Dynamical feedback control with operator learning for the Vlasov-Poisson system

Jingcheng Lu, Li Wang and Jeff Calder

Abstract

To meet the demands of instantaneous control of instabilities over long time horizons in plasma fusion, we design a dynamic feedback control strategy for the Vlasov–Poisson system by constructing an operator that maps state perturbations to an external control field. In the first part of the paper, we propose learning such an operator using a neural network. Inspired by optimal control theory for linearized dynamics, we introduce a low-rank neural operator architecture and train it via adjoint state method. The resulting controller is effective at suppressing instabilities well beyond the training time horizon. To generalize control across varying initial data, we further introduce a novel cancellation-based control strategy that removes the destabilizing component of the electric field. This approach naturally defines an operator without requiring any training, ensures perturbation decay over infinite time, and demonstrates strong robustness under noisy feedback. Numerical experiments confirm the effectiveness of the method in both one- and multidimensional settings.

Contents

1 Introduction

Long-term suppression of plasma instabilities remains a major challenge in achieving controlled nuclear fusion. Without adequate confinement, the plasma can gradually deviate from its desired state, potentially leading to a breakdown of the reaction or a significant reduction in fusion efficiency. For example, the two-stream instability can cause unwanted beam scattering and loss of focus [31]. Likewise, the bump-on-tail instability, often driven by runaway electrons or radiofrequency heating, can reduce plasma heating

efficiency, degrade confinement, and cause significant energy deposition on the reactor walls. Therefore, it is crucial to control such instabilities through the application of carefully designed external forces.

To place the problem on a more concrete footing, we consider the Vlasov–Poisson (VP) system, which is fundamental for describing collisionless plasma dynamics:

$$\begin{cases} \partial_t f(\mathbf{x}, \mathbf{v}, t) + \mathbf{v}\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{v}, t) + (E(\mathbf{x}, t) + H(\mathbf{x}, t)) \cdot \nabla_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t) = 0, \\[2mm] E(\mathbf{x}, t) = -\nabla_{\mathbf{x}} \Phi(x, t), \quad \nabla_{\mathbf{x}} E(\mathbf{x}, t) = -\Delta \Phi(\mathbf{x}, t) = \rho(\mathbf{x}, t) - \rho_{ion}, \\[2mm] \rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}. \end{cases} \tag{1.1}$$

Here $f(\mathbf{x}, \mathbf{v}, t)$ is particle density at location $\mathbf{x}$, time t, with velocity $\mathbf{v}$, $E(\mathbf{x}, t)$ is the self-generated electric field, $\rho(\mathbf{x}, t)$ is the charge density, and ρ_{ion} is the constant density of background ion. $H(\mathbf{x}, t)$ is the external electric field that will be used for the control purposes.

For ease of discussion, we assume that the desired confined plasma state is described by the distribution $\overline{f}(\mathbf{x}, \mathbf{v})$, which is an equilibrium of the system (1.1) when $H = 0$. There has been an extensive study on the stability or instability of the equilibrium state $\overline{f}(\mathbf{x}, \mathbf{v})$ upon perturbation. In particular, when $\overline{f}(\mathbf{x}, \mathbf{v})$ depends only on $\mathbf{v}$ and follows a Maxwellian distribution, i.e., $\overline{f}(\mathbf{x}, \mathbf{v}) = \overline{f}(\mathbf{v}) \propto e^{-|\mathbf{v}-u|^2/2T}$ for some given bulk velocity u and temperature T, this equilibrium is stable, in the sense that small perturbations in plasma beams lead to the damping of electrostatic waves, a phenomenon known as the Landau damping [19, 27]. In contrast, when $\overline{f}(\mathbf{x}, \mathbf{v}) = \overline{f}(\mathbf{v})$ is a mixture of two Gaussians, the equilibrium becomes unstable; that is, an initial disturbance in the beams is amplified, leading to undesirable growth of the perturbation and the transfer of energy from the plasma beams to electrostatic waves [4]. Additionally, (1.1) can also admit spatially inhomogeneous equilibrium, which may be unstable [13] under certain conditions.

To suppress the growth of perturbation over a long time horizon $t \in [0, T]$, we aim to design an external electric field $H(x, t)$ via the PDE-constrained optimization framework:

$$\min_{H \in L^2_{\mathbf{x}, t}} \mathcal{J}(T; H) \quad \text{s.t.} \quad f \text{ solves } (1.1). \tag{1.2}$$

The loss functional $\mathcal{J}(T; H)$ defines the control objective, which may include terms such as a running loss $\frac{1}{2} \int_0^T \|f(t; H) - \overline{f}\|^2_{\mathbf{x}, \mathbf{v}} dt$, a terminal loss $\frac{1}{2} \|f(T; H) - \overline{f}\|^2_{\mathbf{x}, \mathbf{v}}$, and potentially others. A recent study has analyzed how the choice of objective influences the optimization landscape [11]. The parameterization of H is also crucial for effective control. Directly assuming H to be an arbitrary function of x (or of x and t) typically leads to optimization landscapes with many local minima. As observed in [8], a parameterization of the form

$$H(\mathbf{x}, t; \alpha) = \sum_{k=1}^{r} \alpha_k \phi_k(\mathbf{x}, t) \tag{1.3}$$

with α_k as the parameters to be optimized, not only reduces the dimensionality of the control problem but also tends to convexify the optimization landscape. A moment-based formulation was also proposed in [24], offering a potential reduction of the dimension of (1.2).

Despite the above efforts in selecting the objective function $\mathcal{J}$ or the parameterization of H, a key drawback remains: the optimized field H is only effective over the time interval $[0, T]$. To extend its usefulness beyond T, one would need to solve (1.2) again, which is computationally expensive and incompatible with the practical requirement of near-instantaneous control. For this reason, dynamical feedback control appears to be a more practical alternative, provided such a control can be constructed. Several studies have investigated this direction. One approach is to determine a control $H(t, \mathbf{x})$ over short intervals $[t, t + \Delta t]$, where the objective function is restricted to this local time window, instead of seeking

a control valid over the entire time span $[0, T]$. By discretizing the problem and solving the corresponding optimality conditions, an exact relation between H and the solution f can be derived, which then serves as the feedback law [1]. Another line of work employs reinforcement learning, which assumes the underlying model is unknown and instead learns the feedback law in a data-driven manner [6, 30]. Similar strategies have also been applied in plasma shaping [32, 5] and reactor design optimization [28, 3, 16].

In this paper, we propose a novel approach to deriving the feedback control law by designing and learning a neural operator. Specifically, we aim to learn a mapping from deviation of the particle distribution from equilibrium $\delta f(t, \mathbf{x}, \mathbf{v}) := f(t, \mathbf{x}, \mathbf{v}) - \overline{f}(t, \mathbf{x}, \mathbf{v})$ to $H(t, \mathbf{x})$ such that the control objective $\mathcal{J}$ is effectively minimized. This constitutes a mapping between function spaces: from the space in which δf resides to the space where H is defined, i.e., $H[\delta f](t, \mathbf{x})$. A natural way to approximate such a mapping is through a *neural operator*, which refers to a neural network that is designed to learn a mapping between function spaces, i.e., an operator, instead of a function between Euclidean spaces. Neural operators have been extensively studied in recent years across a wide range of applications. Several architectures have been proposed, including the Fourier Neural Operator (FNO) [22, 17, 21], Low-Rank Neural Operator (LNO) [18], Graph Neural Operator (GNO) [2, 23], and DeepONet [25, 33, 20]. Comprehensive overviews can be found in two recent review articles [26, 18].

To construct such an operator for our setting, we design a specific architecture guided by linear optimal control theory. In particular, we first derive a feedback control law for the linearized system, and then use it as a blueprint to design a neural operator for the general nonlinear problem. The neural operator is then trained via an adjoint-based method. The resulting learned neural operator remains effective over much longer time horizons than directly learning the control field H without feedback information, and it is more robust to noise. Finally, to extend the control performance to arbitrarily long times and arbitrary initial perturbations, we introduce a quasi-optimal law based on a cancellation principle.

The rest of the paper is organized as follows. In the next section, we construct a neural operator to obtain the feedback law, with its structure inspired by linear optimal control theory and its implementation carried out through an adjoint-based method. The effectiveness of the resulting neural operator is demonstrated through extensive examples in Section 3. In Section 4, we propose a novel universal feedback law based on a cancellation argument. This law can be viewed as a special case of a neural operator and is training-free. Numerical experiments further confirm the effectiveness of this approach.

2 Dynamical feedback control with low-rank neural operator

In this section, we investigate the structure of the feedback control $\delta f \mapsto H$. For clarity of presentation, we focus on the one-dimensional case, and assume a spatially homogeneous equilibrium, i.e., $\overline{f}(x, v) \equiv \overline{f}(v)$, although the derivations extend naturally to higher dimensions.

2.1 Operator-based external field

To gain primary insight about the appropriate structure of the mapping from deviation δf to the control field $H(\delta f(t, x, v))(t, x)$:

$$H(\cdot)(\cdot): \ \delta f \mapsto H,$$

we first consider the following linearized Vlasov–Poisson system around the target equilibrium $\overline{f}(v)$. More specifically, write $f(t, x, v) = \overline{f}(v) + \delta f(t, x, v)$, and assume that $\|\delta f\| \ll \|\overline{f}\|$, we have the following

linearized equation for the perturbation δf:

$$\begin{cases} \partial_t \delta f + v \partial_x \delta f + (\delta E + H) \partial_v \overline{f} = 0\,, \\[2mm] \delta E(x,t) = -\partial_x \Phi(x,t)\,, \quad -\partial_x^2 \Phi(x,t) = \delta\rho(x,t)\,, \\[2mm] \delta\rho(x,t) = \int_{-\infty}^{\infty} \delta f(x,v,t) dv\,. \end{cases} \tag{2.1}$$

Assuming the control objective takes the form:

$$\mathcal{J}(\delta f, H) = \int_0^{\infty} \frac{1}{2} \|\delta f(\cdot,\cdot,t)\|_{x,v}^2 + \frac{\gamma}{2} \|H(\cdot,t)\|_x^2 dt, \tag{2.2}$$

where $\gamma > 0$ is the weight that penalizes the magnitude of the control $\|H\|$. We can define the corresponding Hamiltonian:

$$\mathcal{H}(\delta f, H, \lambda) = \iint \frac{1}{2} \delta f(x,v,t)^2 dxdv + \int \frac{\gamma}{2} H(x,t)^2 dx - \iint \lambda(x,v,t)\big(v\partial_x \delta f + (\delta E + H)\partial_v \overline{f}\big) dxdv. \tag{2.3}$$

Let $G(x_1, x_2)$ be the Green's function of the Poisson equation with suitable boundary conditions, the perturbation of the self-generated field δE can be written as

$$\delta E(x,t) = \int \partial_{x_1} G(x,y) \delta\rho(y,t) dy = \iint \partial_{x_1} G(x,y) \delta f(y,v',t) dydv'\,.$$

By interchanging of variables (x,v) and (y,v'), we obtain

$$\iint \lambda(x,v,t)\delta E(x,t)\partial_v \overline{f}(v) dxdv = \iint \Big[\iint \partial_{x_1} G(y,x)\lambda(y,v',t)\partial_v \overline{f}(v') dydv' \Big] \delta f(x,v,t) dxdv.$$

Then the Hamiltonian (2.3) can be rewritten as

$$\begin{aligned} \mathcal{H}(\delta f, H, \lambda) = {}& \iint \frac{1}{2} \delta f(x,v,t)^2 dxdv + \int \frac{\gamma}{2} H(x,t)^2 dx \\ &+ \iint \Big[v\partial_x \lambda(x,v,t) - \langle \partial_{x_1} G(\cdot,x), \int \lambda(\cdot,v',t)\partial_v \overline{f}(v') dv' \rangle \Big] \delta f(x,v,t) dxdv \\ &- \iint \lambda(x,v,t) H(x,t)\partial_v \overline{f}(v) dxdv. \end{aligned}$$

By the Pontryagin Maximum Principle, the optimal control, H^*, and the corresponding adjoint variable, λ^*, satisfy the conditions $\partial_t \lambda = -\dfrac{\partial\mathcal{H}}{\partial \delta f}$ and $\dfrac{\partial\mathcal{H}}{\partial H} = 0$, that is,

$$\begin{cases} \partial_t \lambda^* + v\partial_x \lambda^* - \langle \partial_{x_1} G(\cdot,x), \int \lambda^*(\cdot,v',t)\partial_v \overline{f}(v') dv' \rangle = -\delta f\,, \\[2mm] \gamma H^* - \int \lambda^* \partial_v \overline{f} dv = 0\,. \end{cases} \tag{2.4}$$

The first equation shows the linear dependence of λ^* on δf, and the second equation indicates the linear dependence of H^* on λ^*. Consequently, the optimal control H^* for the linearized system (2.1) is a linear functional of the perturbation δf, although an explicit representation of the mapping $\delta f \mapsto H^*$ is not readily available.

Nevertheless, this linear dependence motivates us to construct the feedback control mechanism through a single-layer low-rank neural operator [18]:

$$H[\delta f(t)](x;\theta) = \sum_{k=1}^{r} \phi_k(x;\theta_\phi) \iint \psi_k(y,v;\theta_\psi)\delta f(y,v,t) dydv, \quad \theta = \{\theta_\phi, \theta_\psi\}. \tag{2.5}$$

4

Here, $\phi^{NN}(x; \theta_\phi) := \big(\phi_1(x; \theta_\phi), \phi_2(x; \theta_\phi), \ldots, \phi_r(x; \theta_\phi)\big)$ is a learnable mapping $\mathbb{R}$ to $\mathbb{R}^r$ consisting of basis functions parameterized by θ_ϕ, while the kernel $\psi^{NN}(x, v; \theta_\psi) = \big(\psi_1(x, v; \theta_\psi), \psi_2(x, v; \theta_\psi), \ldots, \psi_r(x, v; \theta_\psi)\big)$ is a learnable mapping from $\mathbb{R}^2$ to $\mathbb{R}^r$, parameterized by θ_ψ. In the following, we will parameterize ϕ^{NN} and ψ^{NN} by neural networks. When considering a bounded domain $x \in \Omega$ with specific boundary conditions, it is preferable for the basis $\phi^{NN}(x; \theta_\phi)$ to explicitly encode these boundary conditions. We emphasize that, unlike (1.3), the form (2.5) introduces time dependence in H through δf. This provides a more effective way to incorporate time dependence into the control than simply allowing α to vary with time in (1.3). The latter approach not only increases the dimensionality of the optimization problem, since discretizing time into N steps enlarges the parameter space by a factor of N_t, but may also fail to capture meaningful time dependence, as direct optimization can yield coefficients that vary little with t.

In practical computations, we have perturbations only on the computational grid:

$$\delta f^\Delta = \{\delta f(x_i, v_j)\}_{1 \le i \le N_x, 1 \le j \le N_v} \in \mathbb{R}^{N_x N_v}.$$

Then the nodal values of the external field $H^\Delta = \{H[\delta f^\Delta](x_i)\}_{1 \le i \le N_x}$ can be represented by:

$$H_i^\Delta = \sum_{k=1}^r \phi_k(x_i) \sum_{\substack{1 \le l \le N_x \\ 1 \le m \le N_v}} \psi_k(x_l, v_m) \delta f(x_l, v_m) \Delta x \Delta v, \quad 1 \le i \le N_x.$$

Equivalently, this can be written in the matrix form as

$$H^\Delta = \Big([\phi_{r \times N_x}^\Delta]^\top \psi_{r \times N_x N_v}^\Delta \Delta x \Delta v\Big) \delta f^\Delta, \tag{2.6}$$

where $\phi_{r \times N_x}^\Delta = [\phi_k(x_i)]_{k,i}$ and $\psi_{r \times N_x N_v}^\Delta = [\psi_k(x_i, v_j)]_{k,i,j}$ are two projection matrices.

Note that (2.6) is reminiscent of the well-known Linear Quadratic Regulator (LQR) [15, 14] for linear ODE systems, where the optimal control problem is given by

$$\min_{a(t)} \int_0^\infty g(t)^\top Q g(t) + a(t)^\top R a(t) \mathrm{d}t,$$

$$s.t. \qquad \frac{dg}{dt} = Ag(t) + Ba(t),$$

where Q and R are positive definite matrices. If we assume that the optimal control takes the form $a(t) = -Kg(t)$, then the feedback matrix K can be solved from the algebraic Riccati equation,

$$\begin{cases} K = R^{-1} B^\top P, \\ A^\top P + PA - PBR^{-1}B^\top P + Q = \mathbf{0}. \end{cases}$$

In the context of the semi-discrete ODE arising from the linearized equation (2.1), the controller (2.6) corresponds to a linear projection from $\delta f^\Delta \in \mathbb{R}^{N_x N_v}$ to $H^\Delta \in \mathbb{R}^{N_x}$, constructed using low-rank matrices.

We note that, although the exact Vlasov–Poisson system is nonlinear with respect to δf, a properly optimized linear controller can effectively suppress perturbations over an extended period. This will be demonstrated through numerical experiments in Section 3.

2.2 Optimization with adjoint state method

Using the form (2.5) for our neural operator that maps δf to H, we need to define a loss function to determine the optimal basis and kernels so that the control remains effective over an extended period and for a wide range of perturbations. Ideally, the control objective should consider a sufficiently diverse

family of initial perturbations, such that the perturbations evolving over time remain within the subspace generated by this initial family. At the same time, the control horizon should be long enough to capture and suppress any instabilities that may arise. In practice, this is achieved by adding noise to a given initial condition, as will be explained in detail in Section 3.1. For simplicity of presentation, here we consider the control problem for a single initial condition and define the loss function over a sufficiently long time interval $[0, T]$.

Specifically, we optimize the external field operator $H_\theta[\delta f]$, through the following problem,

$$\min_\theta \quad \mathcal{J}(\theta) = \frac{1}{2} \int_0^T ||f(\cdot, \cdot, t; H_\theta) - \overline{f}(\cdot, \cdot)||^2_{x,v} \mathrm{d}t \tag{2.7a}$$

$$s.t. \quad \begin{cases} \partial_t f + v \partial_x f + (E[f] + H_\theta[\delta f]) \partial_v f = 0 \,, \\ \\ E = -\partial_x \Phi, \quad -\partial_x^2 \Phi = \rho - \rho_{ion} \,, \\ \\ \rho = \int f \mathrm{d}v \,. \end{cases} \tag{2.7b}$$

Note that the running loss (2.7a) corresponds to a finite-time truncation of the quadratic loss (2.2) with $\gamma = 0$. In our experiments, we observed that including the control penalty $\frac{\gamma}{2}||H||_2^2$ does not lead to any improvement in computational performance.

To compute the gradient $\nabla_\theta \mathcal{J}(\theta)$, we introduce the Lagrangian multiplier,

$$\mathcal{L}(\theta; \lambda) = \frac{1}{2} \int_0^T ||f(t; H_\theta[\delta f]) - \overline{f}||^2_{x,v} \mathrm{d}t + \int_0^T \iint \lambda(x, v, t) \big(\partial_t f + v \partial_x f + (E[f] + H_\theta[\delta f]) \partial_v f\big) \mathrm{d}x \mathrm{d}v \mathrm{d}t \,.$$

Note that when f satisfies the constraint PDE (2.7b), we have $\mathcal{L}(\theta; \lambda) = \mathcal{J}(\theta)$, and consequently, $\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{J}$, for arbitrary λ. Integration by parts yields

$$\nabla_\theta \mathcal{J} = \nabla_\theta \mathcal{L} = \int_0^T \iint (f(x, v, t) - \overline{f}(x, v)) \nabla_\theta f(x, v, t) \mathrm{d}x \mathrm{d}v \mathrm{d}t + \iint \lambda(x, v, t) \nabla_\theta f(x, v, t)|_{t=0}^T \mathrm{d}x \mathrm{d}v$$

$$- \int_0^T \iint \big(\partial_t \lambda + v \partial_x \lambda + (E[f] + H_\theta[\delta f]) \partial_v \lambda\big) \nabla_\theta f \mathrm{d}x \mathrm{d}v \mathrm{d}t$$

$$+ \int_0^T \iint \big[\nabla_\theta E(x, t) + H[\nabla_\theta \delta f(t)](x; \theta) + (\nabla_\theta H)[\delta f(t)](x; \theta)\big] \lambda(x, v, t) \partial_v f(x, v, t) \mathrm{d}x \mathrm{d}v \mathrm{d}t.$$

By assuming that λ solves the adjoint equation

$$\partial_t \lambda + v \partial_x \lambda + (E[f] + H_\theta[\delta f]) \partial_v \lambda = \delta f, \quad \lambda(x, v, T) = 0, \tag{2.8}$$

the gradient of the loss function reduces to

$$\nabla_\theta \mathcal{J} = \int_0^T \iint \big[\nabla_\theta E(x, t) + H_\theta[\nabla_\theta \delta f(t)](x; \theta) + (\nabla_\theta H_\theta)[\delta f](x; \theta)\big] \lambda(x, v, t) \partial_v f(x, v, t) \mathrm{d}x \mathrm{d}v \mathrm{d}t. \tag{2.9}$$

Note that the adjoint equation (2.8) and the corresponding gradient (2.9) may appear slightly different from those obtained using the classical adjoint state method. In the classical approach as is done in [8], one computes $\nabla_\theta E = \frac{\delta E}{\delta f} \nabla_\theta f$ and $\frac{\delta E}{\delta f}$ enters the adjoint equation (2.8), while $\nabla_\theta E$ itself does not appear in (2.9). Likewise, the term $H_\theta[\nabla_\theta \delta f(t)](x; \theta)$ can be eliminated from (2.9) by explicitly writing:

$$H_\theta[\delta f](x; \theta) = \sum_r \phi_r(x) \iint \psi_r(y, v) \delta f(y, v) \mathrm{d}y \mathrm{d}v$$

$$= \sum_r \phi_r(x) \iint \psi_r(y, v)(f(y, v) - \overline{f}(v)) \mathrm{d}y \mathrm{d}v$$

6

and then

$$H_\theta[\nabla_\theta \delta f](x; \theta) = \sum_r \phi_r(x) \iint \psi_r(y, v) \nabla_\theta f(y, v) \mathrm{d}y \mathrm{d}v\,.$$

Consequently $\sum_r \phi_r(x) \psi_r(y, v)$ enters the adjoint equation (2.8), while $H_\theta[\nabla_\theta \delta f](x; \theta)$ no longer appears in (2.9).

Based on this argument, in practice, we simplify the gradient by freezing the perturbation state and using the following approximation[1]:

$$\nabla_\theta \mathcal{J}(\theta) \approx \int_0^T \iint (\nabla_\theta H)[\delta f(t)](x; \theta) \lambda(x, v, t) \partial_v f(x, v, t) \mathrm{d}x \mathrm{d}v \mathrm{d}t. \tag{2.10}$$

This way, the gradient computation follows the classical adjoint state method, with the only difference being a slight modification in the adjoint equation.

This approximation depends only on the adjoint variable and the explicit parameterization of the control kernel, enabling efficient updates using standard optimizers such as gradient descent or Adam. Our experiments indicate that the approximate gradient (2.10) performs well for moderate perturbations. Modern deep learning packages such as torchdiffeq may be employed to implement auto-differentiation based on semi-discrete ODEs.

To achieve high efficiency in long-term simulations, we will numerically solve the Vlasov-Poisson equations (2.7b) using the semi-Lagrangian method, which is fully explicit and not constrained by a CFL condition on the time step:

- Compute $f^{(1)}(x, v) = f(x - \frac{\Delta t}{2} v, v, t^n)$.

- Compute $f^{(2)}(x, v) = f^{(1)}(x, v - (E[f^{(1)}] + H[\delta f^{(1)}])\Delta t)$, with $E[f^{(1)}] = \nabla(-\Delta)^{-1} \rho^{(1)}$, $\rho^{(1)}(x) = \int f^{(1)}(x, v) \mathrm{d}v$.

- Compute $f^{n+1}(x, v) = f^{(2)}(x - \frac{\Delta t}{2} v, v)$.

The solution $f^{n+1}(x, v)$ at the final stage is then taken as the approximation to $f(x, v, t^{n+1})$. When computing the discrete solution on the mesh $\{(x_i, v_j)\}$, we apply bilinear interpolation to obtain the nodal values on the departure grids, $\{(x_i - \frac{\Delta t}{2} v_j, v_j)\}$ for the first and third stages and $\{(x_i, v_j - (E[f^{(1)}](x_i) + H[\delta f^{(1)}](x_i))\Delta t)\}$ for the second stage. Similarly, the backward time-stepping of the adjoint equation (2.8) can be implemented with

- Compute $\lambda^{(-1)}(x, v) = \lambda(x + \frac{\Delta t}{2} v, v, t^n)$.

- Compute $\lambda^{(-2)}(x, v) = \lambda^{(-1)}(x, v + \frac{\Delta t}{2}(E[f^{n-\frac{1}{2}}] + H[\delta f^{n-\frac{1}{2}}]))$, $f^{n-\frac{1}{2}} = \frac{1}{2}(f^n + f^{n-1})$, $\delta f^{n-\frac{1}{2}} = \frac{1}{2}(\delta f^n + \delta f^{n-1})$.

- Compute $\lambda^{(-3)}(x, v) = \lambda^{(-2)}(x, v) - \Delta t \delta f^{n-\frac{1}{2}}(x, v)$.

- Compute $\lambda^{(-4)}(x, v) = \lambda^{(-3)}(x, v + \frac{\Delta t}{2}(E[f^{n-\frac{1}{2}}] + H[\delta f^{n-\frac{1}{2}}]))$.

---

[1]We point out that the approximation (2.10) is constructed based on the small perturbation assumption $\delta f \ll 1$. However, it also turns out to work well even when the perturbation is moderately large. If the system perturbation grows rapidly, it is not always guaranteed that the convergence with the gradient approximation is stable, in which case one may resort to complete auto-differentiation. Nevertheless, our experiments also indicate that the convergence of the latter can be sometimes unstable either. Therefore, the appropriate optimization approach can be case-by-case.

- $\lambda^{n-1}(x, v) = \lambda^{(-4)}(x + \frac{\Delta t}{2}v, v).$

In the following numerical tests, we employ the adjoint-based method described above, except for one bump-on-tail example where we use `torchdiff` to obtain time-independent control. We emphasize that this adjoint-based method has its advantages, as direct automatic differentiation can be computationally expensive, may introduce significant errors, and, in some cases, result in poorer convergence stability compared to gradient approximation.

3   Numerical experiments

3.1   Computational setups

We examine the effectiveness of operator-based feedback control (2.5) through numerical computations. In all test cases, the computational domain is chosen as $(x, v) \in [0, 10\pi] \times [-8, 8]$, with periodic boundary conditions applied at $x = 0$ and $x = 10\pi$. The background ion density is set to $\rho_{ion} = 1$ and the self-generated electric field is computed by

$$
\begin{cases}
E(x, t) = -\partial_x \Phi(x, t), \\[2mm]
-\partial_x^2 \Phi(x, t) = \rho(x, t) - 1, \ \rho(x, t) = \int f(x, v, t) \mathrm{d}v, \\[2mm]
\Phi(0, t) = \Phi(10\pi, t) = 0.
\end{cases}
$$

The discrete solutions are computed on the uniform mesh with $N_x = 100$ and $N_v = 200$ grids in the x and v directions, respectively. The semi-Lagrangian time integration of the forward and backward equations (2.7b) and (2.8) are implemented with time step $\Delta t = 0.2$.

To enforce exact periodic boundary conditions, rather than learning the basis, we fix the basis $\{\phi_k(\cdot)\}_{k=1}^r$ in (2.5) using $r = 31$ trigonometric functions,

$$
\{\phi_k(x)\}_{k=1}^{31} = \left\{ 1, \sin(\tfrac{l}{5}x), \cos(\tfrac{l}{5}x) \right\}_{1 \le l \le 15}.
$$

The integral kernel, $\psi^{NN}(x, v; \theta_\psi) = (\psi_1(x, v), \psi_2(x, v), \cdots, \psi_{31}(x, v))$, remains a multi-output neural network mapping $\mathbb{R}^2$ to $\mathbb{R}^{31}$. Here we set up ψ^{NN} using a 4-layer fully connected neural network with the distribution of neurons $2 - 64 - 32 - 31$. The output of each hidden layer is activated by the `ReLU` function before being fed to the next layer. In all experiments, the controller $H[\delta f]$ is optimized with respect to the running loss $\frac{1}{2} \int_0^T \|f(t) - \overline{f}\|_2^2 \mathrm{d}t$ with $T = 30$, following the lines of Section 2.2.

To stabilize training in the early stages, we initialize the controller close to zero by setting the parameters of the output layer of ψ^{NN} to zeros. The parameters θ_ψ are then updated using the approximate gradient (2.10) over 3000 iterations. For efficient convergence, we first perform 200 steps of `Adagrad` as a preprocessing phase, followed by `Adam` iterations to accelerate training in the later stages. [2]

To expand the exploration of the state space and prevent the controller from overfitting to data within the finite time horizon $t \in [0, T]$, we add a perturbation f_p to the initial data f_0 at each iteration when computing the gradient, defined as

$$
f_p(x, v) = \varepsilon_p \sum_{k=1}^{k_p} \sum_{n=1}^{l_p} \omega_{kn} \widehat{\phi}_k(x) \widehat{h}_n(v).
$$

---

[2] We point out that the combination of Adagrad and Adam iterations is employed due to the fact that the Adagrad method, with fast gradient decay, converges quickly at the early stage but gets stuck afterwards. In contrast, the Adam method can be oscillatory under randomly initialized network parameters, but it helps accelerate iterations after Adagrad preprocessing. Thus, combining optimizers with different features would better ensure the stability and efficiency of training at different phases. The optimal choice of optimizers and/or combinations will be left for future discussions.

Here $\{\widehat{\phi}(x)\}$ and $\{\widehat{h}_l(v)\}$ are orthonormal basis in x and $v$directions, respectively. The weight vector, ω $= \{\omega_{kl}\}_{1 \leq k \leq k_p, 1 \leq n \leq n_p} \in \mathbb{R}^{k_p \times n_p}$, is randomly and uniformly sampled from the $k_p \times n_p$−dimensional unit ball. In our computations, we construct f_p using the normalized trigonometric functions on $x \in [0, 10\pi]$,

$$\{\widehat{\phi}_k(x)\}_{k=1}^{11} = \left\{ \sqrt{\frac{1}{10\pi}}, \sqrt{\frac{1}{5\pi}} \sin(\tfrac{l}{5}x), \sqrt{\frac{1}{5\pi}} \cos(\tfrac{l}{5}x) \right\}_{1 \leq l \leq 5},$$

and the orthonormal Hermite basis functions on $v \in (-\infty, \infty)$,

$$h_n(v) = \sqrt{\frac{1}{\sqrt{2\pi}n!}} He_n(v) \exp(-\tfrac{v^2}{4}), \quad 0 \leq n \leq 5.$$

Here, the Hermite polynomials $He_n(v)$, can be computed recursively,

$$\begin{aligned} He_0(v) &= 1, \\ He_{n+1}(v) &= vHe_n(v) - nHe_{n-1}(v), \quad n \geq 1. \end{aligned}$$

The magnitude of f_p is set to $\varepsilon_p = 10^{-3}$.

At the end of training, we use the future loss, $\|f(2T) - \overline{f}\|_2$, as a measure of generalizability to select the best iteration. The parameters corresponding to this iteration are then adopted as the final trained model. This approach differs from simply taking the parameters at the end of the 3000 iterations. Because our training introduces noise, the iterations may explore multiple local minima rather than converge to a single one. Therefore, the training should be viewed as a search process, and the historically best result is used.

To demonstrate the benefit of introducing dynamical feedback control, we will also include the time-independent counterpart for comparison, given by

$$H_{indep}(x) = \sum_{k=1}^{15} \theta_k \sin(\tfrac{k}{5}x) + \sum_{k=0}^{15} \theta_{k+16} \cos(\tfrac{k}{5}x). \tag{3.1}$$

In this case, the gradient $\nabla_\theta H_{indep}(x; \theta)$ reduces to $(\phi_1(x), \phi_2(x), \cdots \phi_{31}(x))^\top$ in the gradient (2.10).

3.2 Two-stream instability

We consider the following two-stream distribution as the target equilibrium:

$$\overline{f}(v) = \frac{1}{2\sqrt{2\pi}} \exp(-\tfrac{(v-\bar{v})}{2}) + \frac{1}{2\sqrt{2\pi}} \exp(-\tfrac{(v+\bar{v})}{2}), \quad \bar{v} = 2.4.$$

Assume that the Vlasov-Poisson system is initialized with the perturbed state

$$f_0(x, v) = (1 + \varepsilon \cos(\tfrac{x}{5})) \overline{f}(v), \quad x \in [0, 10\pi], \quad \varepsilon = 0.001.$$

The two-stream equilibrium is known to be unstable, with perturbations amplified by the self-generated electric field. Our goal is to suppress this instability using an operator-based external field. The training procedure follows the approach outlined in Section 3.1, employing a learning rate of $lr = 5 \times 10^{-3}$ for the `Adagrad` iterations and $lr = 5 \times 10^{-4}$ for the subsequent `Adam` iterations.

Figure 3.1 shows the evolution of the L^2−state perturbation, $\frac{1}{2} \iint |f(x, v, t) - \overline{f}(v)|^2 \mathrm{d}x \mathrm{d}v$, and the electric energy of the self-generated field, $\frac{1}{2} \int E(x, t)^2 \mathrm{d}x$, over the time interval $t \in [0, 70]$. It is observed

9

that the time-independent external field, optimized over $t \in [0, 30]$, effectively suppresses the perturbation within the optimization interval but quickly loses effectiveness as time progresses. In contrast, the dynamical feedback control improves stability within the optimization interval and maintains effectiveness well beyond it. This behavior is further confirmed by the particle state distribution f, shown in Figure 3.2. Figure 3.3 depicts the external fields at different time points, revealing that for $t > T$, the structure of the operator-based external field can be notably different from that of H_{indep}.
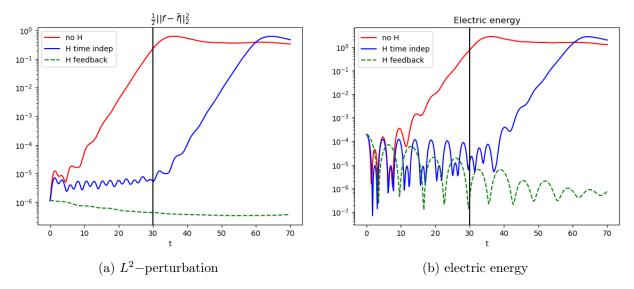

(a) L^2-perturbation

(b) electric energy

Figure 3.1: Two-stream instability. History of L^2-state perturbation and electric energy over $t \in [0, 70]$. No external field (red lines) vs time-independent control (blue lines) vs dynamical feedback control (green lines). The vertical black line marks the terminal time of optimization.

3.3 Bump-on-tail instability

We also examine the performance of our operator-based feedback control when applied to suppress the bump-on-tail instability. The following target equilibrium is considered:

$$\overline{f}(v) = \frac{\omega_1}{\sqrt{2\pi}} \exp(-\frac{(v-\bar{v}_1)^2}{2}) + \frac{\omega_2}{\sqrt{2\pi v_t}} \exp(-\frac{(v-\bar{v}_2)^2}{2v_t}).$$

In our experiment, we set $\omega_1 = 0.9$, $\omega_2 = 0.1$, $\bar{v}_1 = -2$, $\bar{v}_2 = 3.5$, $v_t = 0.25$. We initialize the VP system by imposing a small perturbation to the equilibrium,

$$f_0(x, v) = \overline{f}(v) + \frac{\varepsilon \omega_2}{\sqrt{2\pi v_t}} \exp(-\frac{(v-\bar{v}_2)^2}{2v_t}) \sin(\frac{x}{5}), \quad x \in [0, 10\pi], \quad \varepsilon = 0.003,$$

which leads to the destruction of the high-velocity thin tail as time develops. The controller network is trained with the learning rate $lr = 2 \times 10^{-3}$ for Adagrad preprocessing and $lr = 3 \times 10^{-4}$ for Adam iterations. We compare the solutions produced by the time-independent control (3.1) and the feedback control (2.5) in Figures 3.4–3.5, clearly demonstrating the superior effectiveness of the feedback control.

3.4 Robustness under noisy feedback

In practical applications, measured data may be inaccurate, potentially undermining the effectiveness of feedback control. In this section, we test the robustness of our algorithm under noisy measurements. The
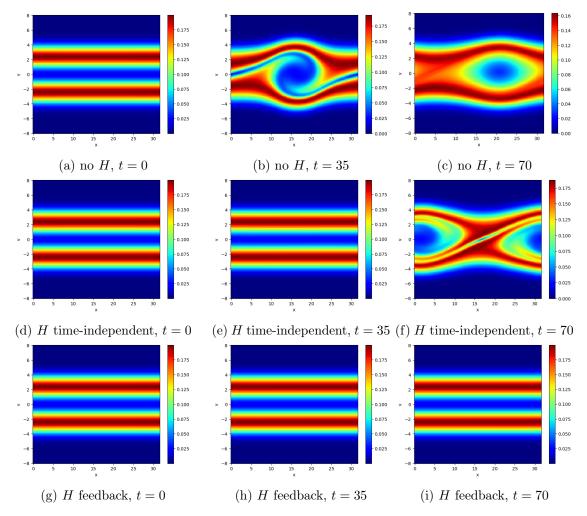
10

Figure 3.2: Two-stream instability. Evolution of $f(x, v, t)$. No external field (first row) vs time-independent control (second row) vs dynamical feedback control (third row).

operator-based controllers are the same as those obtained in Sections 3.2 and 3.3, except that the external field, $H[\delta f^\sigma(t)](x)$, perceives noisy feedback from the environment,

$$\delta f^\sigma(x, v, t) = \delta f(x, v, t) + \sigma \mathcal{N}_{x,v,t} \,. \tag{3.2}$$

Here $\{\mathcal{N}_{x,v,t}\}_{x,v,t}$ are independent and identically distributed according to the standard normal distribution.

Figures 3.7 – 3.8 show simulations of the two-stream instability, with measurements δf affected by noise of varying strengths. The initial particle state perturbation has magnitude $||\delta f_0||_\infty \approx 1.99 \times 10^{-4}$. In Figure 3.7 we observe that when the noise is small (compared to the initial perturbation), the operator-based external field maintains a clear advantage in terms of providing effective long-time instability control. As the magnitude of the noise increases to $\sigma = 1 \times 10^{-4}$ ($\approx 50\%||\delta f_0||_\infty$) the performance of $H[\delta f^\sigma]$ is notably deteriorated. However, for $t > 30$, our feedback control construction can still suppress the growth of the perturbation much better than the time-independent counterpart. As shown in Figure 3.8, the particle distribution generated by noisy feedback control remains relatively close to the target equilibrium up to $t = 70$. Figures 3.9 – 3.10 show the computations for the bump-on-tail instability ($||\delta f_0||_\infty \approx 2.39 \times 10^{-4}$). Although the inaccurate measurement of perturbation weakens the performance of our algorithm, the control remains effective over a long time period, even under relatively large noise.
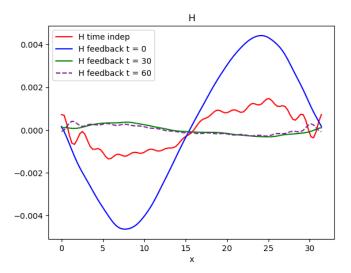
Figure 3.3: Two-stream instability. External fields by time-independent control and dynamical feedback control.
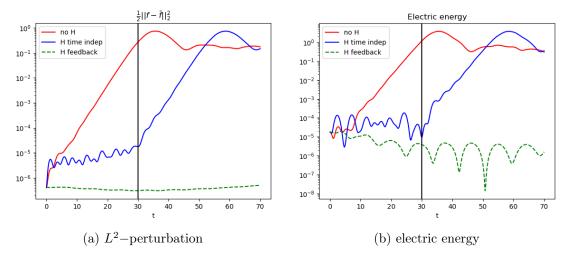

(a) L^2-perturbation

(b) electric energy

Figure 3.4: Bump-on-tail instability. History of L^2-state perturbation and electric energy over $t = 70$. No external field (red lines) vs time-independent control (blue lines) vs dynamical feedback control (green lines). The vertical black line marks the terminal time of optimization.

4 A quasi-optimal universal control

4.1 A cancellation-based construction

While the PDE-constrained optimization problem (2.7) offers a flexible framework for optimizing the controller of the Vlasov–Poisson system (and even more general plasma models such as the magnetically confined Vlasov–Maxwell equations), the optimal control for the nonlinear system (1.1) can be sensitive to the choice of the initial dataset f_0 and the time horizon T. If the dataset is not exhaustive or T is not sufficiently large, the resulting control may lose effectiveness over time, necessitating retraining and incurring significant computational cost. To address this limitation, we propose a universal operator architecture that reduces such dependence. Specifically, we develop a quasi-optimal control strategy based on electric field cancellation, which depends only on the target equilibrium and can therefore be applied robustly across a broad range of initial conditions.
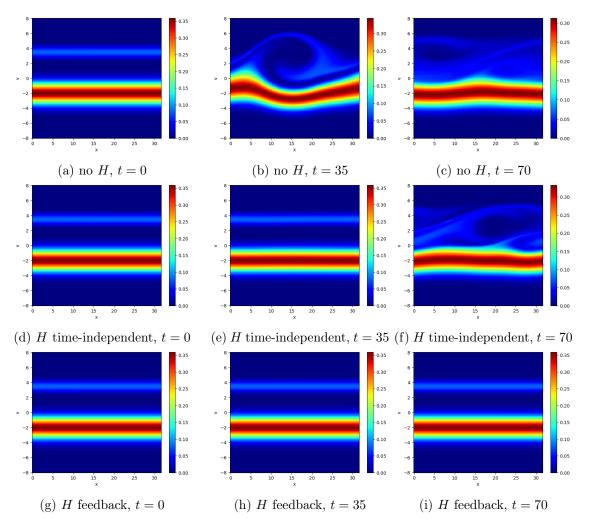
(a) no H, $t = 0$ (b) no H, $t = 35$ (c) no H, $t = 70$

(d) H time-independent, $t = 0$ (e) H time-independent, $t = 35$ (f) H time-independent, $t = 70$

(g) H feedback, $t = 0$ (h) H feedback, $t = 35$ (i) H feedback, $t = 70$

Figure 3.5: Bump-on-tail instability. Evolution of particle distribution $f(x, v, t)$. No external field (first row) vs time-independent control (second row) vs dynamical feedback control (third row).

Subtracting the equilibrium $\overline{f}(x, v)$, which satisfies $v\partial_x \overline{f} + \overline{E}\partial_v \overline{f} = 0$, from the original VP system (1.1), we obtain the evolution equation for $\delta f(x, v, t) = f(x, v, t) - \overline{f}(x, v)$:

$$\partial_t \delta f + v\partial_x \delta f + \overline{E}\partial_v \delta f + (\delta E + H)\partial_v f = 0, \tag{4.1}$$

where $\overline{E}(x)$ is the self-generated electric field of the equilibrium. The electric field perturbation, $\delta E = E - \overline{E}$, can be written in the operator form:

$$\delta E[\delta f(t)](x) = -\partial_x(-\partial_x^2)^{-1}\delta\rho(x, t), \quad \delta\rho(x, t) = \int \delta f(x, v, t)\mathrm{d}v.$$

In particular, when the analytical form of the Green's function $G(x, y)$ is available for the Poisson equation with specific boundary condition, δE can be conveniently evaluated with

$$\delta E[\delta f(t)](x) = -\int \partial_x G(x, y)\delta\rho(y, t)\mathrm{d}y = -\iint \partial_x G(x, y)\delta f(y, v, t)\mathrm{d}v\mathrm{d}y. \tag{4.2}$$

For instance, for the Poisson equation defined on the interval $x \in [a, b]$ with zero boundary data (which

13

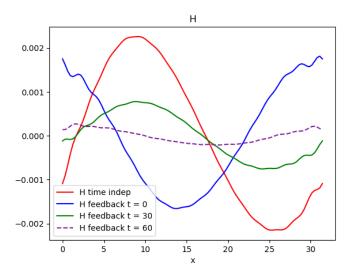Figure 3.6: Bump-on-tail instability. External fields by time-independent control and dynamical feedback control.

(a) $\sigma = 2 \times 10^{-5}$ $(0.1 \cdot ||\delta f_0||_\infty)$ (b) $\sigma = 5 \times 10^{-5}$ $(0.25 \cdot ||\delta f_0||_\infty)$ (c) $\sigma = 1 \times 10^{-4}$ $(0.5 \cdot ||\delta f_0||_\infty)$
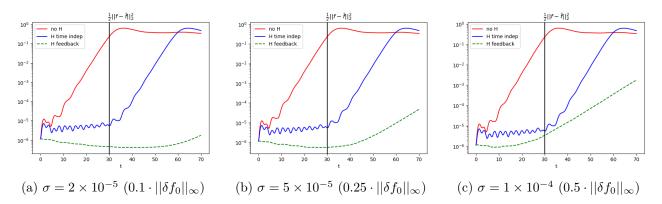
Figure 3.7: Two stream instability with noisy feedback. History of L^2-state perturbation over $t \in [0, 70]$. No external field (red lines) vs time-independent control (blue line) vs dynamical feedback control (green lines) under varying noise magnitudes σ. The vertical black line marks the terminal time of optimization.
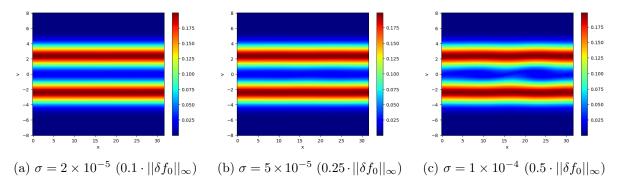

(a) $\sigma = 2 \times 10^{-5}$ $(0.1 \cdot ||\delta f_0||_\infty)$ (b) $\sigma = 5 \times 10^{-5}$ $(0.25 \cdot ||\delta f_0||_\infty)$ (c) $\sigma = 1 \times 10^{-4}$ $(0.5 \cdot ||\delta f_0||_\infty)$

Figure 3.8: Two stream instability with noisy feedback. Particle distribution $f(x, v, t)$ at $t = 70$ generated by dynamical feedback control under varying noise magnitude σ.

is the computational setup in Section 3.1), the Green's function is explicitly given by:

$$
G(x, y) = \begin{cases} \dfrac{(x - a)(b - y)}{b - a}, & a \le x \le y \le b, \\[3mm] \dfrac{(y - a)(b\overline{14}x)}{b - a}, & a \le y \le x \le b. \end{cases}
$$

(a) $\sigma = 2.4 \times 10^{-5}$ $(0.1 \cdot ||\delta f_0||_\infty)$ (b) $\sigma = 6 \times 10^{-5}$ $(0.25 \cdot ||\delta f_0||_\infty)$ (c) $\sigma = 1.2 \times 10^{-4}$ $(0.5 \cdot ||\delta f_0||_\infty)$
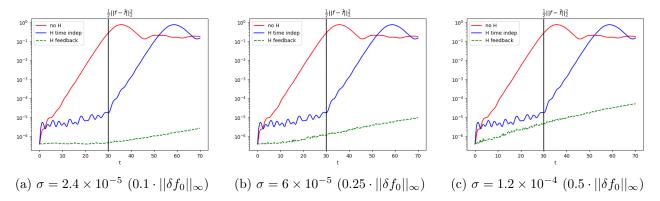
Figure 3.9: Bump-on-tail instability with noisy feedback. History of L^2-state perturbation. No external field (red lines) vs time-independent control (blue line) vs dynamical feedback control (green lines) under varying noise magnitude σ. The vertical black line marks the terminal time of optimization.
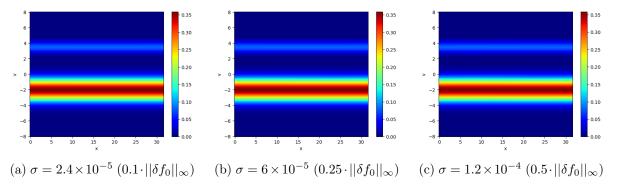

(a) $\sigma = 2.4 \times 10^{-5}$ $(0.1 \cdot ||\delta f_0||_\infty)$ (b) $\sigma = 6 \times 10^{-5}$ $(0.25 \cdot ||\delta f_0||_\infty)$ (c) $\sigma = 1.2 \times 10^{-4}$ $(0.5 \cdot ||\delta f_0||_\infty)$

Figure 3.10: Bump-on-tail instability with noisy feedback. Particle distribution $f(x, v, t)$ at $t = 70$ generated by dynamical feedback control under varying noise magnitude σ.

Alternatively, efficient algorithms such as fast Fourier transforms [10, Chapter 8], multigrid methods [34, 12], and fast multipole methods [29, 9] can be applied to solve the Poisson equation directly.

Physically, perturbations around an unstable equilibrium are amplified by the self-generated electric field. To counter this effect, we consider the construction

$$H[\delta f(t)](x) = -\delta E[\delta f(t)](x) + \delta H[\delta f(t)](x).$$

The goal is to cancel the destabilizing self-generated field E while introducing an auxiliary term δH to further suppress the perturbation. Substituting this external field into (4.1) and integrating the resulting equation against δf, we obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}||\delta f(t)||_2^2 = -\int \overline{E}(x)\Big[\int \delta f(x,v,t)\partial_v\delta f(x,v,t)\mathrm{d}v\Big]\mathrm{d}x - \int \delta H(x,t)\Big[\int \delta f(x,v,t)\partial_v f(x,v,t)\mathrm{d}v\Big]\mathrm{d}x$$

$$= -\int \delta H(x,t)\Big[\int \delta f(x,v,t)\partial_v\overline{f}(x,v)\mathrm{d}v\Big]\mathrm{d}x.$$

(4.3)

The second line is obtained by using the fact that $f = \overline{f} + \delta f$ and $\int \delta f \partial_v \delta f \mathrm{d}v = 0$. In particular, by setting

$$\delta H[\delta f(t)](x) = \gamma \int \delta f(x,v,t)\partial_v\overline{f}(x,v)\mathrm{d}v, \quad \gamma > 0,$$

15

the equation (4.3) implies the decay of perturbation in time,

$$\frac{1}{2}\frac{d}{dt}||\delta f(t)||_2^2 = -\gamma \int \left| \int \delta f(x,v,t)\partial_v \overline{f}(x,v)\mathrm{d}v \right|^2 \mathrm{d}x \leq 0. \tag{4.4}$$

This yields a convenient cancellation-based feedback controller,

$$H[\delta f(t); \overline{f}](x) = -\delta E[\delta f(t)](x) + \gamma \int \delta f(x,v,t)\partial_v \overline{f}(x,v)\mathrm{d}v, \quad \gamma > 0,$$

$$\delta E[\delta f(t)](x) = -\iint \partial_x G(x,y)\delta f(y,v,t)\mathrm{d}v\mathrm{d}y. \tag{4.5}$$

Note that this construction can be regarded as a special case of the linear integral operator (2.5) where both the basis and weights are fixed analytically.

Although the control (4.5) does not ensure $||\delta f(t)|| \overset{t\to\infty}{\longrightarrow} 0$, it still provides satisfactory control when the initial perturbation is already small enough Figure 4.1 presents the evolution of the L^2-state perturbation $\frac{1}{2}||\delta f(t)||_2^2$, and the electric energy $\frac{1}{2}||E(t)||_2^2$, generated by the controller (4.5) in the two-stream instability test case (see Section 3.2 for setups). The results confirm the universality of the controller with respect to different initial states f_0. Meanwhile, a larger γ leads to faster decay in the perturbation. Figure 4.2 demonstrates the performance of the controller under noisy feedback of the form (3.2). The initial data is set to $f_0 = (1 + 0.001\cos(\frac{1}{5}x))\overline{f}(v)$, in consistency with the setting in Section 3.4. The results show the notably improved robustness of the cancellation-based method as compared to the aforementioned neural operator-based construction (see Figure 3.7).

Remark 4.1 *The construction in* (4.5) *is not unique. For example, an alternative choice is:*

$$\delta H(x,t) = \gamma \frac{\int |\delta f(x,v,t)|^2 \mathrm{d}v}{\int \delta f(x,v,t)\partial_v \overline{f}(x,v)\mathrm{d}v + \varepsilon}, \tag{4.6}$$

where a small bias term ε, having the same sign as $\int \delta f(x,v,t)\partial_v \overline{f}(x,v)\mathrm{d}v$, is introduced to mitigate severe round-off errors when the denominator is close to zero. Theoretically, the construction (4.6) *yields a sharper decay estimate,*

$$\frac{1}{2}\frac{d}{dt}||\delta f(t)||_2^2 = -\gamma||\delta f(t)||_2^2 \rightsquigarrow ||\delta f(t)||_2 \leq ||\delta f(0)||_2^2 \cdot e^{-\gamma t}.$$

However, our numerical experiments show that the controller defined in (4.6)*, owing to its more intricate structure, is more sensitive to noisy feedback than the simpler linear controller in* (4.5)*. Consequently, the latter is preferable in practice, as it offers lower computational cost and greater robustness in noisy environments.*

Remark 4.2 *We highlight the difference between our cancellation-based feedback law and the analytical control field obtained in [7], which was derived using a pole-elimination technique. This technique relies on the linearized system, along with a Fourier transform-based method. In contrast, our approach is formulated directly on the original nonlinear system and ensures analytical decay of $||\delta f(t)||_2$, a property that is not guaranteed in [7], particularly when the dynamics extend beyond the linear regime.*

4.2 Extension to higher dimensions

It is worth noting that the derivation (4.1)–(4.5) does not rely on any specific dimensionality. Hence, the operator construction (4.5) can be directly extended to higher-dimensional phase spaces, namely,

$$H[\delta f(t)](\mathbf{x}) = -\delta E[\delta f(t)](\mathbf{x}) + \gamma \int_{\mathbb{R}^d} \delta f(\mathbf{x},\mathbf{v},t)\nabla_\mathbf{v}\overline{f}(\mathbf{x},\mathbf{v})d\mathbf{v},$$

$$\delta E[\delta f(t)](\mathbf{x}) = \nabla\Delta^{-1}\delta\rho(\mathbf{x},t). \tag{4.7}$$

(a) L^2-perturbation

(b) Electric field
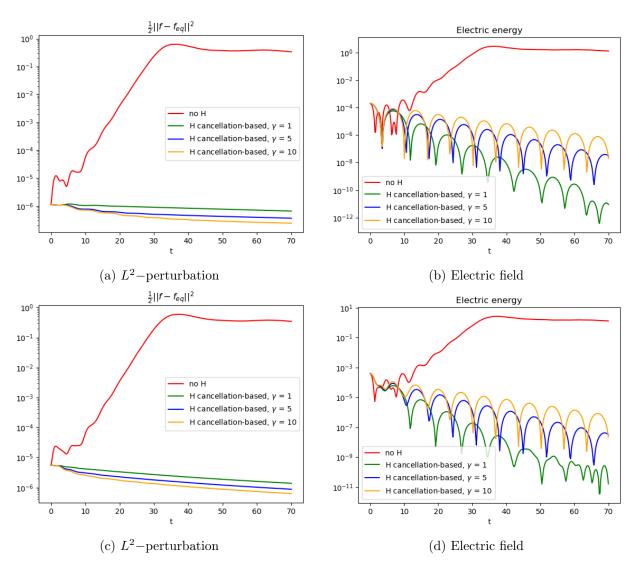

(c) L^2-perturbation

(d) Electric field

Figure 4.1: Two-stream instability. Solutions generated by the cancellation-based controller (4.5) under different initial data. Solutions in the first row are generated with $f_0 = (1+0.001\cos(\frac{1}{5}x))\overline{f}(v)$. Solutions in the second row are generated with $f_0 = (1-0.001\sin(\frac{1}{5}x)+0.002\cos(\frac{2}{5}x))\overline{f}(v)$.

The only added complexity arises from the numerical evaluation of δE and velocity integrals. This universality makes the cancellation-based feedback strategy particularly appealing for plasma control problems beyond the one-dimensional setting.

In Figures 4.3 – 4.4, we present the simulation for the two-stream instability in two dimensions, where the two-dimensional Vlasov-Poisson equations,

$$\begin{cases} \partial_t f(\mathbf{x}, \mathbf{v}, t) + \mathbf{v} \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{v}, t) + (E(\mathbf{x}, t) + H(\mathbf{x}, t)) \nabla_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t) = 0, \\ \\ E(\mathbf{x}, t) = -\nabla \Phi(\mathbf{x}, t), \quad \nabla \cdot E(\mathbf{x}, t) = -\Delta \Phi(\mathbf{x}, t) = \rho(\mathbf{x}, t) - 1, \\ \\ \rho(\mathbf{x}, t) = \int_{\mathbb{R}^2} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}, \end{cases} \qquad (4.8)$$

are solved in the domain $\mathbf{x} = (x, y) \in [0, 10\pi]^2$, $\mathbf{v} = (v_1, v_2) \in [-8, 8]^2$. The equations are discretized using the semi-Lagrangian method with mesh grids $N_x = N_y = 70$, $N_{v_1} = N_{v_2} = 120$ and time step

17

(a) $\sigma = 2 \times 10^{-5}$ $(0.1 \cdot ||\delta f_0||_\infty)$

(b) $\sigma = 5 \times 10^{-5}$ $(0.25 \cdot ||\delta f_0||_\infty)$

(c) $\sigma = 1 \times 10^{-4}$ $(0.5 \cdot ||\delta f_0||_\infty)$

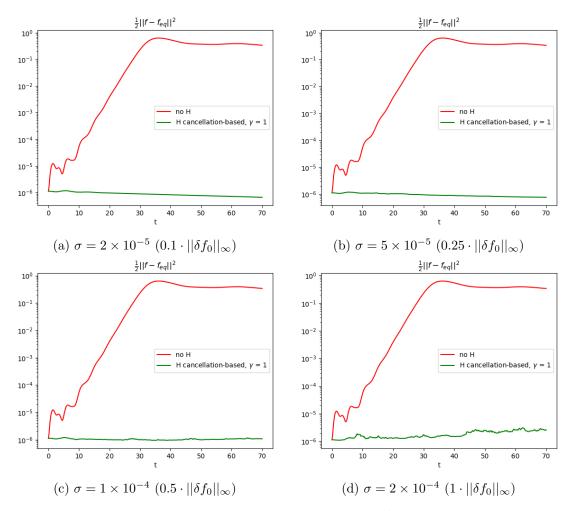(d) $\sigma = 2 \times 10^{-4}$ $(1 \cdot ||\delta f_0||_\infty)$

Figure 4.2: Two stream instability with noisy feedback. History of L^2-state perturbation over $t \in [0, 70]$. No external field (red lines) vs cancellation-based control with $\gamma = 1$ (green lines) under varying noise magnitudes σ.

$\Delta t = 0.15$. Periodic boundary conditions are applied at the four boundaries of the $xy-$region. The target equilibrium is set to

$$\overline{f}(\mathbf{v}) = \frac{1}{4\pi} \exp(-\frac{|\mathbf{v}-\overline{\mathbf{v}}|^2}{2}) + \frac{1}{4\pi} \exp(-\frac{|\mathbf{v}+\overline{\mathbf{v}}|^2}{2}), \quad \overline{\mathbf{v}} = (2, 2),$$

and the system is initialized with the perturbed data,

$$f_0(\mathbf{x}, \mathbf{v}) = \left(1 + \varepsilon \sin(\tfrac{x}{5}) \cos(\tfrac{y}{5})\right)\overline{f}(\mathbf{v}), \quad \varepsilon = 0.01.$$

The results demonstrate that in the absence of control, the perturbation grows rapidly, whereas under the control (4.7), the system is stabilized effectively in the 2D setting.

Remark 4.3 *We emphasize that the derivation of (4.5) (or (4.7)) does not rely on any assumption about $\overline{f}$ or on linearization of the system. The only requirement is access to $\nabla_{\mathbf{v}}\overline{f}(\mathbf{x}, \mathbf{v})$ which can be precomputed and stored. Although our experiments focus on a spatially invariant equilibrium $\overline{f}(\mathbf{v})$ to minimize numerical artifacts in perturbation growth, the method is, in principle, applicable to general anisotropic equilibria. Furthermore, the cancellation-based control can be regarded as a specific instance of the linear operator structure (2.5), whereas the latter—with learnable basis functions and integral kernels—offers substantially*
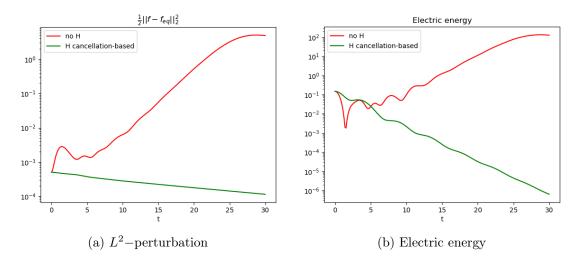
(a) L^2-perturbation

(b) Electric energy

Figure 4.3: 2-D two-stream instability. History of L^2-state perturbation, $\frac{1}{2}||\delta f(t)||_2^2$, and electric energy, $\int |E(\mathbf{x},t)|^2 d\mathbf{x}$, over $t \in [0,30]$. No external field (red lines) vs cancellation-based control (4.7) with $\gamma = 2$ (green lines).

greater flexibility. Consequently, we expect the linear LNO-based controller to generalize naturally to spatially inhomogeneous settings, provided the operator is trained with sufficient accuracy.

Remark 4.4 *Note that in multidimensional settings, the control field* (4.7) *is not a potential field and therefore cannot be directly realized as an electric field. For practical implementation, it may be necessary to combine multiple physical force fields - such as electric and magnetic fields — to approximate or achieve the desired cancellation-based control. Exploring such composite control mechanisms will be an interesting aspect for future investigation.*

5 Conclusion

In this paper, we develop a dynamic feedback control strategy for the Vlasov–Poisson system to address the challenge of long-time instability control in fusion energy applications. The core idea is to construct an operator that maps state perturbations to an external control field. We propose two approaches for constructing such an operator.

The first approach involves learning the operator using a neural network. Rather than relying on an off-the-shelf neural operator, which would impose a significant training burden, we draw inspiration from the linearized system and use optimal control theory to uncover the underlying structure of the ideal operator. This insight guides the design of a low-rank neural operator architecture. To train this model, we derive an adjoint-based gradient computation method. Compared to automatic differentiation, this adjoint approach is often more efficient and yields more accurate gradients, reducing the overall training cost. In the second approach, by explicitly analyzing the evolution equation satisfied by perturbations, we perform a direct energy estimate and propose a cancellation-based control strategy that eliminates the destabilizing component of the electric field. This leads to a novel closed-form operator, completely removing the need for training. The resulting operator is highly robust, even under noisy feedback.

Several interesting directions lie ahead. One concerns the theoretical understanding of the cancellation-based control defined in (4.5). While the decay estimate (4.4) suggests that the perturbation cannot be completely eliminated, since components orthogonal to $\partial_v \overline{f}$ are not directly controlled, numerical experiments nonetheless demonstrate effective overall decay. This raises the intriguing question of whether there is an underlying structure yet to be uncovered that explains the effectiveness of this operator. In
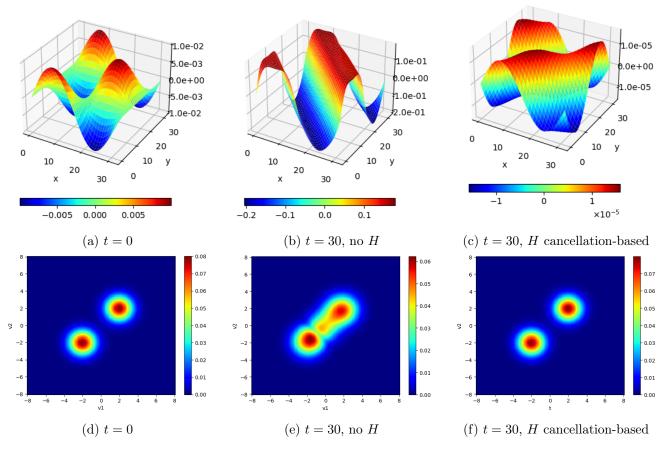
Figure 4.4: 2-D two-stream instability. The first row displays surface plots of the density perturbation $\delta\rho$. The different ranges of the color bars should be noted. The second row displays the contours of velocity distribution $f(6\pi, 5\pi, v_1, v_2, t)$. The solution of the cancellation-based control (4.7) is obtained with $\gamma = 2$.

particular, it would be interesting to show that if the initial perturbation is orthogonal to $\partial_v \overline{f}$, then this orthogonal component still diminishes over time under the evolution governed by (4.1). Another promising direction involves the learning-based approach. To ensure long-term effectiveness of the learned operator, it is crucial to train it on a sufficiently rich set of perturbations, especially those likely to arise dynamically over time. This motivates the development of strategies for self-generating representative perturbations during training. Finally, an even more intriguing avenue is to extend the proposed strategy to the design of external magnetic fields. Since magnetic fields are often easier to tune in practice, this extension could significantly enhance the feasibility of real-world control implementations.

Acknowledgment

References

[1] Giacomo Albi, Giacomo Dimarco, Federica Ferrarese, and Lorenzo Pareschi. Instantaneous control strategies for magnetically confined fusion plasma. *Journal of Computational Physics*, page 113804, 2025.

[2] Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 workshop on integration of deep neural models and differential equations*, 2020.

[3] Ian Char, Joseph Abbate, László Bardóczi, Mark Boyer, Youngseog Chung, Rory Conlin, Keith Erickson, Viraj Mehta, Nathan Richner, Egemen Kolemen, et al. Offline model-based reinforcement learning for tokamak control. In *Learning for Dynamics and Control Conference*, pages 1357–1372. PMLR, 2023.

[4] Francis F Chen. *Introduction to plasma physics and controlled fusion*, volume 1. Springer, 2016.

[5] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

[6] S Dubbioso, G De Tommasi, Adriano Mele, G Tartaglione, M Ariola, and A Pironti. A deep reinforcement learning approach for vertical stabilization of tokamak plasmas. *Fusion Engineering and Design*, 194:113725, 2023.

[7] Lukas Einkemmer, Qin Li, Clément Mouhot, and Yukun Yue. Control of instability in a vlasov-poisson system through an external electric field. *arXiv preprint arXiv:2407.15008*, 2024.

[8] Lukas Einkemmer, Qin Li, Li Wang, and Yang Yunan. Suppressing instability in a vlasov–poisson system by an external electric field through constrained optimization. *Journal of Computational Physics*, 498:112662, 2024.

[9] Frank Ethridge and Leslie Greengard. A new fast-multipole accelerated poisson solver in two dimensions. *SIAM Journal on Scientific Computing*, 23(3):741–760, 2001.

[10] Christian Grossmann. *Numerical treatment of partial differential equations*. Springer, 2007.

[11] Martin Guerra, Qin Li, Yukun Yue, and Leonardo Zepeda-Núñez. What metric to optimize for suppressing instability in a vlasov-poisson system? *arXiv preprint arXiv:2504.10435*, 2025.

[12] Thomas Guillet and Romain Teyssier. A simple multigrid scheme for solving the poisson equation with arbitrary domain boundaries. *Journal of Computational Physics*, 230(12):4756–4771, 2011.

[13] Yan Guo and Walter A Strauss. Instability of periodic bgk equilibria. *Communications on Pure and Applied Mathematics*, 48(8):861–894, 1995.

[14] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. 1961.

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[16] Jinsu Kim and Jaemin Seo. Design optimization of nuclear fusion reactor through deep reinforcement learning. *arXiv preprint arXiv:2409.08231*, 2024.

[17] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

[18] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[19] Lev Davidovich Landau. On the vibrations of the electronic plasma. *Uspekhi Fizicheskikh Nauk*, 93(3):527–540, 1967.

[20] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

[21] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

[22] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[23] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020.

[24] Jingcheng Lu, Li Wang, and Jeff Calder. Controlling instability in the vlasov-poisson system through moment-based optimization. *arXiv preprint arXiv:2508.18412*, 2025.

[25] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[26] Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022.

[27] Clément Mouhot and Cédric Villani. On Landau damping. *Acta Mathematica*, 207(1):29–201, 2011.

[28] Majdi I Radaideh, Isaac Wolverton, Joshua Joseph, James J Tusar, Uuganbayar Otgonbaatar, Nicholas Roy, Benoit Forget, and Koroush Shirvan. Physics-informed reinforcement learning optimization of nuclear assembly design. *Nuclear Engineering and Design*, 372:110966, 2021.

[29] Vladimir Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of computational physics*, 60(2):187–207, 1985.

[30] Jaemin Seo, SangKyeun Kim, Azarakhsh Jalalvand, Rory Conlin, Andrew Rothstein, Joseph Abbate, Keith Erickson, Josiah Wai, Ricardo Shousha, and Egemen Kolemen. Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature*, 626(8000):746–751, 2024.

[31] Dmytro Sydorenko, Igor D Kaganovich, PLG Ventzek, and L Chen. Effect of collisions on the two-stream instability in a finite length plasma. *Physics of Plasmas*, 23(12), 2016.

[32] Brendan D Tracey, Andrea Michi, Yuri Chervonyi, Ian Davies, Cosmin Paduraru, Nevena Lazic, Federico Felici, Timo Ewalds, Craig Donner, Cristian Galperti, et al. Towards practical reinforcement learning for tokamak magnetic control. *Fusion Engineering and Design*, 200:114161, 2024.

[33] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.

[34] Jun Zhang. Fast and high accuracy multigrid solution of the three dimensional poisson equation. *Journal of Computational Physics*, 143(2):449–461, 1998.