partial_t E = -\kappa(J - \langle J \rangle_x). \tag{10}$$

We therefore define the Ampère residual

$$R_{\mathrm{A}}(t, x) = \partial_t E_\theta(t, x) + \kappa\big(J_\theta(t, x) - \langle J_\theta(t, \cdot) \rangle_x\big). \tag{11}$$

For exact transport and integration, Equation (10) follows from Poisson, Equation (1), and continuity, Equation (9). With finite steps, quadrature, and collocation, R_{A} supplies a direct temporal constraint during optimization.

Initial field value and initial field derivative are distinct conditions. The field parameterization implies

$$\mathbf{e}_\theta(0) = \mathbf{e}_0, \qquad \partial_t \mathbf{e}_\theta(0^+) = T_{\mathrm{norm}}^{-1} \mathcal{N}_\theta(\gamma(0)). \tag{12}$$

At $t = 0$, $\tau = 0$ eliminates the correction and fixes the coefficients at $\mathbf{e}_0$. The product derivative, however, leaves $\mathcal{N}_\theta(\gamma(0))/T_{\mathrm{norm}}$, which still depends on the network. A fixed starting value therefore permits different initial slopes. Our velocity-symmetric initial conditions have zero current, so Ampère's relation requires a zero initial field derivative. For general data, the actual initial current specifies this derivative. The initial-derivative loss $\mathcal{L}_0$ enforces that relation without reference trajectories.

3.4 Training objective and progressive time intervals

At spacetime collocation points $\mathcal{B}$ and fixed initial spatial points $\mathcal{X}_0$, let $\mathbb{E}$ denote a sample average and $s, s_0 > 0$ residual scales. The three physical objectives are

$$\begin{aligned} \mathcal{L}_{\mathrm{P}} &= \mathbb{E}_{(t,x)\in\mathcal{B}}\left(R_{\mathrm{P}}(t,x)/s\right)^2, \\ \mathcal{L}_{\mathrm{A}} &= \mathbb{E}_{(t,x)\in\mathcal{B}}\left(R_{\mathrm{A}}(t,x)/s\right)^2, \\ \mathcal{L}_0 &= \mathbb{E}_{x\in\mathcal{X}_0}\left(R_{\mathrm{A}}(0,x)/s_0\right)^2. \end{aligned} \tag{13}$$

The current mean uses uniform spatial points at each sampled time. We also use a coefficient-smoothing regularizer $\mathcal{L}_{\mathrm{s}}$, whose discrete definition is given in Appendix A; it is an optimization aid rather than a conservation law. With nonnegative weights $\lambda_{\mathrm{A}}, \lambda_0, \lambda_{\mathrm{s}}$, the total objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{P}} + \lambda_{\mathrm{A}}\mathcal{L}_{\mathrm{A}} + \lambda_0\mathcal{L}_0 + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}}. \tag{14}$$

Residual scales and weights set the relative optimization strength of the terms. Appendix A gives their values, time features, and network architectures. Training uses only the initial distribution and equation parameters; reference solutions enter evaluation only.

Because late-time moments and gradients depend on earlier fields, training expands the same network over endpoints $T_1 < \cdots < T_S$, retaining model and optimizer states while sampling both previous and newly added intervals. The final network returns E_θ; distribution and moment queries use backward characteristics and velocity quadrature. Appendix A lists the windows, sampling, and update counts.

4 Experiments

4.1 Experimental settings and comparison metrics

We use two complementary test cases. Gaussian transport tests distribution reconstruction driven by the learned field coefficients, while Landau damping tests distribution error, electric-field decay, and mass drift as phase mixing creates fine velocity structure.

All methods share a 128×128 evaluation grid within each case; the reported Gaussian and Landau NFCR results use seeds 20260903 and 20260902, respectively. In both cases, SL uses a 128×128 phase-space grid, linear interpolation, Strang splitting, and step 0.05; the particle-in-cell method (PIC) (Hockney & Eastwood, 1988) uses 128 spatial nodes, 2^{18} particles, and step 0.01. The three Landau neural methods share $t_{\max} = 8$, 3,600 updates, and the same seed. Appendix A gives the hardware and full configurations.

For a quantity q on the common evaluation grid, let q_{ref} be its numerical reference, N_q its number of sampled values, i a node index, and $\|\cdot\|_2$ the discrete Euclidean norm. We denote absolute RMS by error_q, relative error by $\mathrm{error}_q^{\mathrm{rel}}$, and field strength by E_{rms}, with N_x spatial nodes:

$$\begin{aligned} \mathrm{error}_q &= \sqrt{\frac{1}{N_q}\sum_i (q_i - q_{\mathrm{ref},i})^2}, \qquad \mathrm{error}_q^{\mathrm{rel}} = \frac{\|q - q_{\mathrm{ref}}\|_2}{\|q_{\mathrm{ref}}\|_2}, \\ E_{\mathrm{rms}} &= \sqrt{\frac{1}{N_x}\sum_i E_i^2}. \end{aligned} \tag{15}$$

The Landau damping experiment also evaluates $\delta f = f - f_{\mathrm{M}}$, where f_{M} is the Maxwellian in Equation (17). The first two columns of Table 2 report relative errors, and the last reports absolute electric-field RMS error. Prediction and reference subtract the same f_{M}, so δf and f have the same

absolute error; the relative perturbation error exposes deviations against the smaller perturbation. All fields are recomputed from each distribution on the common evaluation grid using the same Poisson solver.

4.2 Gaussian transport results

This case tests localized transport under weak coupling. We use $x \in [0, 128)$, $v \in [-6, 6]$, final time $T = 1$, $\kappa = 0.1$, and $\rho_{\mathrm{b}} = 1/128$. With spatial and velocity widths σ_x, σ_v and shortest periodic displacement d_{128}, the initial distribution is

$$f_0(x, v) = \frac{1}{2\pi\sigma_x\sigma_v} \exp\left[-\frac{d_{128}(x - 64)^2}{2\sigma_x^2} - \frac{v^2}{2\sigma_v^2}\right], \qquad \sigma_x = 4, \quad \sigma_v = 0.8. \tag{16}$$

NFCR, SL-PINN, and PINN each receive 2,000 updates; full configurations are in Appendix A. The reference uses fifth-order linear upwind reconstruction (UPW5), following the corresponding HyPar scheme (Ghosh, 2026); it is computed on a 1024×1024 grid with step 0.00125 and resampled to the common evaluation grid. Appendix B specifies reconstruction and time integration.

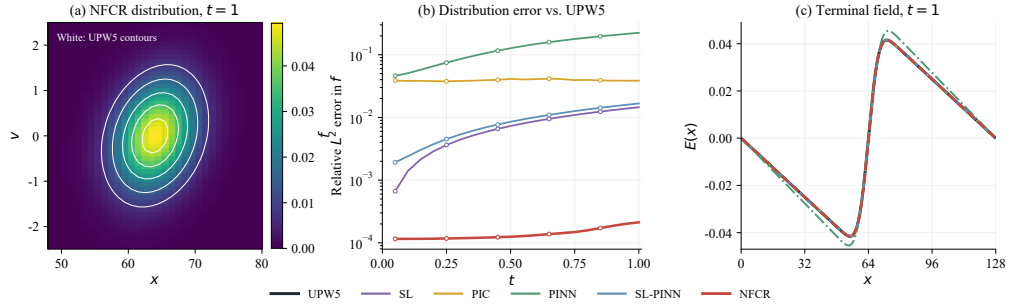

Figure 2: Gaussian common-grid comparison. Left: NFCR at $t = 1$ with UPW5 contours; middle: full-domain distribution errors against UPW5; right: terminal electric fields. Fields use the same Poisson solve, and errors start at $t = 0.05$.

Table 1: Gaussian relative errors at $t = 1$ on the common 128×128 grid against resampled high-resolution UPW5. Values are ratios; density and field are computed from each distribution.

Method	$\mathrm{error}_f^{\mathrm{rel}}$	$\mathrm{error}_\rho^{\mathrm{rel}}$	$\mathrm{error}_E^{\mathrm{rel}}$
SL	0.01455	0.01379	0.004423
PIC	0.03853	0.006279	0.0009082
SL-PINN	0.01670	0.002396	0.001202
PINN	0.2241	0.1143	0.09871
NFCR	0.0002124	9.410×10^{-5}	2.927×10^{-5}

Figure 2 shows that NFCR preserves the terminal peak location and UPW5 contours and has lower full-domain distribution error than SL-PINN and PINN. Its terminal relative errors in distribution, density, and field are 0.0002124, 9.410×10^{-5}, and 2.927×10^{-5} (Table 1); the SL-PINN and PINN distribution errors are approximately 79 and 1055 times larger. Allocating network capacity to field coefficients while numerical characteristics generate the distribution thus reduces the errors observed when fitting a moving distribution or a full inverse flow.

4.3 Landau damping results

This case tests reconstruction of progressively finer velocity structure driven by eight field coefficients, while evaluating electric-field decay and mass drift. We use $x \in [-\pi, \pi)$, $v \in [-6, 6]$, and $\rho_{\mathrm{b}} = \kappa = 1$, with

$$f_0(x, v) = (1 + 0.1 \cos x) f_{\mathrm{M}}(v), \qquad f_{\mathrm{M}}(v) = (2\pi)^{-1/2} \exp(-v^2/2). \tag{17}$$

NFCR trains for 3,600 updates across four windows; full configurations are in Appendix A. UPW5–RK4 combines fifth-order upwind reconstruction with classical fourth-order Runge–Kutta time integration; the reference uses a 1024×1024 grid with step 0.001 and is resampled to the common 128×128 grid.

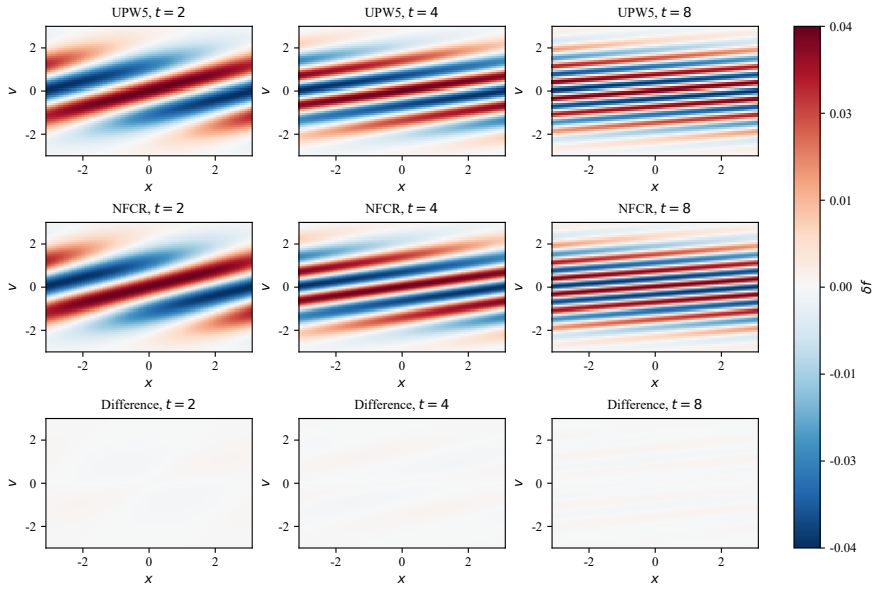

Figure 3: Landau perturbations at $t = 2, 4, 8$: UPW5, NFCR, and their difference from top to bottom. Rows share one symmetric color scale; $|v| \leq 3$ is displayed and norms use the full domain.

Figure 3 shows NFCR recovering progressively finer phase-space bands, while its slices follow UPW5 at all three times with smaller deviations than PINN (Figure 4). At $t = 8$, NFCR's relative distribution error is 0.001952; PINN and SL-PINN are approximately 45 and 77 times larger (Table 2). The network outputs only eight field coefficients while characteristic transport generates the fine velocity scales. PIC error additionally contains particle-sampling and histogram-binning effects.

Table 2: Landau errors against UPW5 at $t = 8$: relative f and δf errors, followed by absolute field RMS error. PIC uses an unsmoothed histogram; all fields use the same Poisson solve.

Method	$\text{error}_f^{\text{rel}}$	$\text{error}_{\delta f}^{\text{rel}}$	error_E
SL	0.006986	0.08226	5.906×10^{-5}
PIC	0.1237	1.457	0.0002327
PINN	0.08836	1.040	0.004424
SL-PINN	0.1499	1.765	0.01713
NFCR	0.001952	0.02298	2.687×10^{-6}

NFCR follows the damping trend. Its terminal absolute RMS field error, 2.687×10^{-6}, is smaller than 5.906×10^{-5}, 0.004424, and 0.01713 for SL, PINN, and SL-PINN, respectively; Figure 5 also shows how the error changes as the reference field decays.

Across four snapshots, NFCR's maximum relative mass drift is 7.651×10^{-9}, compared with 0.1129 for SL-PINN; SL and UPW5 are smaller. This observation is consistent with composition from the initial condition and locally area-preserving transport, but does not establish grid-independent conservation. Appendices D and E report seed stability and modal ablations.

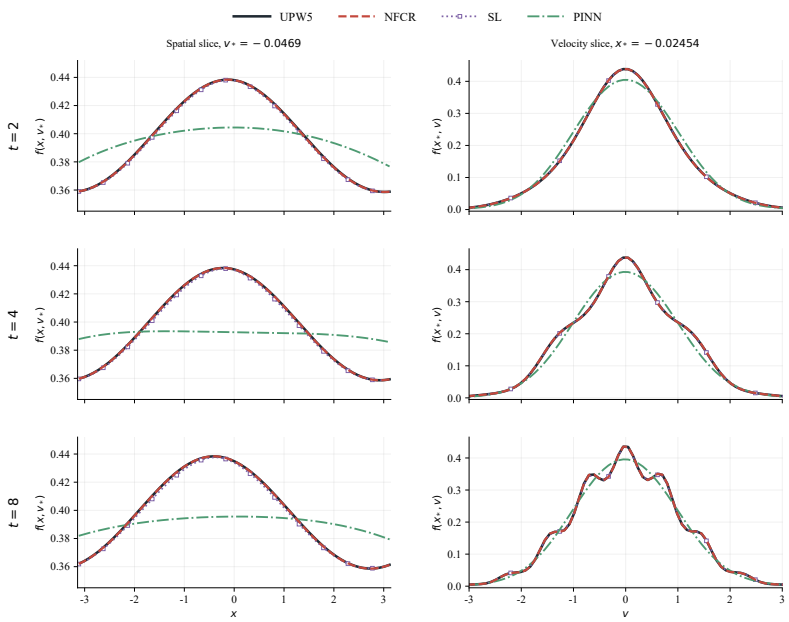

Figure 4: Landau slices at $t = 2, 4, 8$; open squares distinguish SL curves.

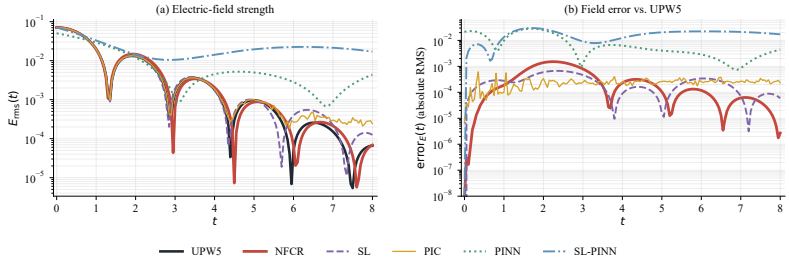

Figure 5: Landau field strength (left) and absolute RMS error against UPW5 (right), computed uniformly from each distribution.

5 Conclusion

NFCR learns time-varying electric-field Fourier coefficients and reconstructs the Vlasov–Poisson distribution from prescribed initial data through differentiable backward characteristics. Its symmetric integrator locally preserves phase-space area, while composition with nonnegative initial data preserves nonnegativity. Poisson and Ampère constraints built from reconstructed density and current keep the field consistent with characteristic transport. The network thus focuses on the comparatively smooth field while transport generates spatial motion and fine velocity structure without an evolving phase-space grid. In the tested Gaussian and Landau configurations, NFCR achieves the lowest terminal distribution errors, follows the reference field, and keeps Landau mass drift at the 10^{-8} scale. Future work will study adaptive representations, nonlinear regimes, and parameterized models across initial conditions.

AI USE STATEMENT

In this work, generative AI tools assisted with method development and refinement, mathematical derivations, experimental design, implementation of NFCR and comparison methods, interpretation of numerical results, and Chinese–English translation. These tools also supported literature search and organization, manuscript drafting and revision, and preparation of evaluation code and figures. All reported numerical results were obtained from numerical experiments. The authors reviewed all AI-assisted work, checked the mathematical derivations and their assumptions, verified consistency between the implementations and their descriptions, and checked the reported metrics, figures, tables, and baseline configurations. The authors also consulted the original literature to verify citation support and the positioning of the contributions, and checked the technical consistency of the Chinese and English manuscripts. The authors take full responsibility for the submitted text, mathematical statements, code, figures, tables, references, and experimental conclusions.

REPRODUCIBILITY STATEMENT

Section 3 specifies the NFCR field representation, backward characteristic integration, self-consistency constraints, and training procedure. Appendix A reports the implementation environment, random seeds, model and baseline configurations, sampling strategy, and common evaluation protocol; Appendix B documents the UPW5–RK4 reference solver and its discretization resolutions; Appendix C states the assumptions behind the structural properties; and Appendices D and E report repeatability across random seeds and ablations over spatial Fourier modes. Together, these sections specify the numerical settings and evaluation procedures used for the results in the main paper and appendix. An anonymized supplementary package containing the implementation, experiment configurations, processed results, and scripts used to reproduce the reported tables and figures is provided with the submission.

REFERENCES

C. Z. Cheng and Georg Knorr. The integration of the Vlasov equation in configuration space. *Journal of Computational Physics*, 22(3):330–351, 1976. doi: 10.1016/0021-9991(76)90053-X.

Debojyoti Ghosh. HyPar: Hyperbolic–parabolic partial differential equations solver. Software and documentation, 2026. URL `https://github.com/debog/hypar`. Fifth-order upwind reconstruction, revision 7b03b9803f67; accessed September 11, 2026.

Robert T. Glassey. *The Cauchy Problem in Kinetic Theory*. Society for Industrial and Applied Mathematics, Philadelphia, 1996.

Robert W. Hockney and James W. Eastwood. *Computer Simulation Using Particles*. Adam Hilger, Bristol, 1988. doi: 10.1887/0852743920.

Matthias Kirchhart and R. Paul Wilhelm. The numerical flow iteration for the Vlasov–Poisson equation. *SIAM Journal on Scientific Computing*, 46(3):A1972–A1997, 2024. doi: 10.1137/23M154710X.

Edoardo Monti, Kateryna Morozovska, Aurora Poggi, and Khemraj Shukla. Semi-lagrangian physics-informed neural networks (SL-PINNs) for solving hyperbolic partial differential equations (PDEs). In *AI&PDE Workshop at ICLR 2026*, 2026. URL `https://openreview.net/forum?id=LYU0dchvlN`.

Clément Mouhot and Cédric Villani. On Landau damping. *Acta Mathematica*, 207(1):29–201, 2011. doi: 10.1007/s11511-011-0068-9.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Eric Sonnendrücker, Jean Roche, Pierre Bertrand, and Alain Ghizzo. The semi-lagrangian method for the numerical resolution of the Vlasov equation. *Journal of Computational Physics*, 149:201–220, 1999. doi: 10.1006/jcph.1998.6148.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021. doi: 10.1016/j.cma.2021.113938.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421: 116813, 2024. doi: 10.1016/j.cma.2024.116813.

Baiyi Zhang, Guobiao Cai, Huiyan Weng, Weizong Wang, Lihui Liu, and Bijiao He. Physics-informed neural networks for solving forward and inverse Vlasov–Poisson equation via fully kinetic simulation. *Machine Learning: Science and Technology*, 4(4):045015, 2023. doi: 10.1088/2632-2153/ad03d5.

Appendices A–C document the experimental settings, the UPW5–RK4 reference, and the structural properties; Appendices D and E report seed repeatability and Fourier-mode sensitivity.

A Implementation, baselines, and evaluation conventions

A.1 NFCR configurations

Table 3: NFCR training settings. N_v^q is the number of training quadrature nodes, distinct from the evaluation grid.

Setting	Gaussian	Landau
Training interval	$[0, 1]$	$[0, 8]$
T_{norm}	1	8
Spatial modes K	20	4
Hidden layers / width	3 / 64	4 / 128
Time-input dimension	9	3
Time slices / spatial points per slice	8 / 64	8 / 64
N_v^q / characteristic step	96 / 0.01	128 / 0.1
Adam learning rate	3×10^{-4}	5×10^{-4}
Total updates	2,000	3,600
Residual scales $s = s_0$	0.01	1
$\lambda_{\mathrm{A}}, \lambda_0, \lambda_{\mathrm{s}}$	$0.1, 1, 10^{-4}$	$1, 1, 10^{-4}$
Main-text random seed	20260903	20260902

NFCR is implemented in PyTorch with float32 on an NVIDIA RTX 3060 Laptop GPU. For a stage endpoint T_s, let $u_m = mh_s$ be M_s equally spaced nodes with $h_s = T_s/M_s$. With $\|\cdot\|_2$ denoting the Euclidean norm, coefficient smoothing is

$$\mathcal{L}_{\mathrm{s}} = \frac{1}{2KM_s} \sum_{m=0}^{M_s-1} \left\| \frac{\mathbf{e}_\theta(u_{m+1}) - \mathbf{e}_\theta(u_m)}{h_s} \right\|_2^2, \qquad u_m = mh_s, \quad h_s = T_s/M_s. \tag{18}$$

This term is an optimization regularizer rather than a conservation law.

Gaussian uses $\gamma(\tau) = (\tau, \{\sin(2\pi m\tau), \cos(2\pi m\tau)\}_{m=1}^4)$; Landau uses $\gamma(\tau) = (\tau, \sin \pi\tau, \cos \pi\tau)$. Hidden activations are tanh, and the final layer's weights and biases start at zero. The initial Ampère loss uses 64 fixed spatial points and the same velocity quadrature as the main residual. Gaussian time slices are sampled randomly from the discrete grid aligned with the characteristic step. Landau uses deterministic representatives of time strata and stage budgets of 800, 800, 800, and 1,200 updates, carrying the model and optimizer state across stages.

The initial Landau field is $0.1 \sin x$, which lies in the spatial basis used by the model. The Gaussian initial coefficients are obtained by solving Poisson's equation from the initial density on 4,096 spatial points and then truncating to the selected modes. The initial-density calculation includes the fraction of the velocity Gaussian contained in the numerical interval $[v_{\min}, v_{\max}]$, keeping the initial field consistent with the truncated velocity domain. In both cases, the normalization scale equals the training endpoint, so $\tau = t/T_{\mathrm{norm}}$ spans $[0, 1]$ over the training interval.

A.2 BASELINE CONFIGURATIONS

The classical semi-Lagrangian baseline uses a 128×128 grid, Strang splitting, linear interpolation, and step 0.05 in both cases. Both PIC baselines use 128 spatial nodes, 2^{18} particles, step 0.01, quiet initialization, cloud-in-cell (CIC) deposition, FFT Poisson, and leapfrog integration. The PIC distribution is an unsmoothed histogram on the common 128×128 cells, so its distribution error includes particle-sampling and binning errors.

Gaussian SL-PINN uses five hidden layers of width 128, normalized inputs, 2,000 updates, 8,192 collocation points per update, eight time slices, a 128×128 field grid, and learning rate 3×10^{-4}. The Jacobian and periodic-boundary losses both have weight 0.01; the learned inverse map is composed with the analytic initial distribution. Gaussian PINN uses the same depth and width, outputs a non-negative distribution through softplus, and trains for 2,000 updates. These configurations describe the evaluated implementations and budgets, not the best attainable performance of either method.

Landau SL-PINN uses seven hidden layers of width 64, 20,480 collocation points per update, eight time slices, and a 128×128 field grid. AdamW uses learning rate 10^{-4}; the Jacobian and periodic-boundary losses both have weight 0.1. Seed 20260902 is trained on $[0, 8]$ for 3,600 updates. Landau PINN uses a seven-hidden-layer, width-64 SiLU network with a softplus output and the same collocation count, time slices, field grid, seed, and update count. Its Vlasov, initial-condition, and periodic-boundary loss weights are $1, 1, 0.1$, with 2,048 initial points and 1,024 boundary points; AdamW uses learning rate 10^{-4} and weight decay 0.01. Neither neural baseline uses reference trajectories for training.

A.3 COMMON EVALUATION CONVENTIONS AND TWO ELECTRIC FIELDS

Gaussian and Landau both use a common 128×128 cell-centered evaluation grid. Gaussian saves 21 evaluation times. Landau saves distribution snapshots at $t = 0, 2, 4, 8$ and additionally uses dense output with spacing 0.05 for Figure 5. Temporal interpolation is not used to synthesize terminal distributions.

NFCR uses two related fields with different roles. The field $E_\theta(t, x)$ is represented by the Fourier coefficients output by the time network; it drives characteristics and enters the Poisson and Ampère residuals. The field $E_\rho(t, x)$ is instead obtained by computing density from the reconstructed distribution and solving Poisson with all spatial FFT modes. To give every method the same comparison convention, all main-text field curves and metrics use the latter procedure: each method first supplies its distribution, and the same float64 Poisson solver then computes its field. The former field is reported separately only in the modal study in Appendix E, where the capacity of the spatial basis must be distinguished from the error of the complete distribution reconstruction.

Matching the evaluation grid does not equate training windows, particle counts, or numerical budgets. The main-text seeds were specified before evaluation; the multi-seed batch is not used to select the best main-text run. Appendix B gives the high-resolution references, time integration, and boundary treatment.

B UPW5–RK4 REFERENCE SOLVER

UPW5 denotes fifth-order linear upwind reconstruction with the fixed coefficients used by HyPar (Ghosh, 2026). Along either uniform coordinate direction, let u_i be the distribution value at node i, and let $u^-_{i+1/2}$ and $u^+_{i+1/2}$ be the left- and right-biased reconstructions used in the numerical flux:

$$\begin{aligned} u^-_{i+1/2} &= \frac{u_{i-2}}{30} - \frac{13u_{i-1}}{60} + \frac{47u_i}{60} + \frac{27u_{i+1}}{60} - \frac{u_{i+2}}{20}, \\ u^+_{i+1/2} &= -\frac{u_{i-1}}{20} + \frac{27u_i}{60} + \frac{47u_{i+1}}{60} - \frac{13u_{i+2}}{60} + \frac{u_{i+3}}{30}. \end{aligned} \tag{19}$$

Positive transport speed selects the left-biased flux, and negative speed selects the right-biased flux. The implementation checks neighboring transport coefficients: a uniformly signed neighborhood uses upwind selection, whereas a neighborhood crossing zero uses a local Lax–Friedrichs flux with the local maximum absolute speed. The transport speed is v along position and E along velocity. The scheme uses fixed linear coefficients, without WENO nonlinear weights.

Flux differences in both directions form the conservative semi-discrete right-hand side, which is advanced by classical four-stage, fourth-order Runge–Kutta. Each stage recomputes density from the current distribution and solves Poisson with the complete spatial FFT spectrum, setting the zero field mode to zero. Position is periodic; velocity uses three zero-valued ghost cells. Negative distribution values are not clipped. Computation uses float64.

The Gaussian reference uses a 1024×1024 grid and step 0.00125; the Landau reference uses a 1024×1024 grid and step 0.001. Landau requires the smaller step because its shorter spatial period imposes a stricter explicit CFL condition at this resolution. Both references are resampled to the common 128×128 grid by the same periodic bilinear cell-center rule, and the common-grid array at $t = 0$ is reset from the analytic initial condition. These are numerical references for the stated discretizations and retain spatial, temporal, velocity-truncation, and resampling errors.

C Structural properties of characteristic reconstruction

For a fixed smooth field and away from the coordinate seam of the periodic representation, the backward drift and acceleration Jacobians are

$$\mathbf{J}_{\mathrm{D}} = \begin{pmatrix} 1 & -h/2 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{J}_{\mathrm{A}} = \begin{pmatrix} 1 & 0 \\ -h\partial_x E_\theta & 1 \end{pmatrix}, \qquad \det(\mathbf{J}_{\mathrm{D}}\mathbf{J}_{\mathrm{A}}\mathbf{J}_{\mathrm{D}}) = 1. \tag{20}$$

Each factor has determinant one, so their composition preserves area locally. Spatial `wrap` identifies equivalent coordinates within a period and introduces no additional interpolation. For a smooth time-dependent field, midpoint field evaluation and the symmetric drift–acceleration–drift composition have local error $O(h^3)$ per step and, under standard smoothness and stability assumptions, $O(h^2)$ error in the backtraced position and velocity over a fixed finite interval. This order concerns integration under a prescribed field, not convergence of the learned field to a self-consistent solution.

If $f_0 \geq 0$, reconstruction $f_{\theta,h} = f_0 \circ \Psi^{\theta,h}_{0\leftarrow t}$ remains nonnegative. If the complete integration domain were invariant under the map and quadrature were exact, local area preservation could support integral conservation. A truncated velocity interval need not be invariant, however, and finite quadrature and evaluation grids introduce further errors. Local area preservation therefore does not imply exact mass or energy conservation of the reported arrays. The Ampère derivation also assumes negligible velocity-boundary flux; the main text retains the finite-domain boundary term before invoking this assumption.

D Repeatability across random seeds

Across the five Gaussian runs, terminal relative distribution error is $5.343 \times 10^{-5} \pm 6.652 \times 10^{-5}$, absolute RMS field error is $4.394 \times 10^{-8} \pm 5.477 \times 10^{-8}$, and maximum relative mass drift is $8.346 \times 10^{-8} \pm 5.778 \times 10^{-9}$. The corresponding Landau values are 0.001722 ± 0.0002660, $8.793 \times 10^{-5} \pm 0.0001583$, and $1.397 \times 10^{-8} \pm 1.641 \times 10^{-9}$. Here $\pm$ denotes the sample standard deviation, not a confidence interval.

Mean mass drift in both cases remains between 10^{-8} and 10^{-7}, with sample standard deviations well below the corresponding means. Low mass drift is therefore repeatable across the tested initializations and sampling realizations. Distribution and field errors vary more, leaving room to improve optimization consistency. This batch is not used to select the reported main-text runs.

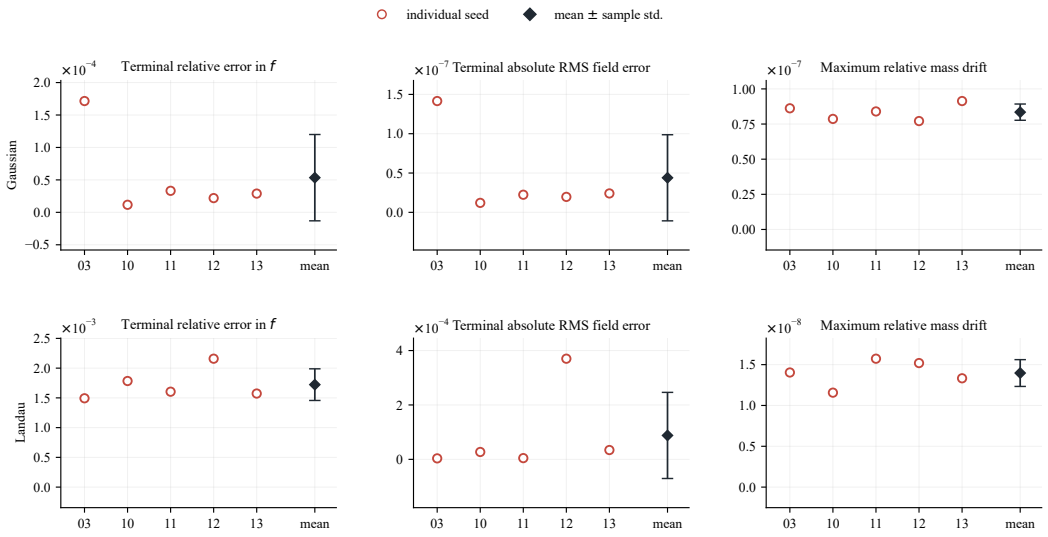

Figure 6: Five-seed NFCR results for Gaussian (top) and Landau (bottom), using seeds 20260903 and 20260910–20260913. Circles show individual runs; diamonds and error bars show the mean and sample standard deviation. Columns report terminal relative distribution error, terminal absolute RMS field error, and maximum relative mass drift.

E FOURIER-MODE SENSITIVITY OF THE COMPACT REPRESENTATION

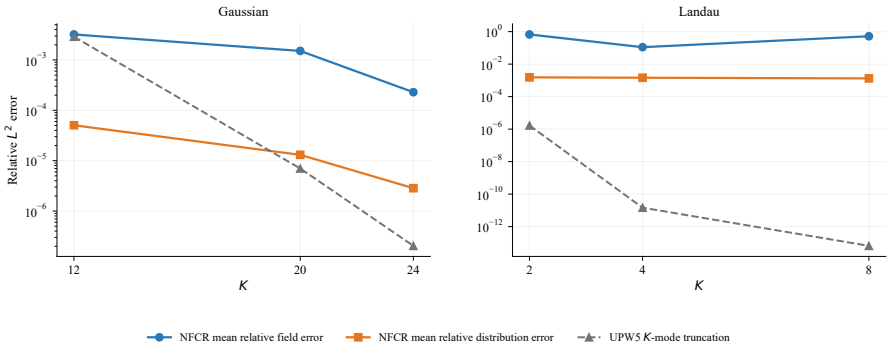

Figure 7: Effect of spatial mode count K. Blue: mean relative network-field error; orange: mean relative distribution error; gray dashed: UPW5 field truncation error. Gaussian uses $K = 12, 20, 24$ and Landau uses $K = 2, 4, 8$; results use one seed.

The main experiments use $K = 20$ for Gaussian and $K = 4$ for Landau. In Gaussian, increasing K from 12 to 24 reduces the mean field and distribution errors from $(0.003210, 5.052 \times 10^{-5})$ to $(0.0002293, 2.847 \times 10^{-6})$, while the UPW5 K-mode truncation error falls from 0.002896 to 2.040×10^{-7}. The larger mode counts therefore improve the result, but the additional gain beyond $K = 20$ is small relative to the added representation size. We use $K = 20$ because it already provides sufficient accuracy for the main Gaussian experiment.

In Landau, the UPW5 K-mode truncation error falls from 1.679×10^{-6} at $K = 2$ to 6.539×10^{-14} at $K = 8$. The mean error of the field obtained from the network Fourier coefficients is lowest at $K = 4$, 0.1104, whereas the mean distribution error is lowest at $K = 8$, 0.001325, compared with 0.001451 at $K = 4$. The difference between $K = 4$ and $K = 8$ is modest over the reported interval, so four modes provide a more compact field representation with comparable overall behavior and are used in the main experiment. Complete errors also include optimization, characteristic integration, quadrature, and self-consistency, so this is not strict modal convergence.