Non-linear control variate in δf methods using symplectic neural networks

Congrès National d'Analyse Numérique

Victor Fournet (victor.fournet@ipp.mpg.de)

with Martin Campos-Pinto (NMPP), Emmanuel Franck (IRMA), Victor Michel-Dansac (IRMA)

Max Planck Institute for Plasma Physics, Garching

Introduction

Vlasov Poisson system: $f(t,x,v)$ distribution of particles, $\phi(t,x)$ electric potential

$$\partial_t f + v \cdot \nabla_x f - \nabla_x \phi \cdot \nabla_v f = 0, \qquad -\Delta \phi = \int f \, \mathrm{d}v - 1.$$

Forward Flow $\Phi_{0,t}$ defined by characteristic curves

$$\begin{cases} \dfrac{dX}{dt} = V, \quad X(t=0) = x \\ \dfrac{dV}{dt} = -\nabla \phi(X), \quad V(t=0) = v \end{cases}$$

- The density can be expressed in terms of the initial density f_{init} and the flow map $\Phi_{0,t}$ as
$$f(t,x,v) = f_{init}(\Phi_{0,t}^{-1}(x,v).)$$
- If $0 = t_0 < \cdots < t_n$ is a partition of $[0,t]$, and $\Phi_{t_i,t_{i+1}}$ is the flow map from t_i to t_{i+1}, then the density can be expressed as
$$f(t,x,v) = f_{init}(\Phi_{0,t}^{-1}(x,v).), \quad \Phi_{0,t}^{-1} = \Phi_{t_{n-1},t_n}^{-1} \circ \ldots \Phi_{t_0,t_1}^{-1}$$

Family of Flow based numerical methods

- Semi Lagrangian method: Gives approximation of backward flow $\Psi_h \approx \Psi = \Phi^{-1}$:

$$f(t,x,v) \approx f_{init}(\Psi_h(0,t;x,v)).$$

  - Has many advantages but need a 6D phase space grid.
- Particles in Cell (PIC) method: Gives approximation $\Phi_h \approx \Phi$ and push markers:

$$f(t^{n+1},x,v) \approx \sum_{k=1}^{N_p} w_k S_\varepsilon(x-x_k^{n+1})\delta(v-v_k^{n+1}), \quad (x^{n+1},v^{n+1}) = \Phi_h(x^n,v^n).$$

  - Weight of the particles

$$w_k^n = \frac{f_{\text{init}}(x_k^0,v_k^0)}{N_p g(x_k^0,v_k^0)}$$

    .
  - Moments of f,

$$I(a) = \int a(v) f^n(x,v)\,\mathrm{d}v = \frac{1}{N_p}\sum_{k=1}^{N_p} \frac{f_{\text{init}}(x_k^0,v_k^0)}{g(x_k^0,v_k^0)} a(v_k^n) S_\varepsilon(x-x_k^n)$$

  - Error scale as $\approx \frac{\sigma}{\sqrt{N_p}}$

δf PIC with static bulk densities

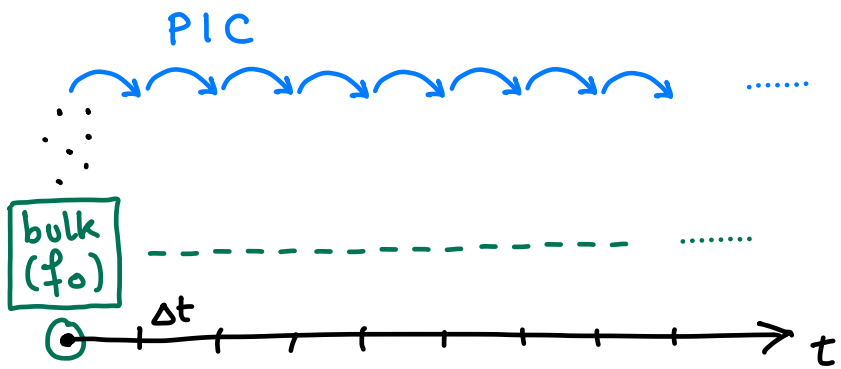
δf ansatz for the solution

$$f^n = f_0 + \delta f^n, \quad \rho^n = \underbrace{\int f_0 \, \mathrm{d}v}_{\text{Analytical}} + \underbrace{\int (\delta f)^n \, \mathrm{d}v}_{\text{Noisy but small}}$$

- f_0: **Fixed** explicit bulk density (control variate) - typically a Maxwellian
- δf^n represented by numerical particles

$$\delta f^n(z) = \sum_{k=1}^{N_p} \delta w_k^n \varphi_\varepsilon(z - z_k^n), \quad \delta w_k^n = \frac{f_{init}(z_k^0) - f_0(z_k^n)}{N_p g(z_k^0)}$$

- If $\|\delta f^n\| \ll \|f_0\|$: denoising effect.



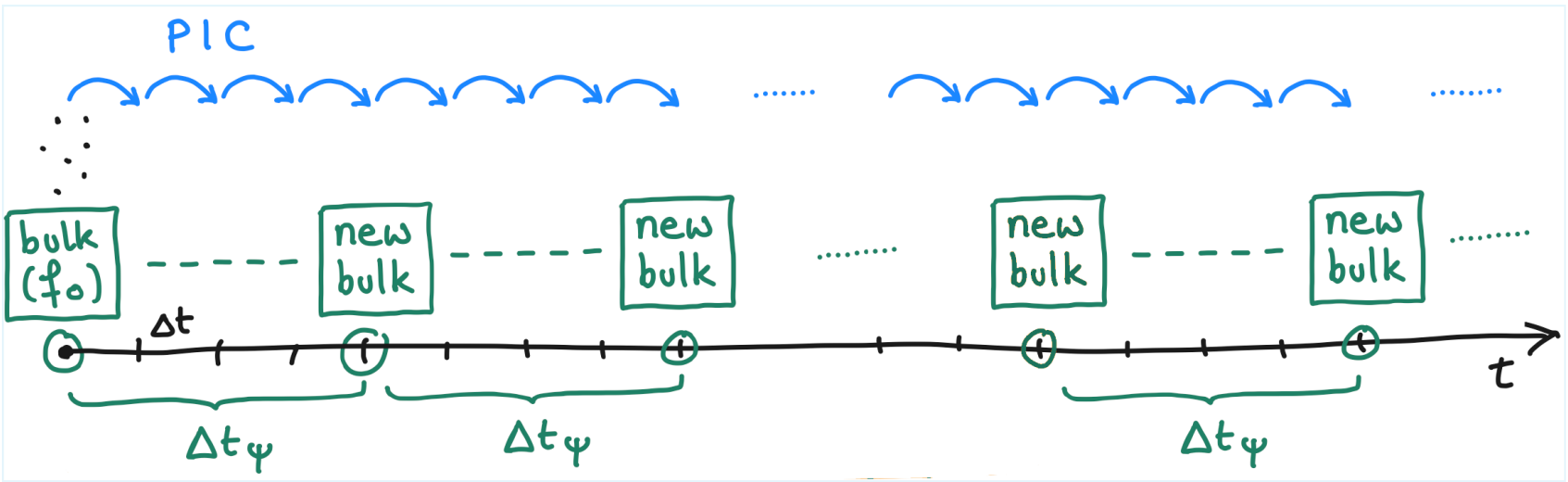
δf PIC with dynamic bulk densities

Alternative: evolve the bulk f_0 (control variate).

$$f^n = f_0^m + \delta f^n, \quad \rho^n = \int f_0^m \, \mathrm{d}v + \int (\delta f)^n \, \mathrm{d}v, \quad m = \left\lfloor \frac{n}{N_\psi} \right\rfloor$$

Two choices

- How to transport of the bulk itself
- How to compute the moments of the bulk



δf PIC with dynamic bulk densities

Alternative: evolve the bulk f_0 (control variate).

$$f^n = f_0^m + \delta f^n, \quad \rho^n = \int f_0^m \, \mathrm{d}v + \int (\delta f)^n \, \mathrm{d}v, \quad m = \left\lfloor \frac{n}{N_\Psi} \right\rfloor$$

- Common strategy: Assume f_0 is a Maxwellian and evolves its moments with fluid equations [1] .
- Alternative: δf-FBL[2]:
  - Transport passive markers (without charge) on a grid.
  - Compute local Backward flow using linear or quadratic Taylor expansion
  - Reconstruct density on a grid.
  - Issue on high dimension.

[1] Gyrokinetic flux-driven simulations in mixed TEM/ITG regime using a delta-f PIC scheme with evolving background M. Murugappan, L. Villard, S. Brunner, G. Di Giannatale, B. F. McMillan, and A. Bottino

[2] A δf PIC method with Forward-Backward Lagrangian reconstructions M. Campos Pinto, M. Pelz, and P.-H. Tournier

δf with neural flow-based bulk densities

- Recently, neural network have shown promising results to avoid the curse of dimensionaly while being good approximator.

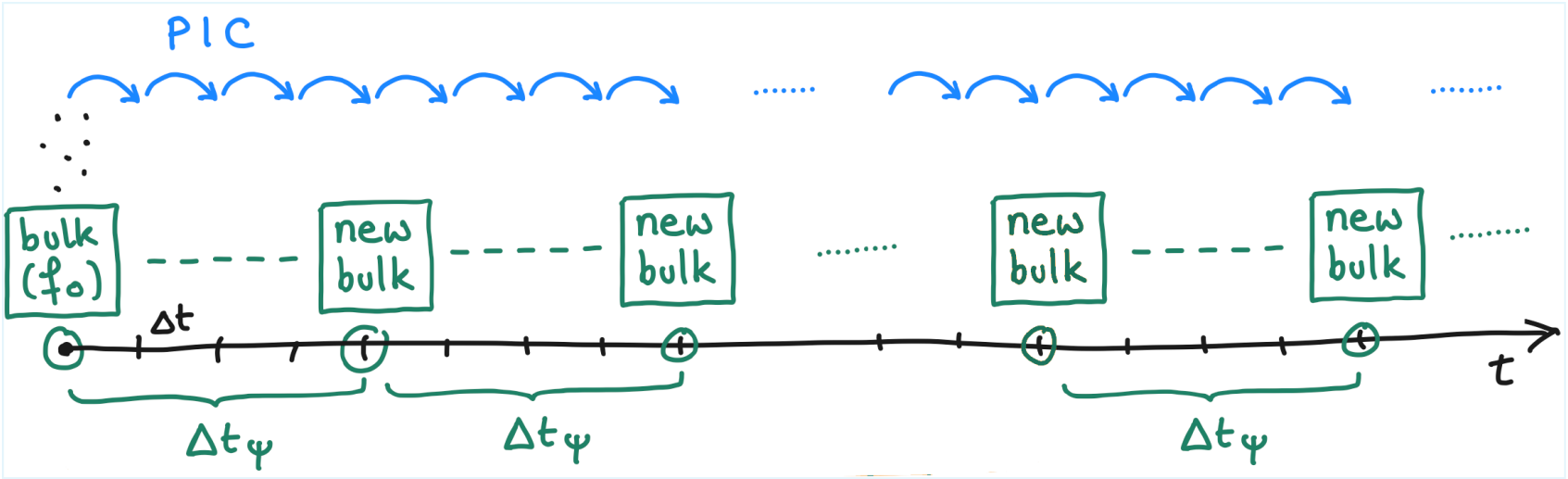
$$f^n = f_0^m + \delta f^n, \quad \rho^n = \int \widetilde{f_0^m}\,\mathrm{d}v + \int \delta f^n\,\mathrm{d}v, \quad m = \left\lfloor \frac{n}{N_\Psi} \right\rfloor$$

$$\delta f^n(z) = \sum_{k=1}^{N_p} \delta w_k^n \varphi_\varepsilon(z - z_k^n), \quad \delta w_k^n = \frac{f_{init}(z_k^0) - \widetilde{f_0^m}(z_k^n)}{N_p g(z_k^0)}$$

- Transport of the density: Neural networks trained on particle trajectory

$$f_0^m(x,v) = f_0\left(\Psi_{\theta^1} \circ \cdots \circ \Psi_{\theta^m}(x,v)\right).$$

- Computation of the moments: Projection on a coarse B-Spline grid.



Symplectic neural networks for the flow

$$f_{\text{up}}(x,v) = \begin{pmatrix} x + \nabla V_{\text{up}}(v) \\ v \end{pmatrix}, \quad f_{\text{down}}(x,v) = \begin{pmatrix} x \\ v + \nabla V_{\text{down}}(x) \end{pmatrix},$$

Lemma: Any symplectic map can be approximated by composition of shear maps.

Definition 1: SympNet

Gradient module are functions of the form

$$\psi_{\text{up}}(x,v) = \begin{pmatrix} x + \widetilde{\sigma}_{K,b}(v) \\ v \end{pmatrix}, \quad \psi_{\text{down}}(x,v) = \begin{pmatrix} x \\ v + \widetilde{\sigma}_{K,b}(x) \end{pmatrix},$$

$$\widehat{\sigma}_{K,b}(x) = K^T \sigma(Kx + b) \approx \nabla V : \mathbb{R}^d \to \mathbb{R}^d, \quad K \in \mathbb{R}^{n \times d}, b \in \mathbb{R}^n,$$

with σ activation function. A **symplectic neural network** is defined as

$$\Psi_\theta(x,v) = \psi^k_{\text{up}} \circ \psi^k_{\text{down}} \circ \cdots \circ \psi^1_{\text{up}} \circ \psi^1_{\text{down}}(x,v).$$

SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems Pengzhan Jin, Zhen Zhang, Aiqing Zhu, Yifa Tang, George Em Karniadakis

Periodic symplectic neural network

$$\widetilde{\sigma}_{K_1,K_2,b_1,b_2}(x) = \begin{pmatrix} -K_1 \mathrm{diag}(\sin(2\pi x/L)) \\ K_2 \mathrm{diag}(\cos(2\pi x/L)) \end{pmatrix} \cdot \sigma \left(\begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix} \begin{pmatrix} \cos(2\pi x/L) \\ \sin(2\pi x/L) \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)$$

with $K_1, K_2 \in \mathbb{R}^{n\times d}$, $b_1, b_2 \in \mathbb{R}^n$. One has $\widetilde{\sigma}_{K_1,K_2,b_1,b_2} \approx \nabla V_{\mathrm{per}} : \mathbb{T} \to \mathbb{R}$

Definition 2: Periodic SympNet

Gradient module are functions of the form

$$\psi_{\mathrm{up}}(x,v) = \begin{pmatrix} x + \widehat{\sigma}_{K,a,b}(v) \mathrm{mod} L \\ v \end{pmatrix}, \quad \widetilde{\psi}_{\mathrm{down}}(x,v) = \begin{pmatrix} x \\ v + \widetilde{\sigma}_{K_1,K_2,b_1,b_2}(x) \end{pmatrix}$$

$$\widehat{\sigma}_{K,a,b}(x) = K^T \sigma(Kx + b) \approx \nabla V : \mathbb{R}^d \to \mathbb{R}^d, \quad K \in \mathbb{R}^{n\times d}, b \in \mathbb{R}^n.$$

A **periodic symplectic neural network** is defined as

$$\Psi_\theta(x,v) = \psi^k_{\mathrm{up}} \circ \widetilde{\psi}^k_{\mathrm{down}} \circ \cdots \circ \psi^1_{\mathrm{up}} \circ \widetilde{\psi}^1_{\mathrm{down}}(x,v).$$

Algorithm: Neural δf-PIC with symplectic flow

$$f^n = f_0^m + \delta f^n, \quad m = \left\lfloor \frac{n}{N_\Psi} \right\rfloor$$

1. *Initial condition*: set $f_0^0 := f_{init}$, compute and store $\int f_0^0 \, \mathrm{d}v$, initialize particles $\{(x_k^0, v_k^0)\}_{k=1}^{N_p}$
2. *Time-stepping*:
   - For each $n < N$: push particles and update weights *(PIC timescale Δt)*

$$(x_k^n, v_k^n) \xrightarrow{\mathrm{VP}} (x_k^{n+1}, v_k^{n+1}), \qquad \delta w_k^{n+1} = \frac{f_{init}(z_k^0) - f_0^m(z_k^{n+1})}{N_p \, g(z_k^0)}$$

   - If $n = mN_\Psi$ Train SympNet and update bulk:

$$\theta^m = \mathrm{argmin}_\theta \frac{1}{N_p} \sum_k \left| \Psi_\theta(x_k^n, v_k^n) - (x_k^{n-N_\Psi}, v_k^{n-N_\Psi}) \right|^2$$

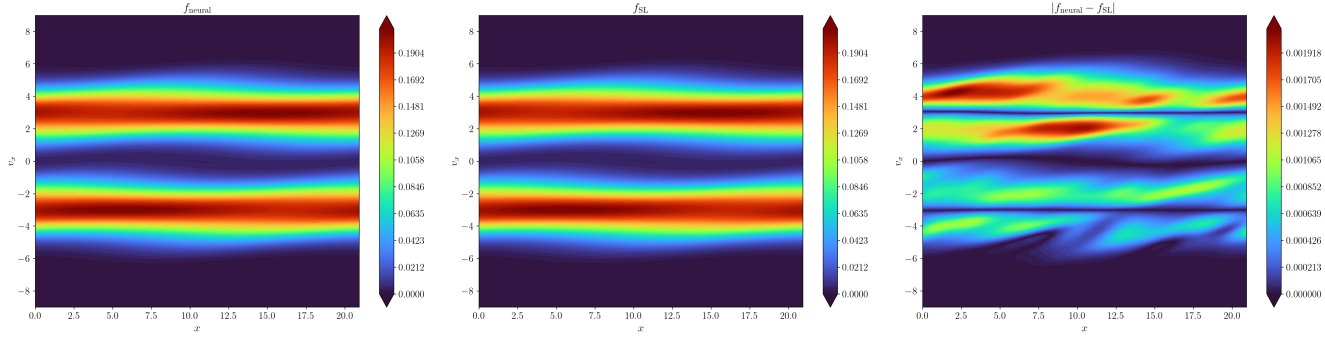
$$f_0^m(x, v) = f_0(\Psi_{\theta^1} \circ \cdots \circ \Psi_{\theta^m}(x, v))$$

Compute and store

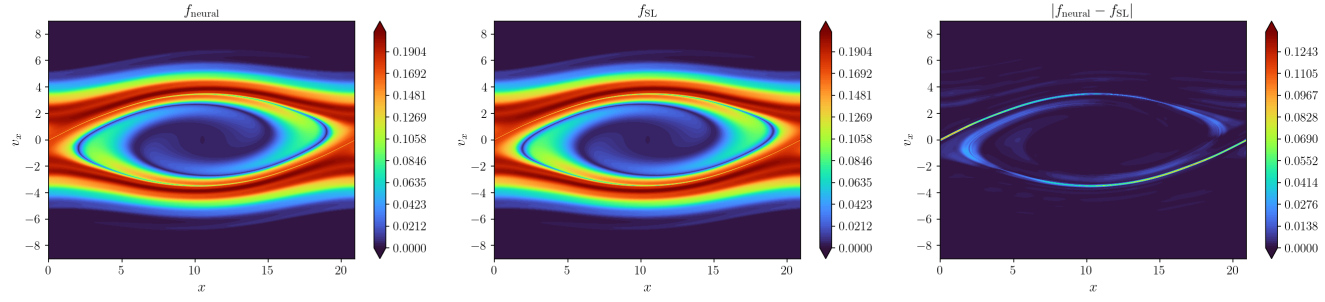
$$\int \widetilde{f_0^m} \, \mathrm{d}v$$

1D1V Two-stream instability: Phase space density

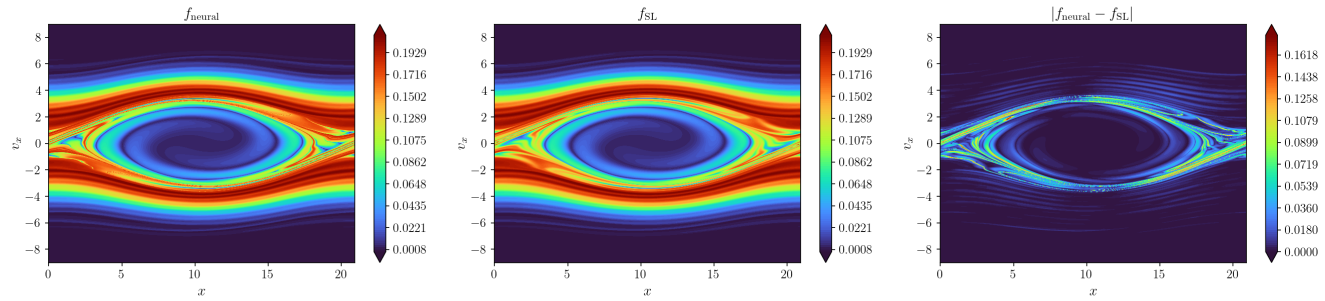
$$f_{init}(x,v) = \frac{1}{2\sqrt{2\pi}}(e^{-(v-3)^2/2} + e^{-(v+3)^2/2})(1 + 0.05\cos(0.5x))$$



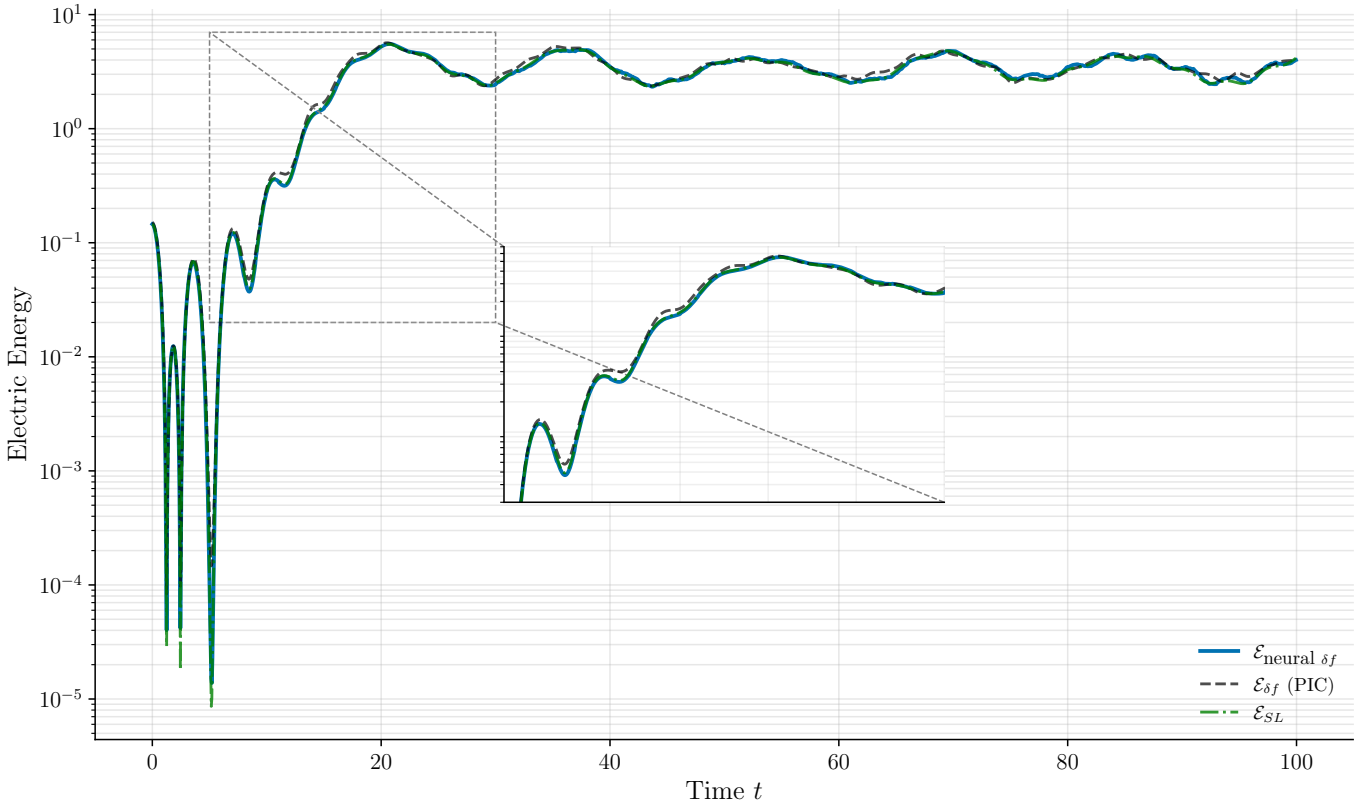
Comparison at $t = 25.05$



Comparison at $t = 50.05$

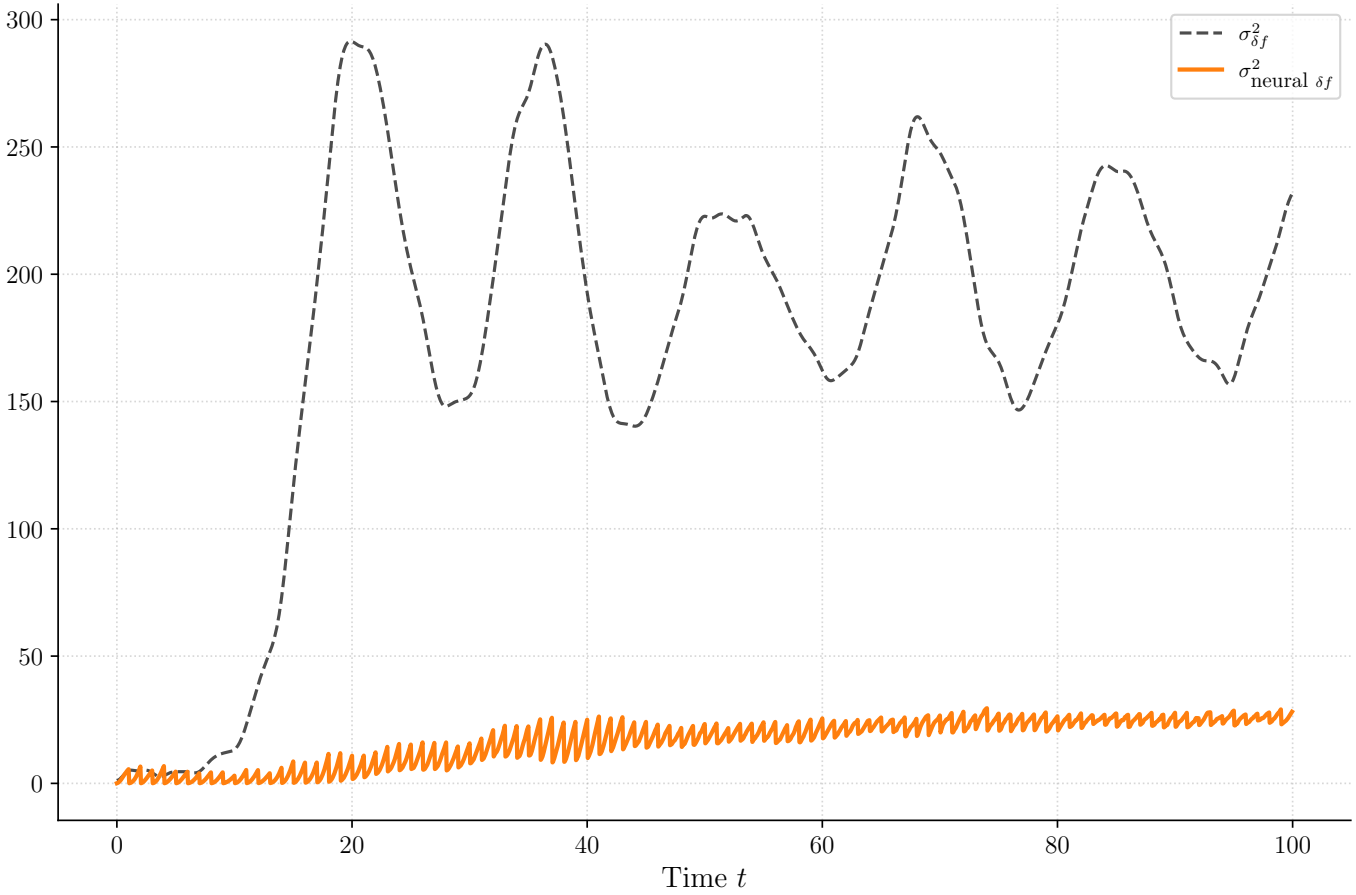


Two-stream instability: Electric energy



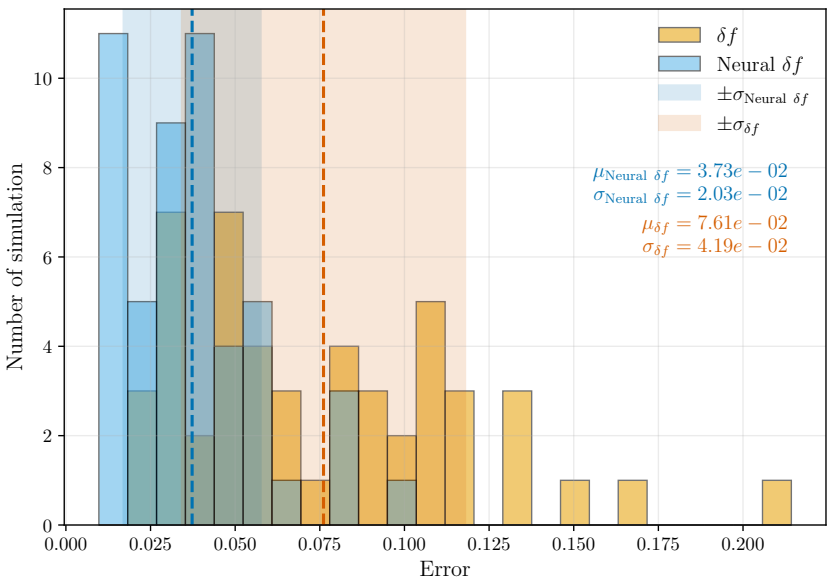
Two-stream instability: Weight Variance

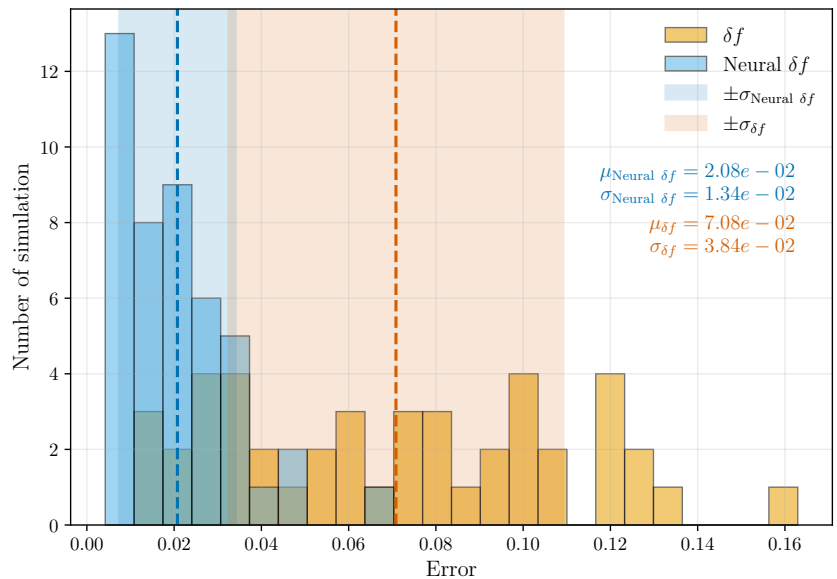
$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$



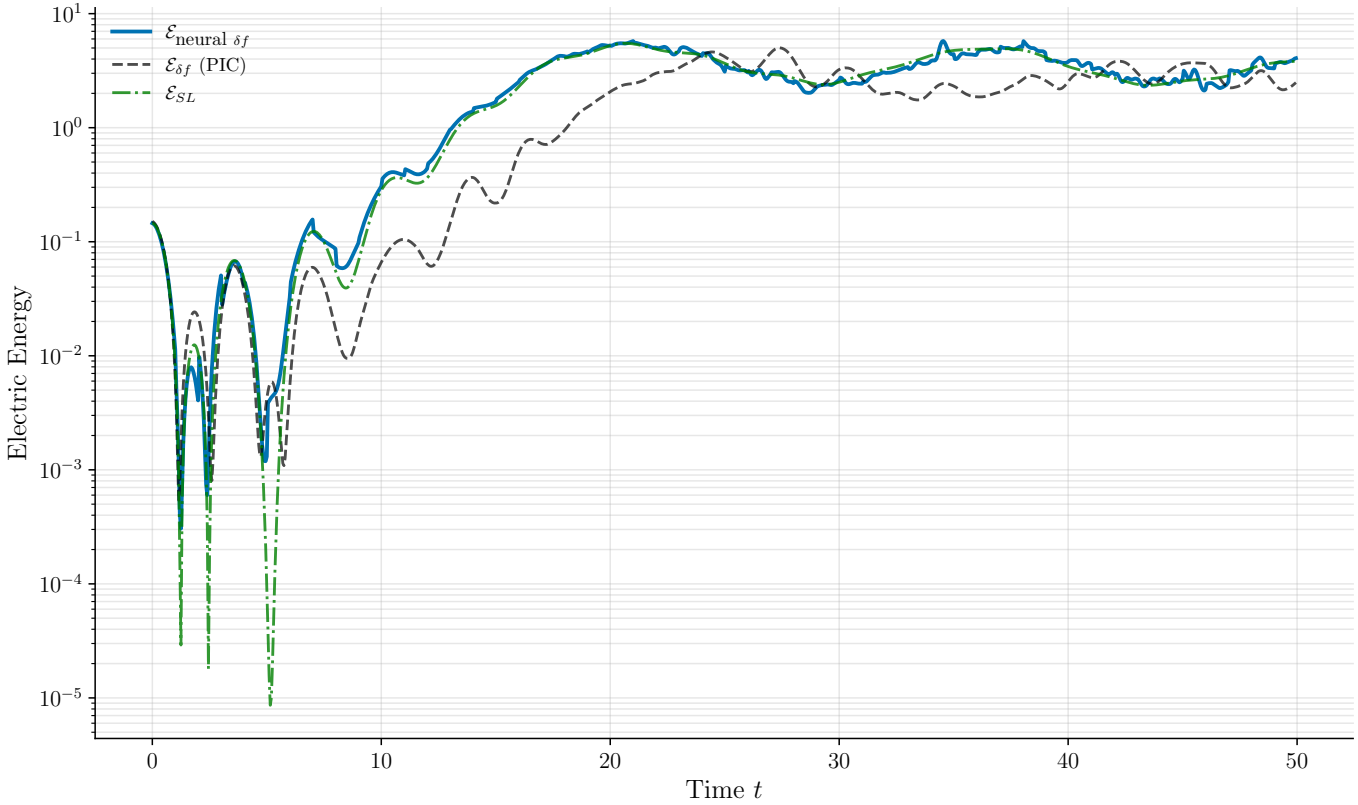
Statistics of error on 50 runs: $\mathcal{E}_{PIC}$ vs $\mathcal{E}_{neural}$

$$\mathcal{E}_{PIC} = \sqrt{\int |E_{PIC} - E_{BSL}|^2 \, \mathrm{d}t} \;,\; \mathcal{E}_{neural} = \sqrt{\int |E_{neural} - E_{BSL}|^2 \, \mathrm{d}t}$$



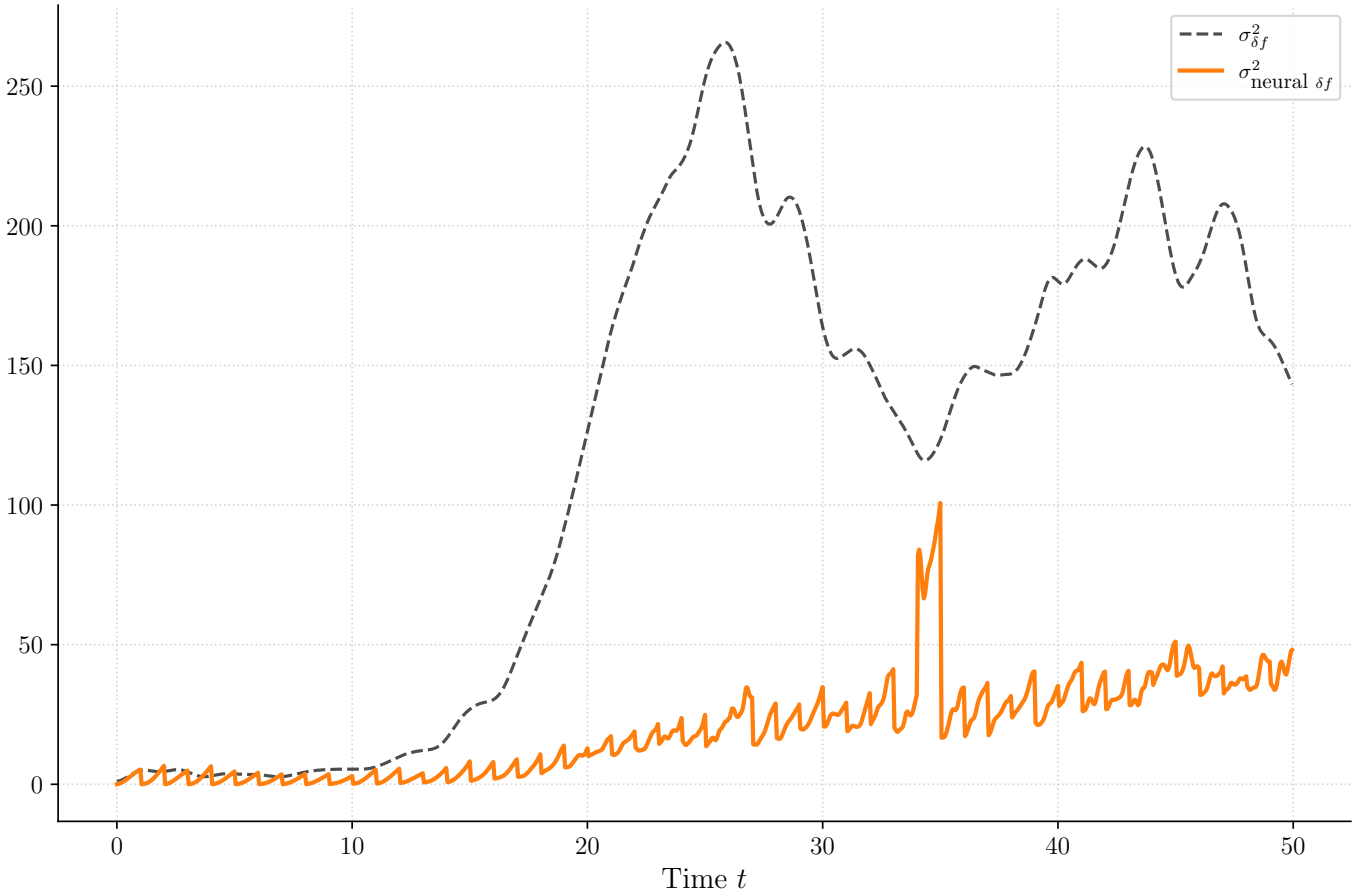


Two-stream instability: Electric energy with $N_p = 1000$

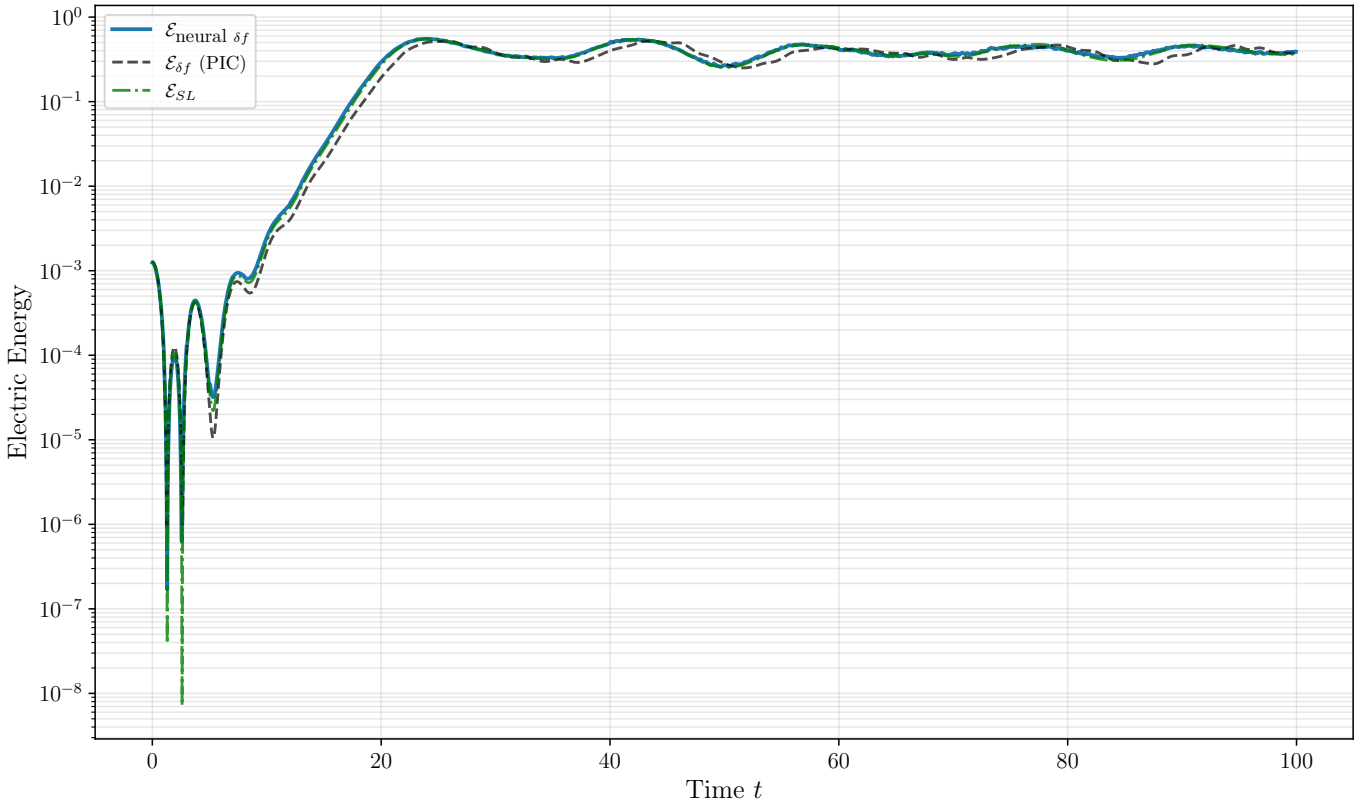


Two-stream instability: Weight Variance

$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$

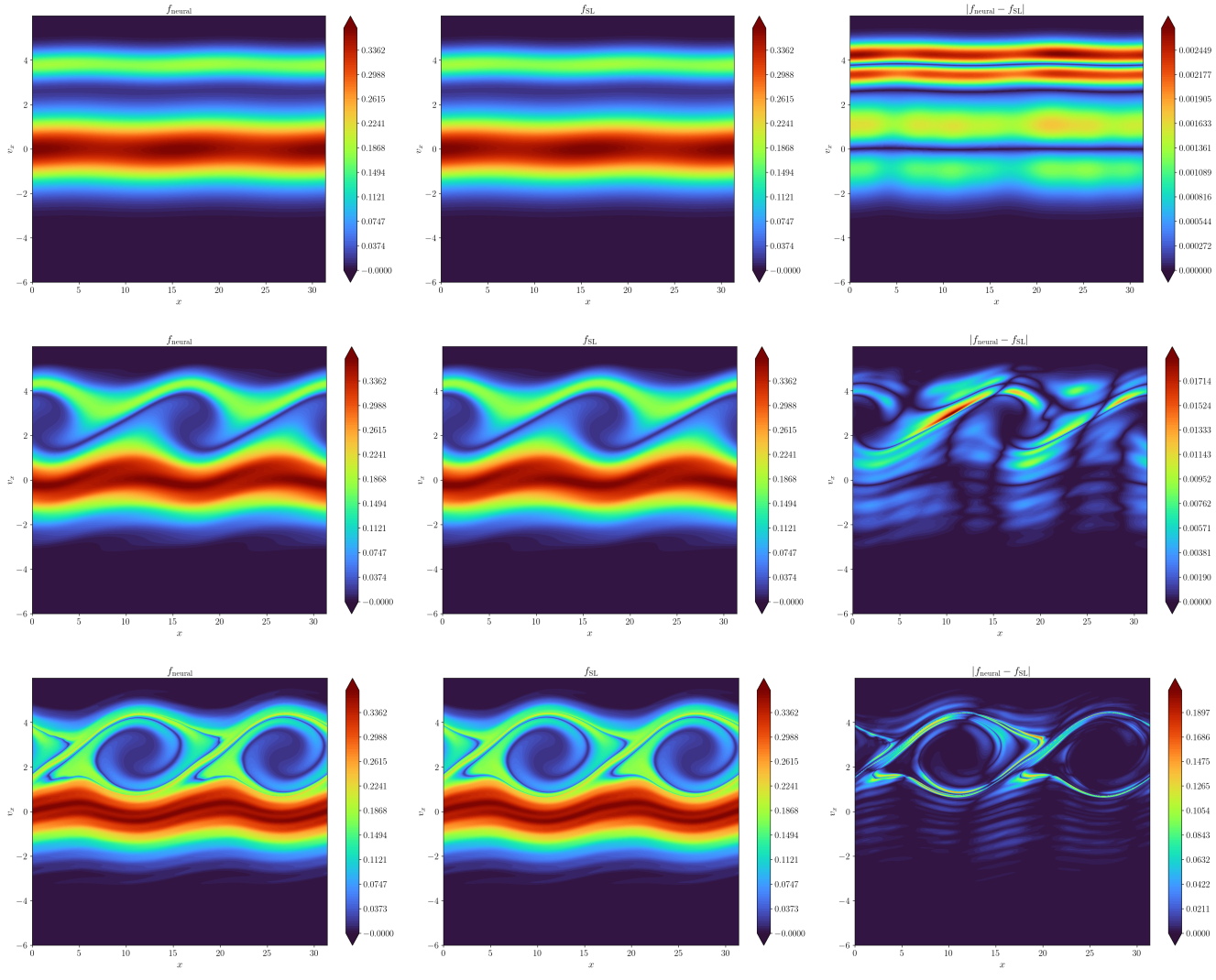


Two-stream instability: Electric energy, lower perturbation

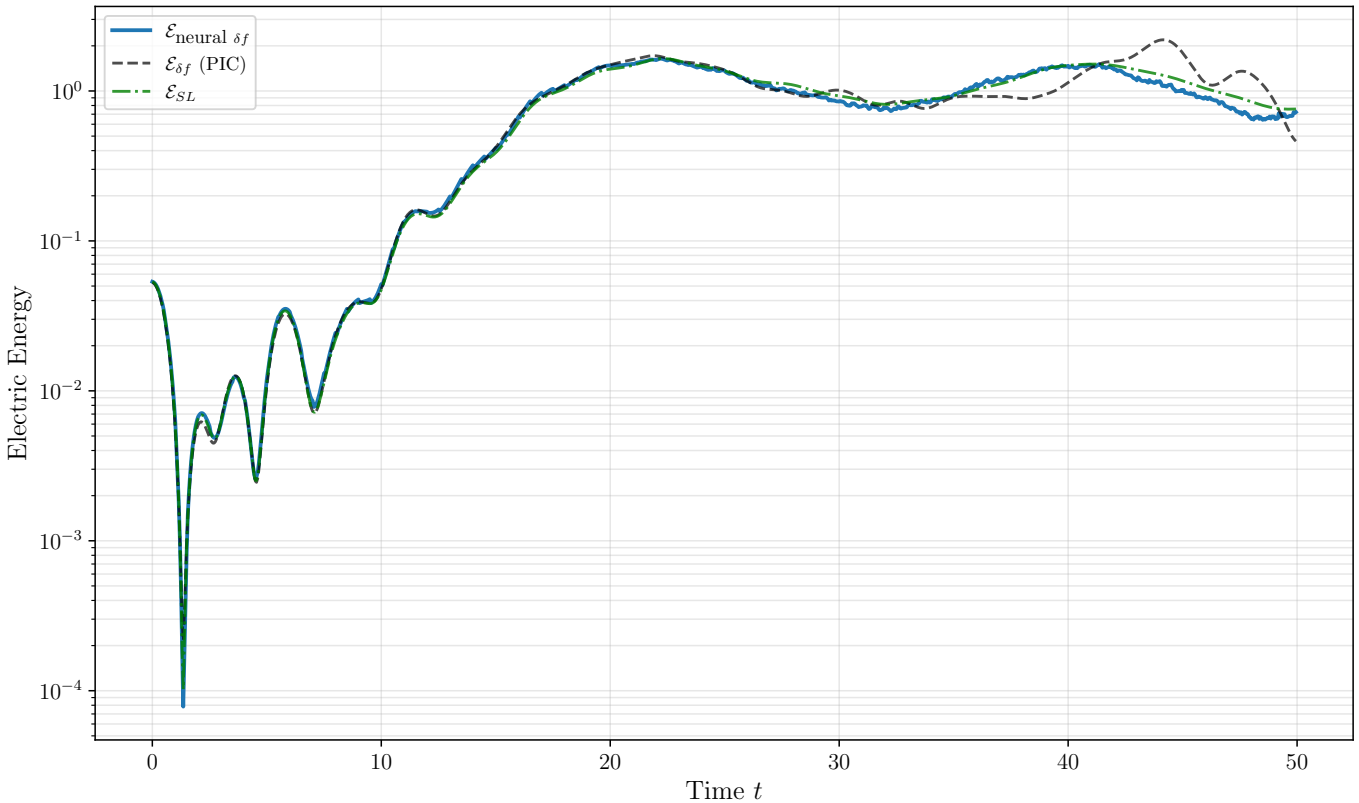


1D1V Bump-on-tail instability: Phase space density

$$f_{init}(x,v) = \left(\frac{0.9}{\sqrt{2\pi}} e^{-v^2/2} + \frac{0.2}{\sqrt{10\pi}} e^{\frac{-(v-3.8)^2}{10}} \right) (1 + 0.03\cos(0.4x))$$

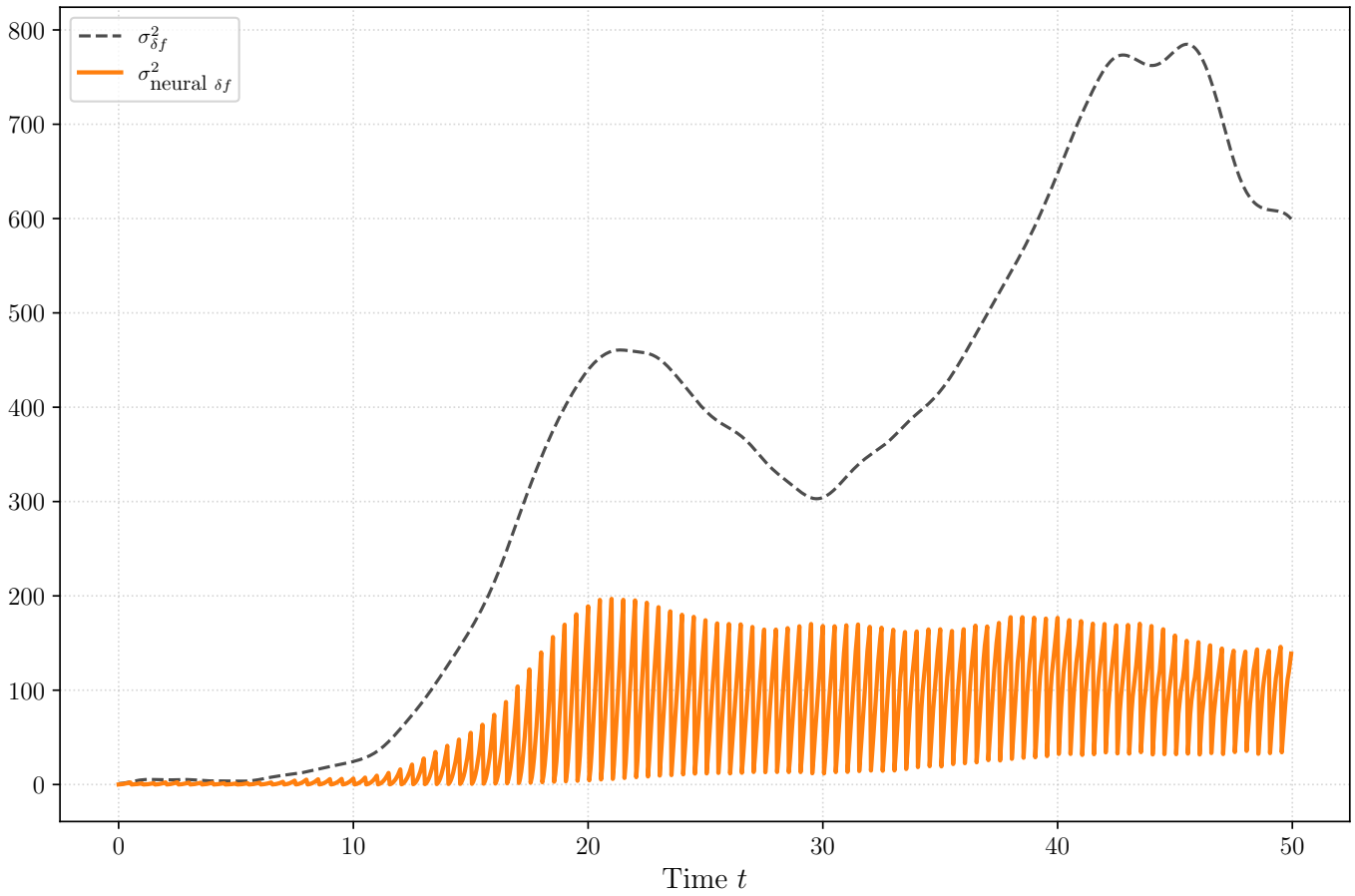


Bump-on-tail instability: Electric energy



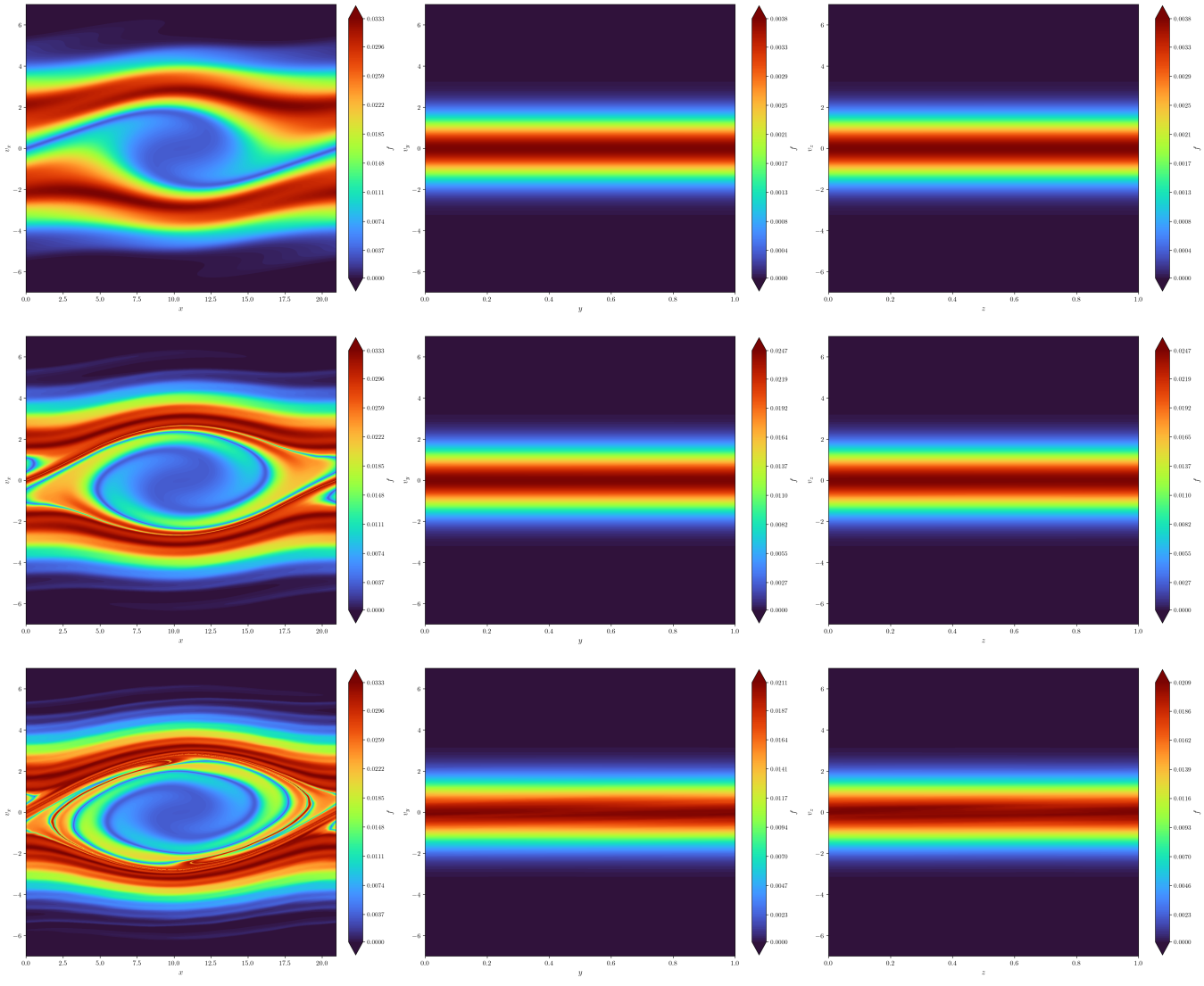
Bump-on-tail instability: Weight Variance

$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$

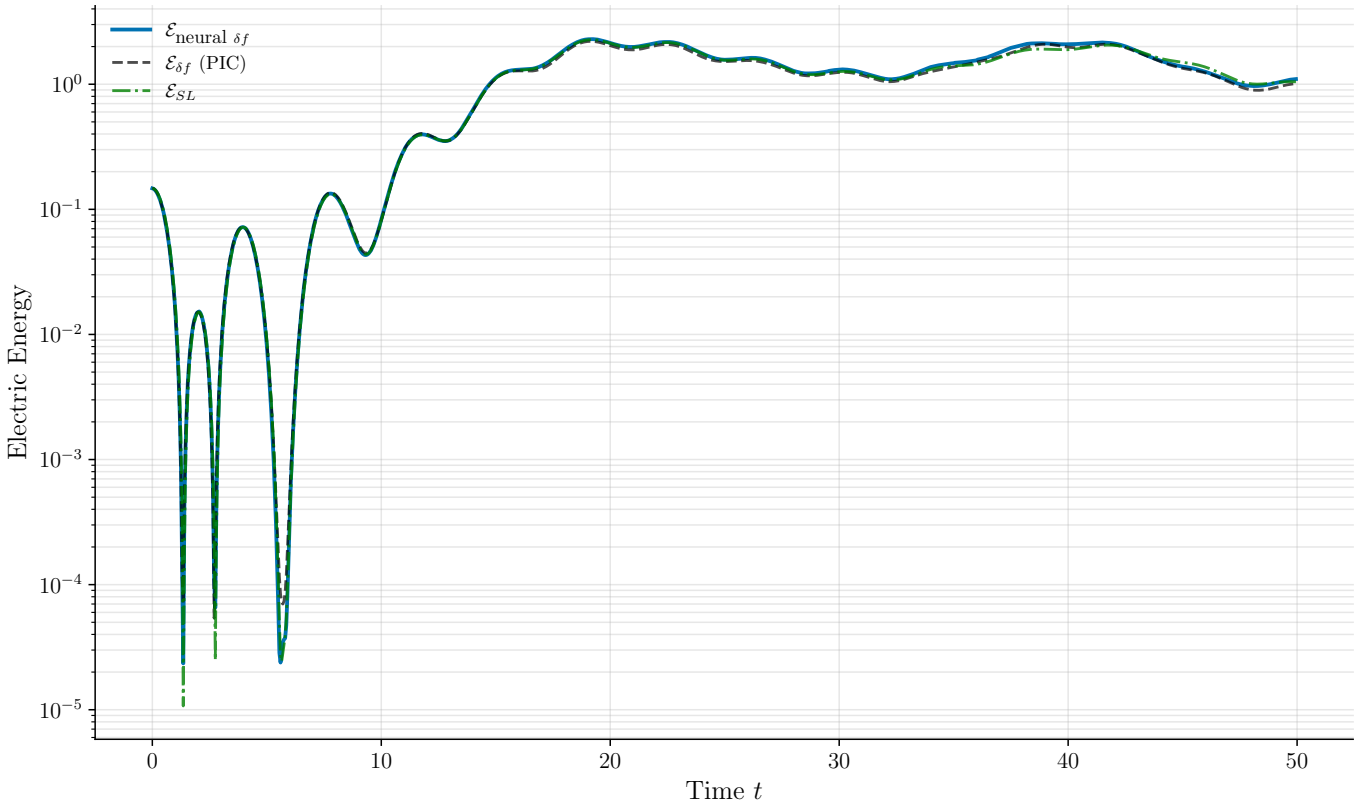


3D3V Two stream instability

$$f_{init}(x,v) = \frac{1}{2\pi} e^{-(v_y+v_z)^2/2} \frac{1}{2\sqrt{2\pi}} (e^{-(v_x-2.4)^2/2} + e^{-(v_x+2.4)^2/2})(1+0.05\cos(0.5x))$$



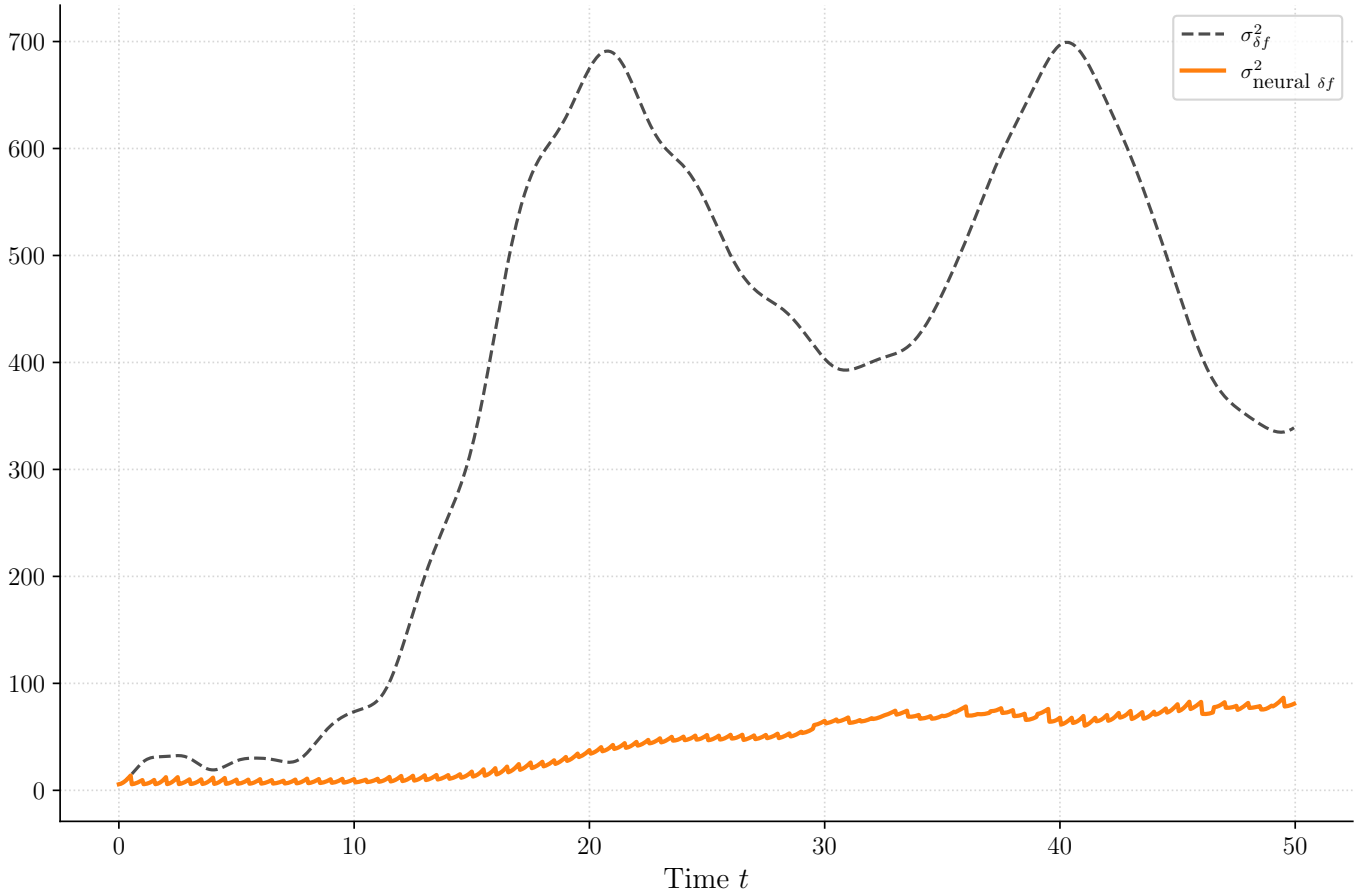
3D Two-stream instability: Electric energy



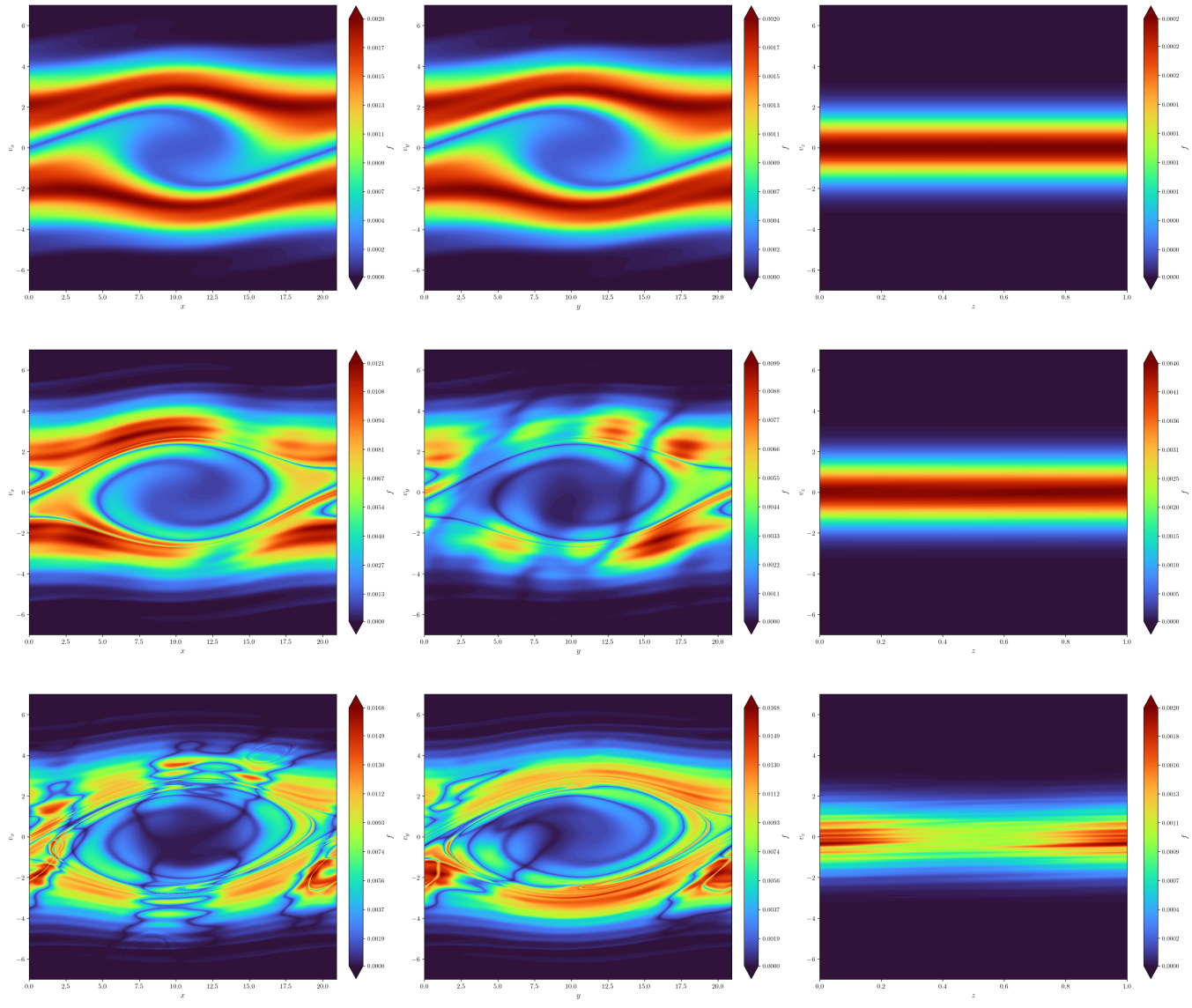
Plot SL: courtesy of Nils Schild. BSL6D: Nils Schild, Mario Räth, Sebastian Eibl, Klaus Hallatschek, Katharina Kormann, A performance portable implementation of the semi-Lagrangian algorithm in six dimensions, Computer Physics Communications, Volume 295, 2024, 108973, ISSN 0010-4655, https://doi.org/10.1016/j.cpc.2023.108973.

3D Two-stream instability: Weight Variance

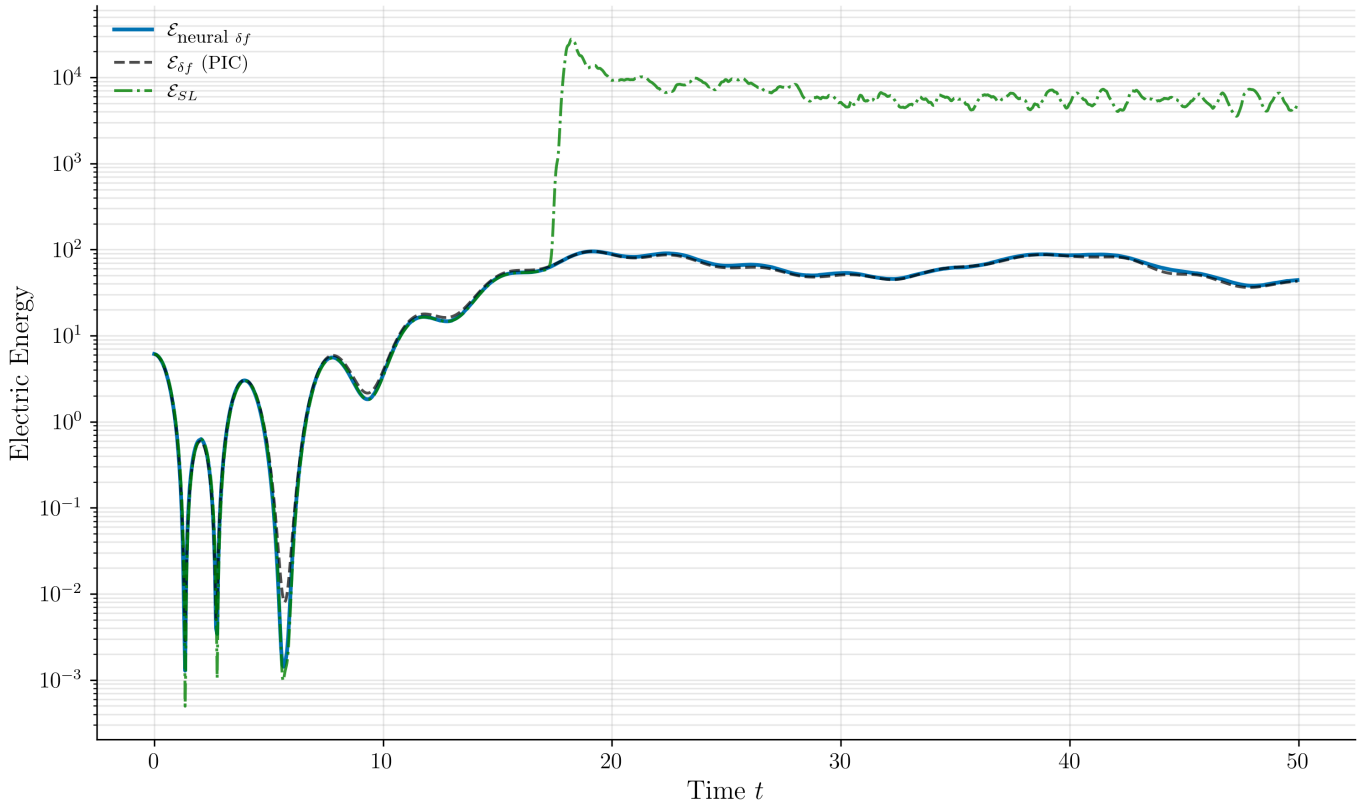
$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$



3D3V Four stream instability



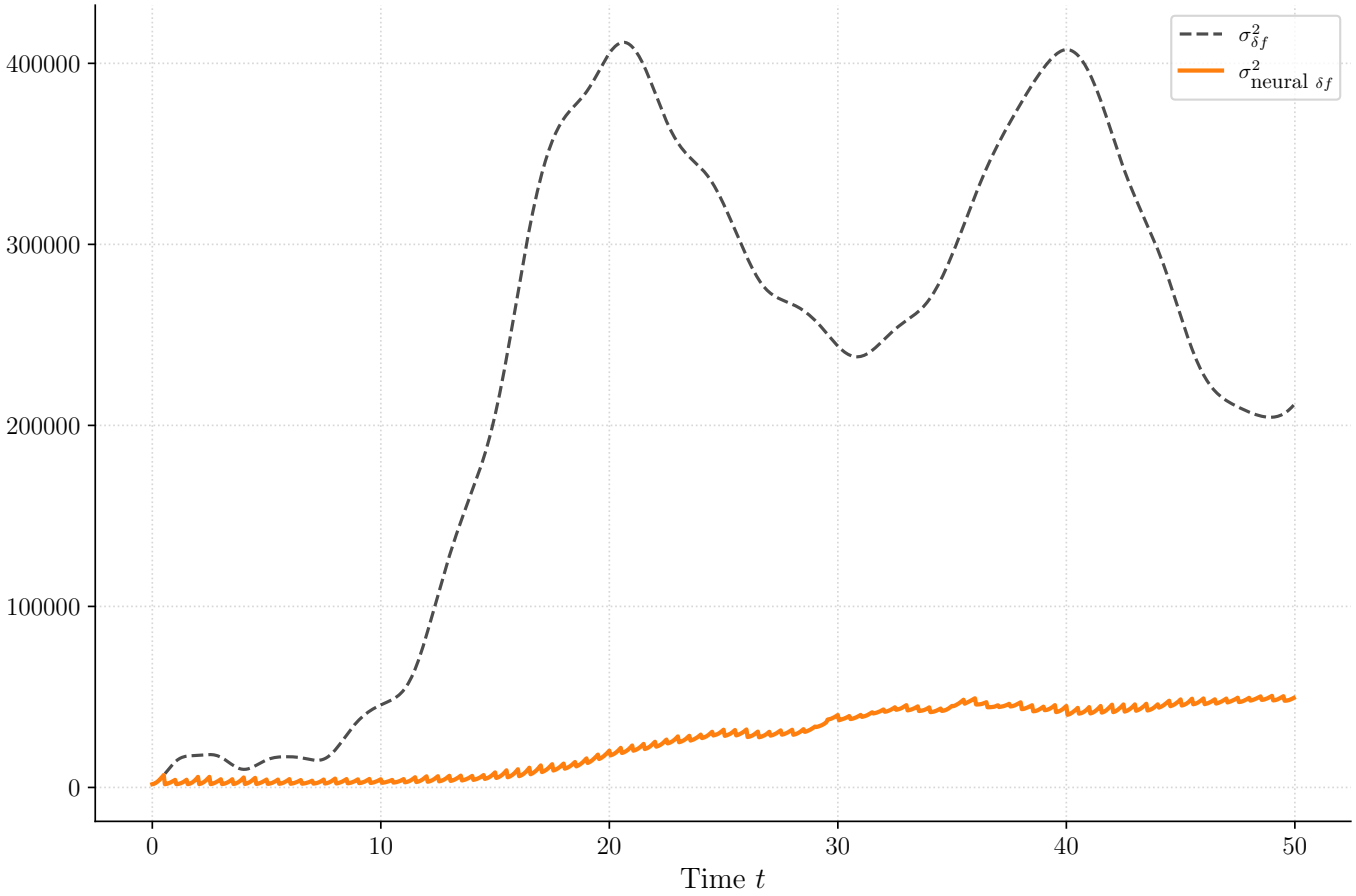
3D Four-stream instability: Electric energy



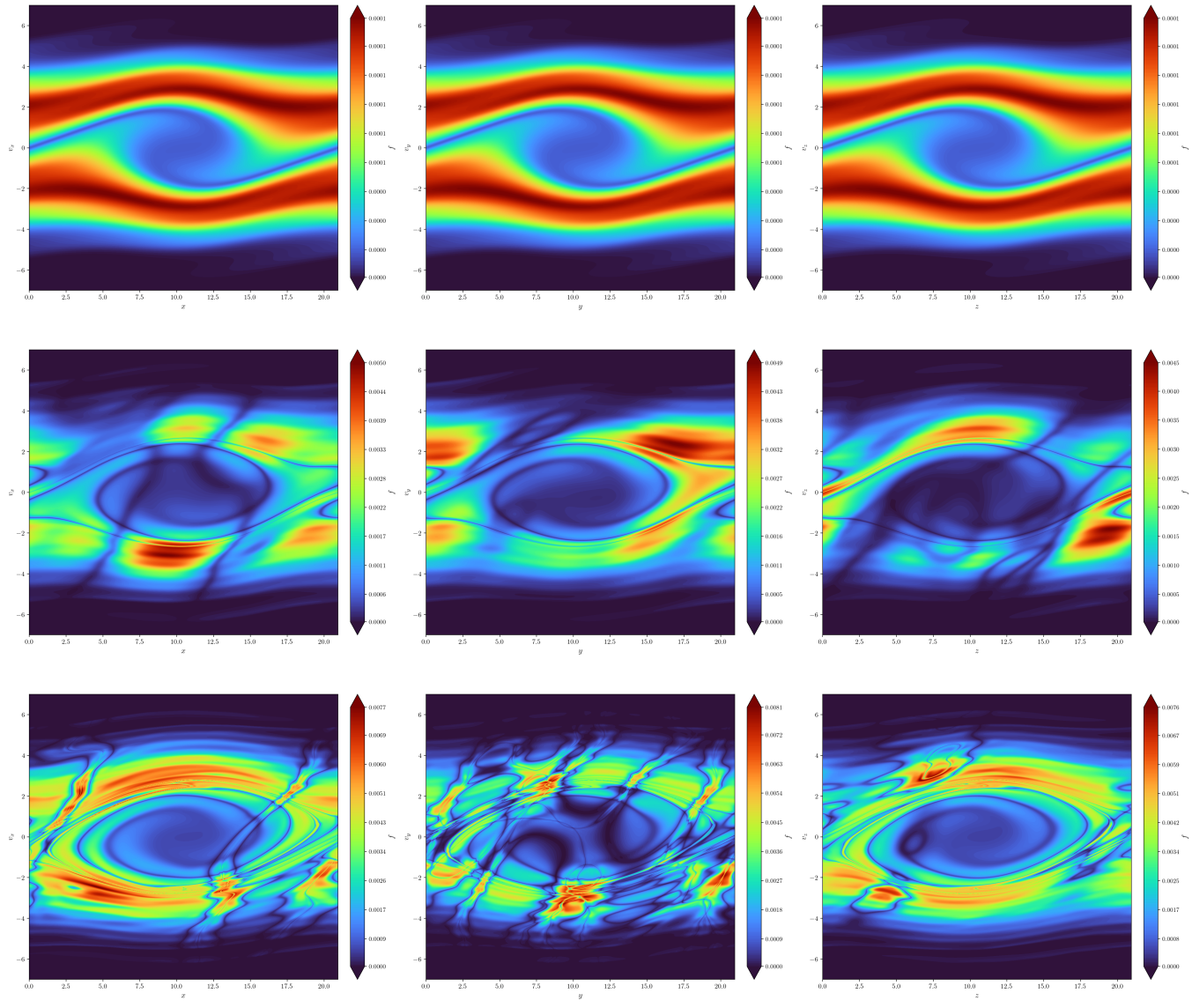
Plot SL: courtesy of Nils Schild. BSL6D code: Nils Schild, Mario Räth, Sebastian Eibl, Klaus Hallatschek, Katharina Kormann, A performance portable implementation of the semi-Lagrangian algorithm in six dimensions, Computer Physics Communications, Volume 295, 2024, 108973, ISSN 0010-4655, https://doi.org/10.1016/j.cpc.2023.108973.

3D Four-stream instability: Weight Variance

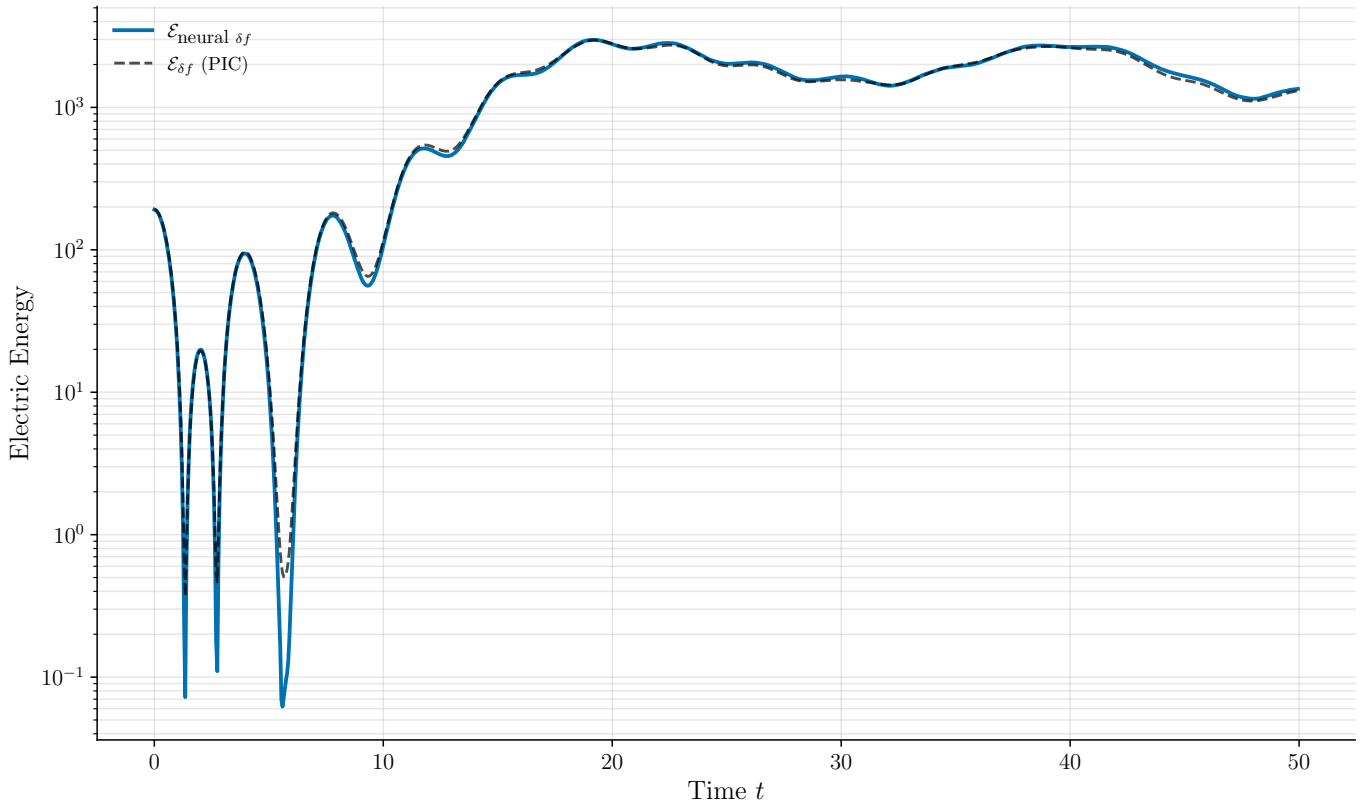
$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$



3D3V Six stream instability

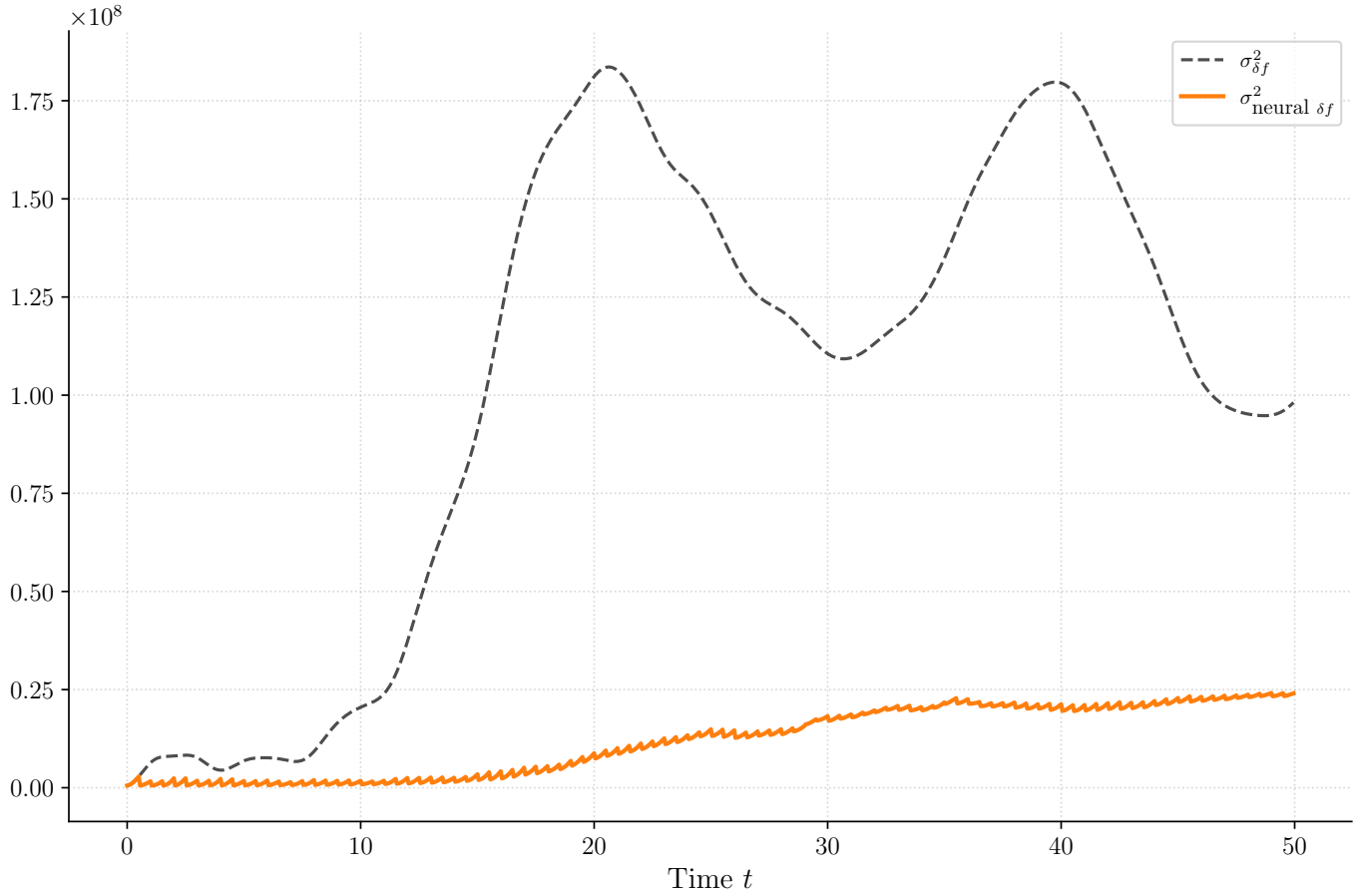


3D Six-stream instability: Electric energy



3D Six-stream instability: Weight Variance

$$\sigma^2 = \frac{1}{N_p}\sum_{k=1}^{N_p}(\delta w_k - \delta\bar{w}_k), \quad \delta\bar{w}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\delta w_k$$



Conclusion

Summary

- δf-**PIC with neural control variate:** use a symplectic neural network Ψ_θ to evolve the bulk f_0^m along the flow, reducing the noisy part δf^n.
- **Structure-preserving:** symplecticity of Ψ_θ is enforced by construction, with periodic boundary conditions built into the architecture.
- **Numerical result:** Validation against semi Lagrangian code, variance reduction of δw_k resulting in a decreased error for the electric energy.

Next steps include

- Extend to gyrokinetic models (with Alberto Bottino and Thomas Hayward-Schneider).