Learning non-canonical Hamiltonian dynamics

for the guiding center problem

Clémentine Courtès[1,2] Emmanuel Franck[1,2] Michael Kraus[3]
Laurent Navoret[1,2] **Léopold Trémant**[4]

[1]IRMA, UMR 7501 Université de Strasbourg et CNRS
[2]Inria Nancy-Grand Est, MACARON Project (Strasbourg, France)
[3]Max-Planck-Institut für Plasmaphysik (Garching, Germany)
[4]Laboratoire de Mathématiques de Lens, Université d'Artois (Lens, France)

Numkin

06/11/2024

Outline

Outline

What are Hamiltonian dynamics?

Derive the dynamics on the *position & momentum* from a Hamiltonian $(q, p) \mapsto H(q, p) \in \mathbb{R}$,

$$\dot{q} = \nabla_p H(q, p), \qquad \dot{p} = -\nabla_q H(q, p)$$

$$\underbrace{u = \begin{bmatrix} q \\ p \end{bmatrix}}_{\text{coordinates}}, \qquad \underbrace{\dot{u} = J^{-1} \nabla H(u)}_{\substack{\text{Hamiltonian} \\ \text{dynamics}}}, \qquad \underbrace{J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}}_{\text{symplectic form}}$$

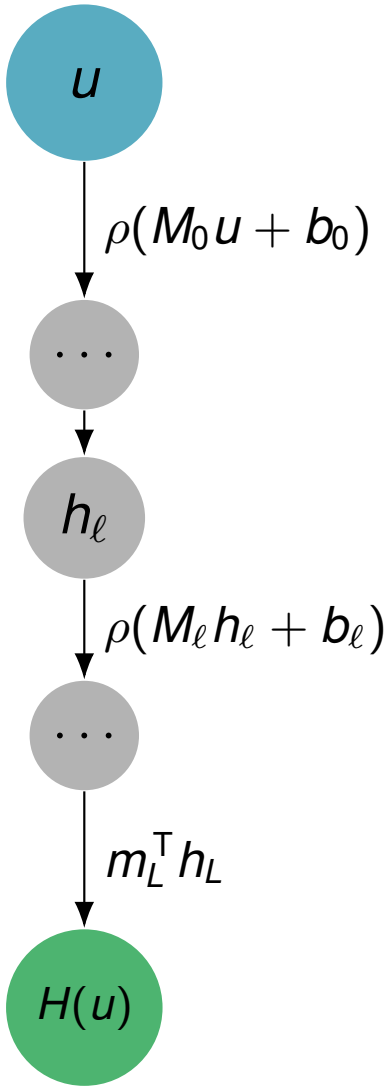
Non-canonical problems

- the symplectic form derives from a potential

  Lotka-Volterra | magnetic field lines | guiding center

Learn a symplectic potential
and a Hamiltonian with a neural network

[Greydanus, Dzamba, and Yosinski 2019]
[Chen, Matsubara, and Yaguchi 2021]



Lagrangian formulation

$$\mathcal{L}(z, z_t)$$

extremise the action

$$\frac{\delta}{\delta z}\left[\int_{t_0}^{t_1} \mathcal{L}\left(z(t), \dot{z}(t)\right) \mathrm{d}t\right] = 0$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\frac{\partial \mathcal{L}}{\partial z_t}(z, \dot{z})\right] = \frac{\partial \mathcal{L}}{\partial z}(z, \dot{z})$$

degeneracy condition

$$\frac{\partial^2 \mathcal{L}}{\partial z_t^i \partial z_t^j} = 0$$

$$\sum_{j=1}^{2d} \frac{\partial^2 \mathcal{L}}{\partial z_t^i \partial z^j} \dot{z}^j = \frac{\partial \mathcal{L}}{\partial z^i}$$

The canonical case

$$\mathcal{L}(q, p, q_t) = p \cdot q_t - H(q, p)$$

Note: with $H(q, p) = \frac{1}{2}p^2 + V(q)$,

$$\mathcal{L} = \frac{1}{2}p^2 - V(q) = K(p) - V(q).$$

The non-canonical case

$$\mathcal{L}(z, z_t) = A(z) \cdot z_t - H(z)$$

yields a symplectic form

$$W(z) = D_z A(z)^{\mathsf{T}} - D_z A(z)$$

and dynamics

$$\dot{z} = W(z)^{-1} \nabla H(z).$$

The Lotka-Volterra problem

Dynamics

$$\dot{x} = x - xy, \qquad \dot{y} = -y + xy.$$

Lagrangian structure

$$A(x,y) = \begin{pmatrix} -\ln(y)/x \\ 0 \end{pmatrix}, \qquad H(x,y) = x + y - \ln(xy).$$

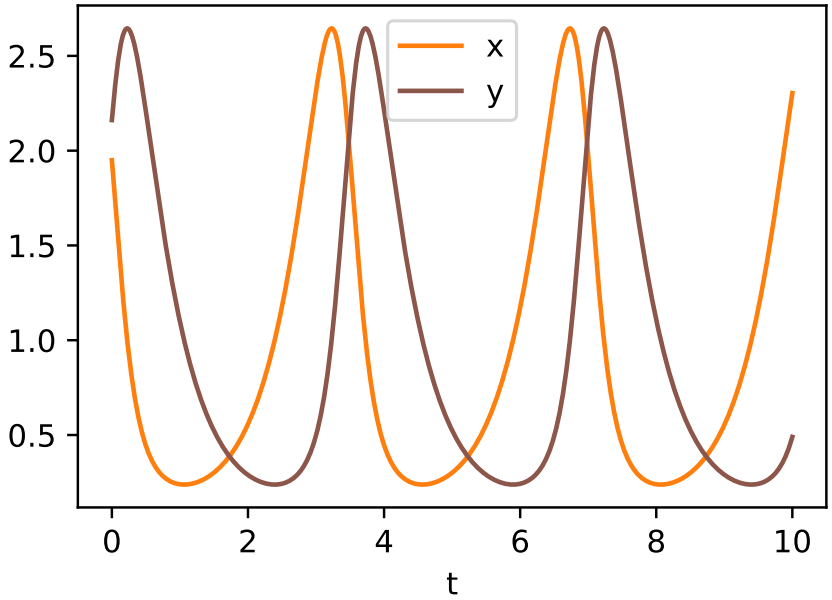


Figure: Time evolution

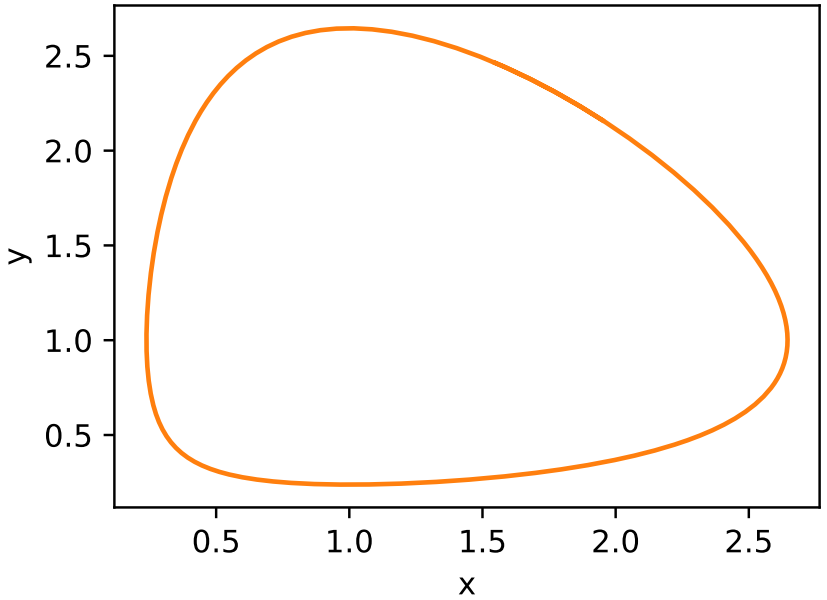


Figure: Phase-space evolution

Guiding center - The equations

Asymptotic model in tokamaks with a strong magnetic field

- poloidal-toroidal coordinates $X = (r, \theta, \phi)$
- momentum is just u in the toroidal direction
- magnetic potential $A = (0, A_\theta, A_\phi)$, from which $\mathbf{B} = \nabla \times A$
- magnetic field unit vector $b = R_0 + r\cos(\theta)$, here $R_0 = 1$
- assume axisymmetry (invariance of dynamics w.r.t. toroidal angle ϕ)

Choose a "nice" Lagrangian e.g. [Qin, Guan, and W. M. Tang 2009]

$$\mathcal{L}(r, \theta, \phi, u, \theta_t, \phi_t) = A_\theta(r, \theta)\theta_t + \big(A_\phi(r) + ub(r, \theta)\big)\phi_t - H(r, \theta, u).$$

Guiding center - Interesting (for us) trajectories

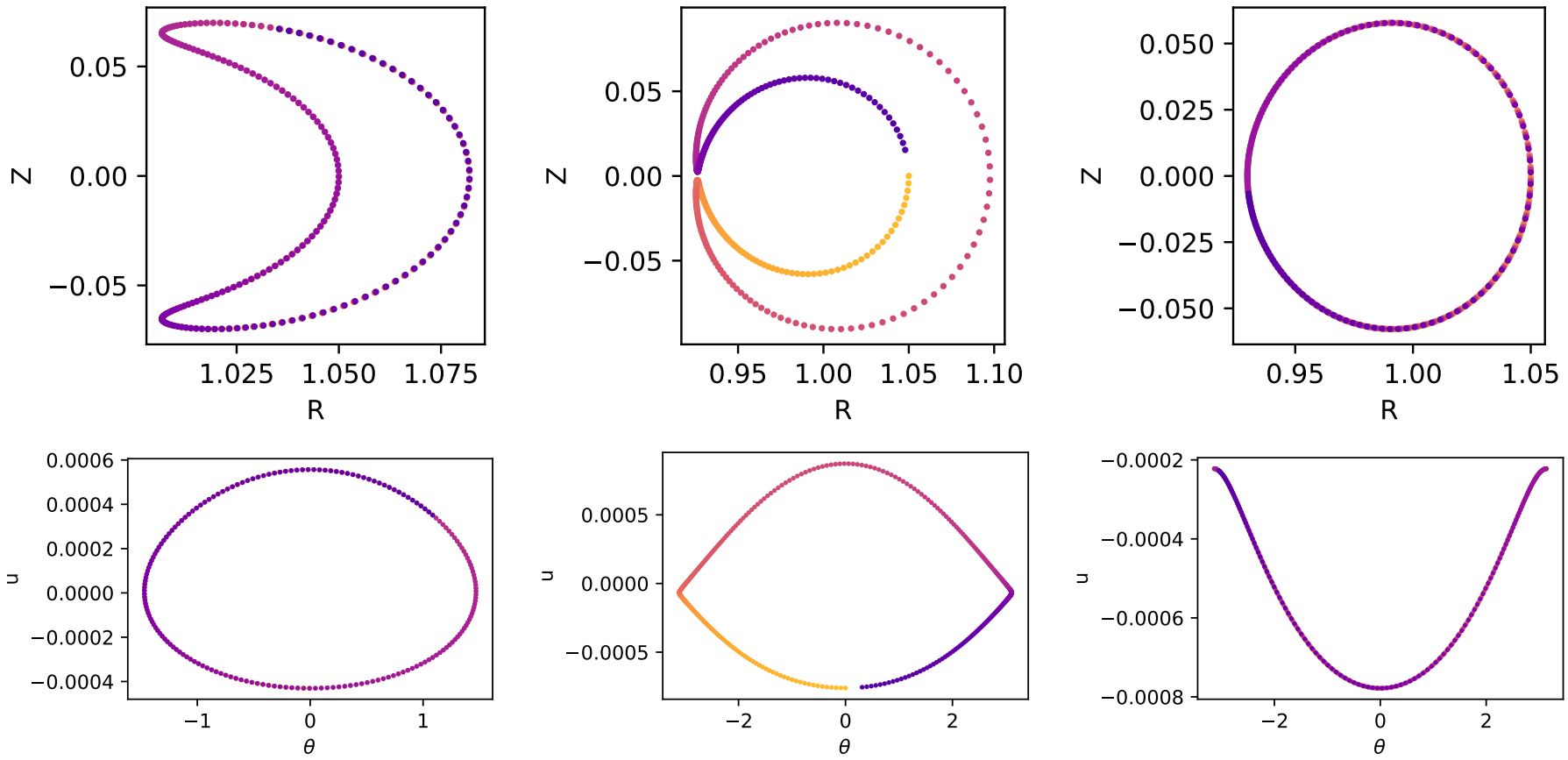
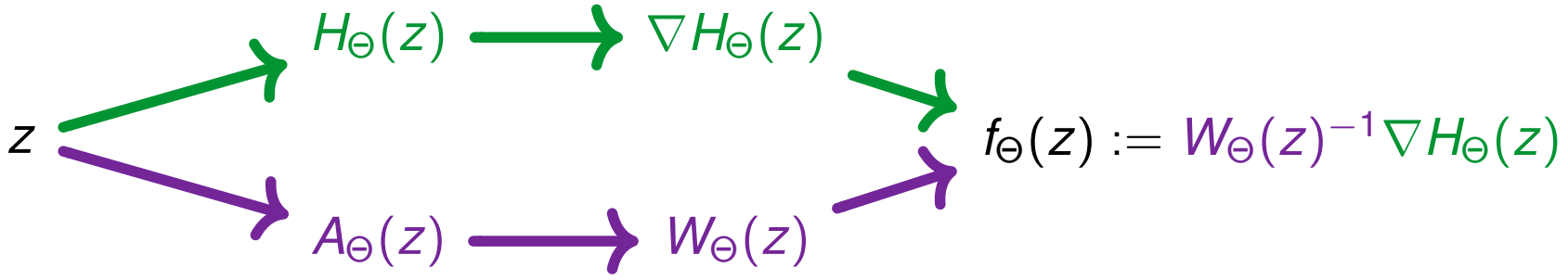


Figure: Different trajectories in the R-Z and θ-u planes, with initial conditions $r_0 = 0.05$, $\theta_0 = 0$, $\phi_0 = 0$, $\mu = 2.25 \times 10^{-6}$ and different initial velocities, generating different qualitative dynamics. In reading order, deeply trapped $u_0 = -4.306 \times 10^{-4}$, barely trapped $u_0 = -7.61 \times 10^{-4}$, barely passing $u_0 = -7.782 \times 10^{-4}$.

Learning procedure [Chen, Matsubara, and Yaguchi 2021]



Dataset

$$\{(z, f(z)),\ z \text{ sampled randomly}\}$$

Loss function

$$\mathcal{C}(\Theta) = \mathbb{E}_z\left[\|f_\Theta(z) - f(z)\|_{\mathrm{vf}}^2\right]$$

with the natural Gram norm,

$$\|\delta\|_{\mathrm{vf}}^2 = \delta^{\mathsf{T}} \left(\mathbb{E}_z\left[f(z)^{\mathsf{T}} f(z)\right]\right)^{-1} \delta$$

⚠ No gauge fixing

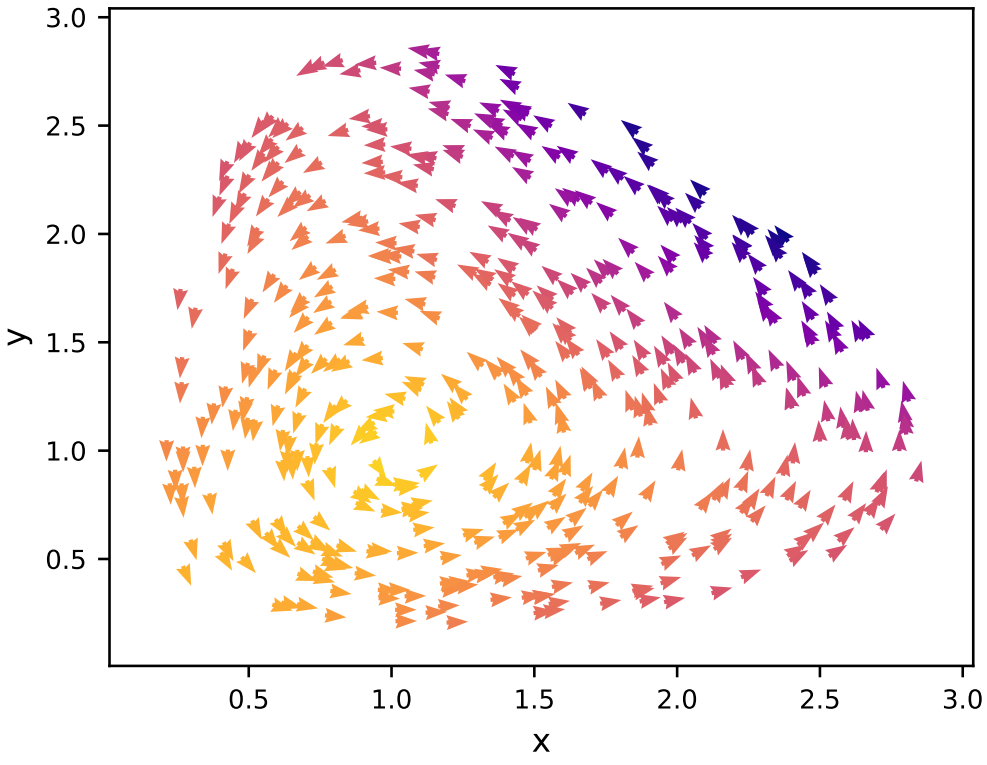


Figure: Sample of the dataset for Lotka-Volterra

The importance of structure – Lotka-Volterra

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{pmatrix}0 & -xy\\ xy & 0\end{pmatrix}\begin{pmatrix}1-1/x\\ 1-1/y\end{pmatrix}$$

fit the symplectic potential A and the Hamiltonian H on $\dot{x}, \dot{y}$

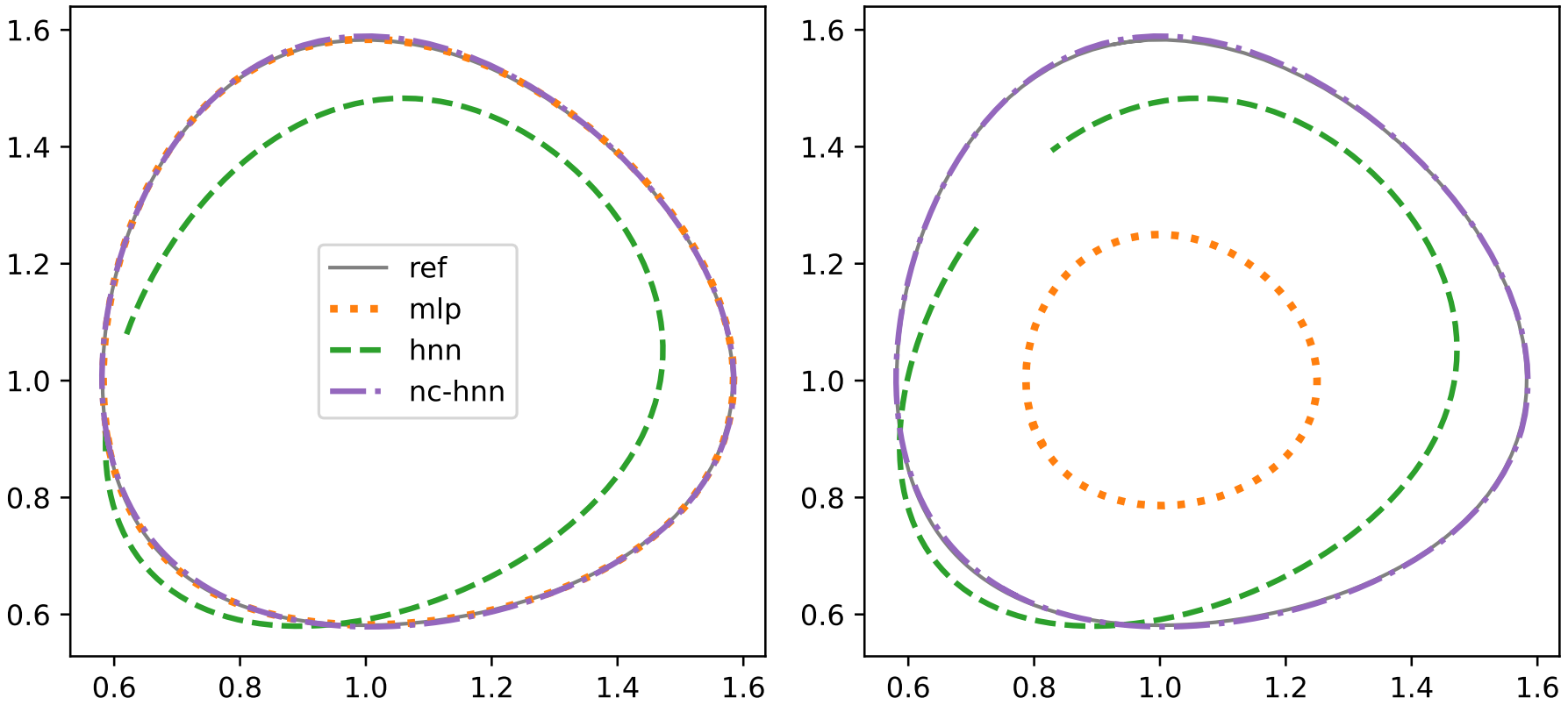


Figure: Sample trajectory in the validation set for (left) $t \in [0, 6.5]$ and (right) $t \in [793.5, 800]$, computed using DOP853 with tolerances $a_{\text{tol}} = r_{\text{tol}} = 10^{-10}$.

Application to the guiding center

Pre-processing axisymmetry & 2π-periodicity [Dong and Ni 2021] w.r.t. θ

$$z = \begin{pmatrix} r \\ \theta \\ \phi \\ u \end{pmatrix} \mapsto \begin{bmatrix} \frac{r - r_{\min}}{r_{\max} - r_{\min}} \\ \cos_{\odot}(\theta \mathbb{1} + \varphi^{[\Theta]}) \\ \frac{u - u_{\min}}{u_{\max} - u_{\min}} \end{bmatrix} \quad \begin{matrix} \mapsto H_\Theta \quad \sim \text{4k parameters} \\ \mapsto A_\Theta \quad \sim \text{4k parameters} \end{matrix}$$

Generating the dataset some "width" around $(r, \theta, \phi) = (0.05, 0, 0)$

$$r \in (0.04, 0.06), \qquad \theta \in (-\tfrac{1}{10}\pi, \tfrac{1}{10}\pi),$$
$$\phi \in (0, 2\pi), \qquad u \in (-8.5 \times 10^{-4}, -3 \times 10^{-4})$$

- 6k points sampled with a latin hypercube method
- simulate for 3 (banana —deeply trapped) periods
- 20 time-steps per period

The learnt on the guiding center

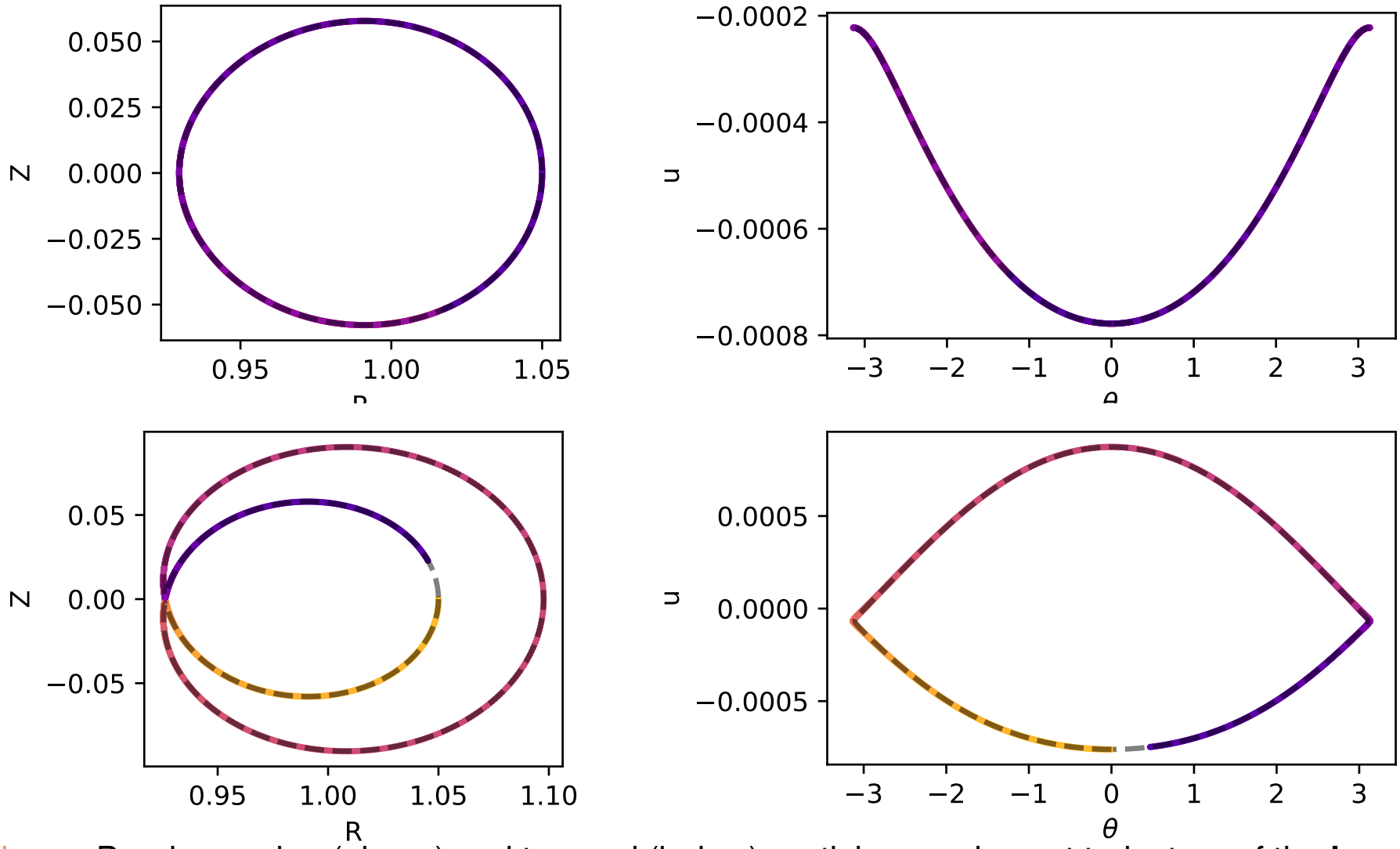


Figure: Barely passing (above) and trapped (below) particle, quasi-exact trajectory of the **learnt** model in the poloidal plane (left) and θ-u plane (right).

Outline

Standard vs geometric numerical schemes

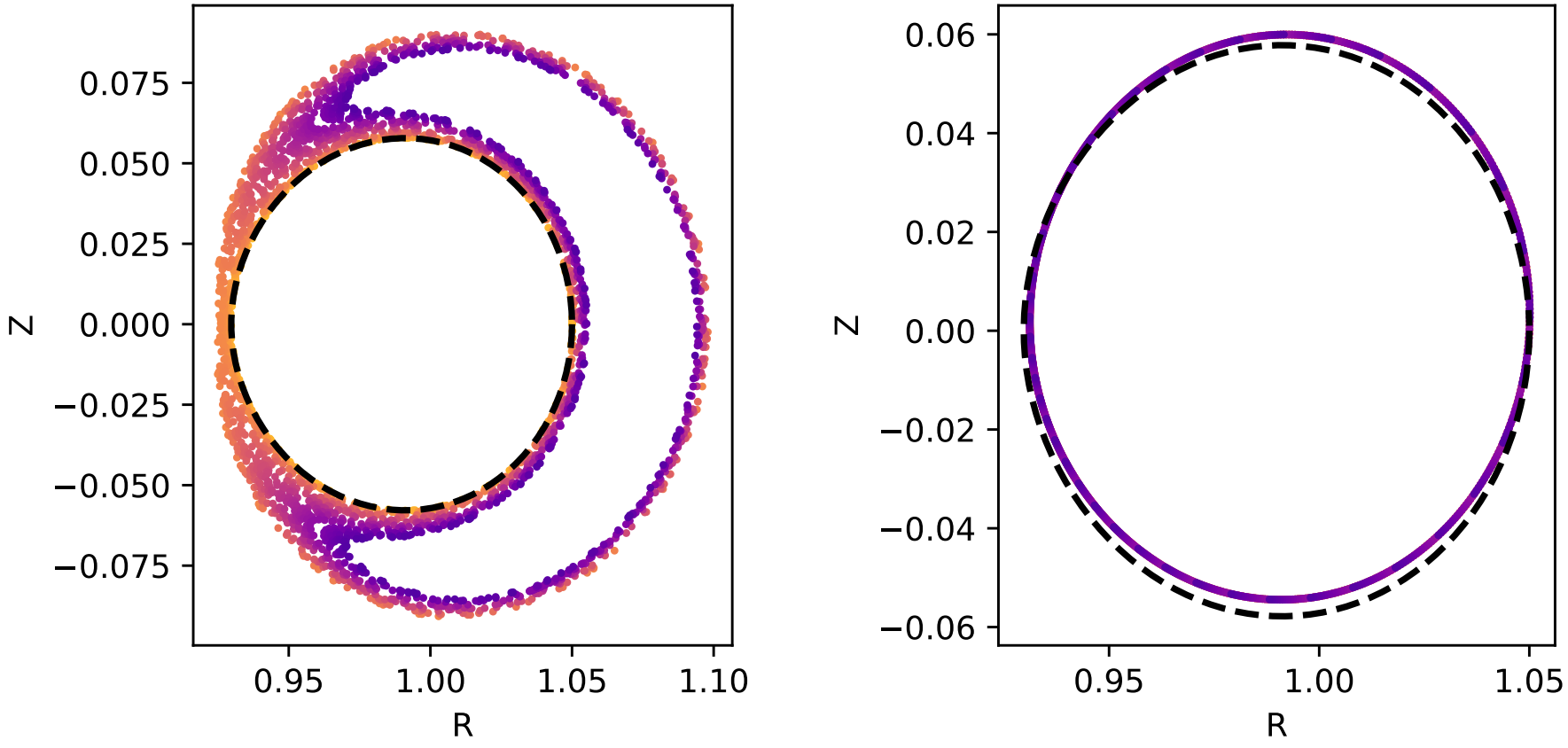


Figure: Barely passing particle, simulated with RK4 (left) and DVI (right), for 1300 periods with about 15 time-steps per period. Points are drawn every 11 time-steps.

Variational integrators (VI) in a nutshell

$\mathcal{L}(z, z_t)_{|z_t=\dot{z}}$

$L_{\Delta t}(z_n, z_{n+1}) = \mathcal{L}\left(z_{n+1}, \dfrac{z_{n+1} - z_n}{\Delta t}\right)$

Extremise the (discrete) action

$\dfrac{\delta}{\delta z}\left[\int \mathcal{L}\left(z(t), \dot{z}(t)\right) \mathrm{d}t\right] = 0$

$\dfrac{\partial}{\partial z_n}\left[\sum_k L_{\Delta t}(z_k, z_{k+1})\right] = 0$

(Discrete) Euler-Lagrange equations

$\dfrac{\mathrm{d}}{\mathrm{d}t}\left[\dfrac{\partial \mathcal{L}}{\partial z_t}(z, \dot{z})\right] = \dfrac{\partial \mathcal{L}}{\partial z}(z, \dot{z})$

$\dfrac{1}{\Delta t}\left[\dfrac{\partial \mathcal{L}_+}{\partial z_t} - \dfrac{\partial \mathcal{L}_-}{\partial z_t}\right] = \dfrac{\partial \mathcal{L}_-}{\partial z}$

Time-evolution

$\dot{z} = f(z) := W(z)^{-1} \nabla H(z)$

$S_{\Delta t}(z_n, z_{n+1}) = 0$

Degeneracy condition and ansatz [Ellison et al. 2018]

$\det(D_1 D_2 L_{\Delta t}) \neq 0 \Rightarrow$ parasitic oscillations

➔ need $\operatorname{rank}(D_1 D_2 L_{\Delta t}) = d$

Gauge ansatz

$$z = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad A(z) = \begin{bmatrix} \vartheta(x,y) \\ 0 \end{bmatrix}$$

$$\mathcal{L}(x, y, x_t) = \vartheta_j(x,y) x_t^j - H(x,y)$$

which generates the symplectic form

$$W(z) = \begin{bmatrix} D_x\vartheta^\mathsf{T} - D_x\vartheta & -D_y\vartheta \\ (D_y\vartheta)^\mathsf{T} & 0 \end{bmatrix}$$

Guiding center $x = (\theta, \phi)$, $y = (r, u)$

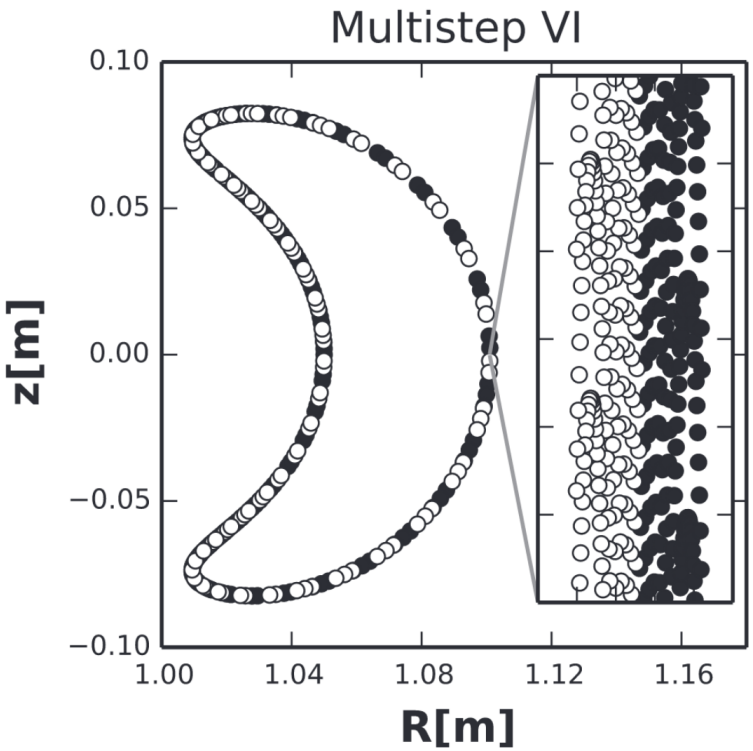


Figure: Result of a multistep VI, stolen from [Ellison et al. 2018, Fig. 5]. "Even-indexed points are marked in white and odd-indexed points in black."

⚠ For canonical problems, use the tautological potential [Vermeeren 2018]

An appropriate numerical scheme

The "appropriate" scheme we consider is from [Ellison et al. 2018],

$$\begin{cases} \vartheta_{n+1} = \vartheta_n + (D_x \vartheta_n)^\mathsf{T}(x_n - x_{n-1}) - \Delta t \nabla_x H_n \\ x_{n+1} = x_n + \Delta t \left(D_y \vartheta_{n+1}\right)^{-\mathsf{T}} \nabla_y H_{n+1} \end{cases}$$

where indices denote the time step of the arguments.

Notation setting $z_{n+1} = (x_{n+1}, y_{n+1})$, we write the scheme as

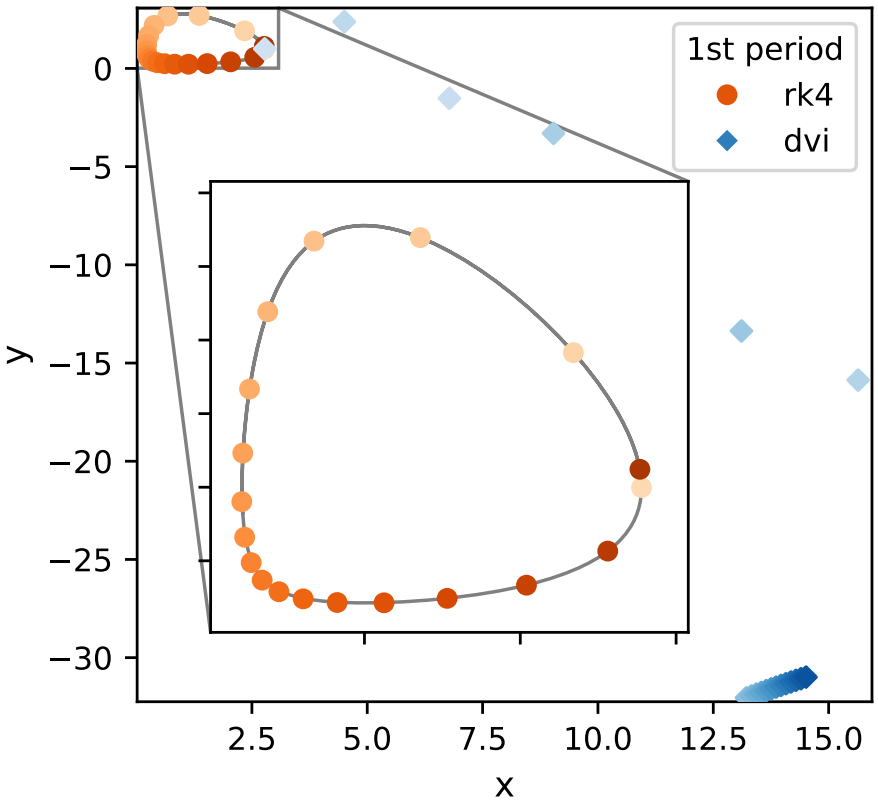
$$\underset{z_{n+1}}{\text{solve}} \quad S_{\Delta t}(z_n, z_{n+1}) = 0, \qquad \text{or} \qquad z_{n+1} = \Phi_{\Delta t}(z_n).$$

- initial guess with Hamlicit Euler step, then Newton iterations.

> The scheme is sensitive to perturbations on ϑ, even if they don't impact the z-dynamics!

Application to learnt problems

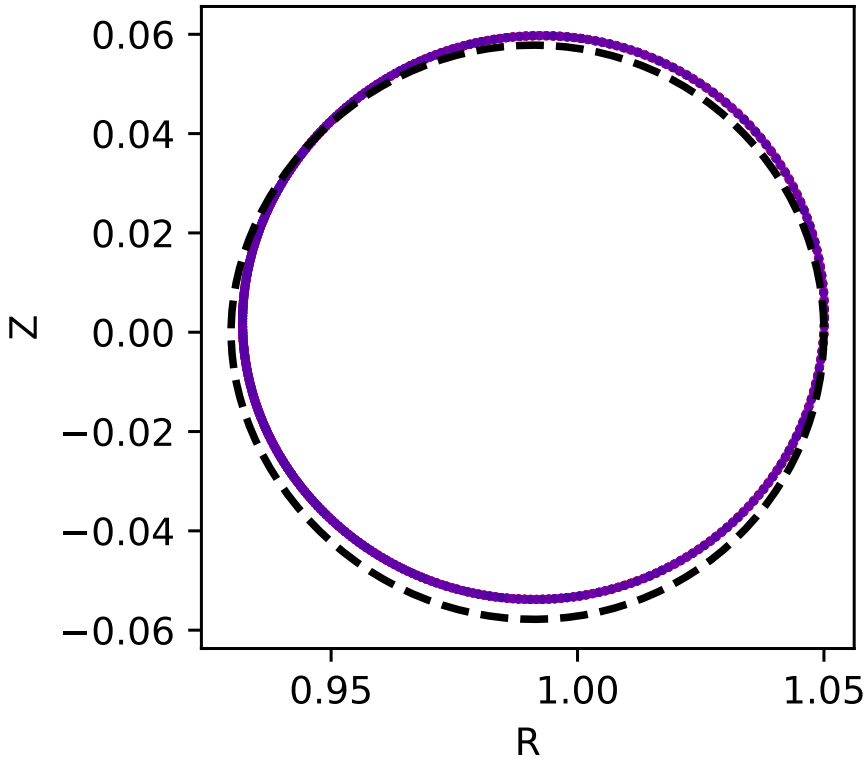
Lotka-Volterra



Diverges after just 1 iteration

Why does this fail?

Guiding center (barely passing)



Behaves like the reference model

Why does this work?

Sensitivity to perturbations

Introduce a perturbation

$$\vartheta \leftarrow \vartheta + \cos(3x)$$

- no impact on the continuous dynamics
- makes $t \mapsto \vartheta\big(x(t), y(t)\big)$ stiff and causes larger numerical errors
- such “invisible” perturbations appear during learning

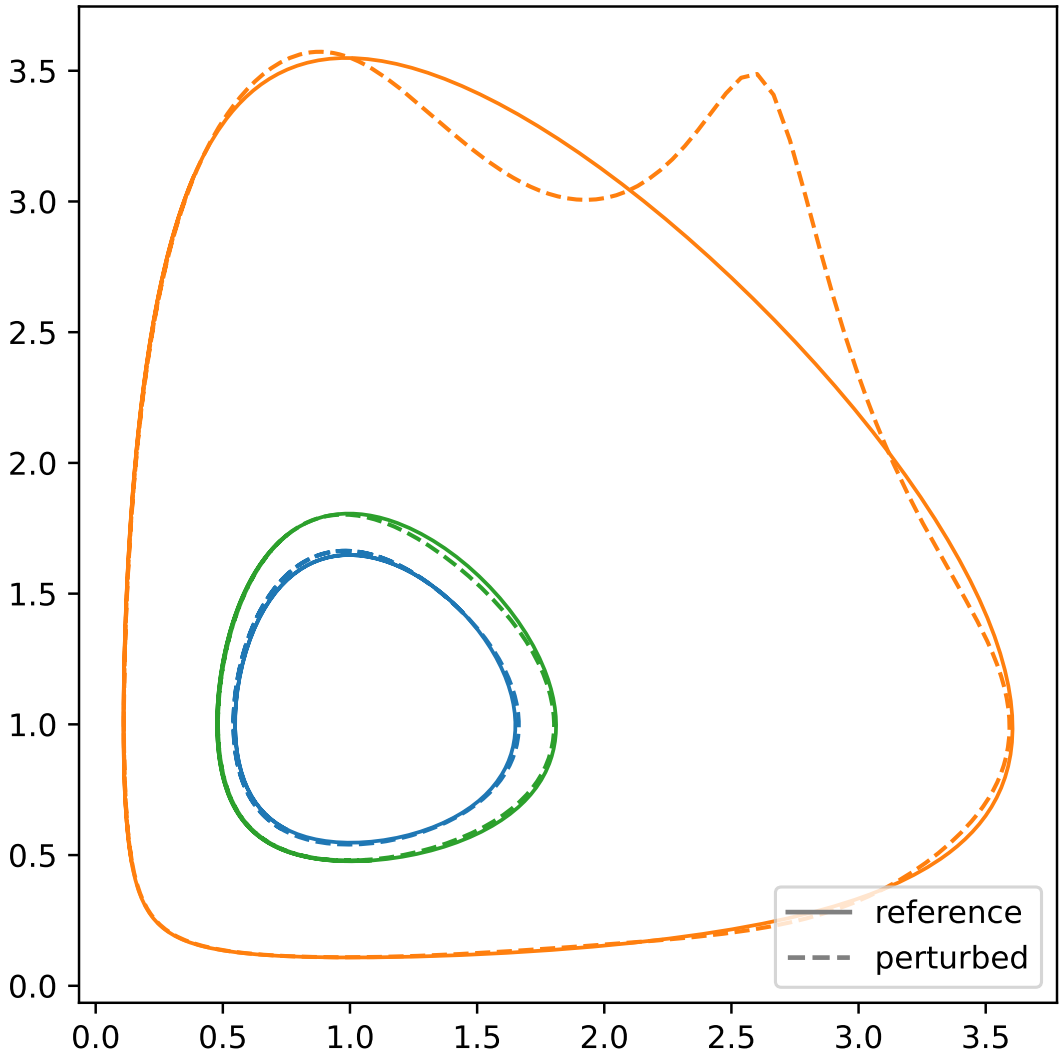


Figure: Sample trajectories computed with a time-step $\Delta t = 0.01$.

Regularised learning procedure

Dominant error term which we denote $z \mapsto r(z)$,

$$z_{\Delta t} = \varphi_{\Delta t}(z) + \frac{\Delta t^2}{2} r(z) + \mathcal{O}(\Delta t^3)$$

with $z_{\Delta t}$ s.t. $S_{\Delta t}(z, z_{\Delta t}) = 0$.

This error involves second-derivatives of ϑ, H, but most importantly

it involves $D_x\vartheta + D_x\vartheta^{\mathsf{T}}$,
it is clearly **gauge-dependent**

Modify the norm for numerics

$$\mathcal{L}_{\mathrm{vf}}(\Theta) = \mathbb{E}_z\left[\|f_\Theta(z) - f(z)\|_{\mathrm{vf}}^2\right] + \varepsilon\, \mathbb{E}_z\left[\|r_\Theta(z)\|_{\mathrm{vf}}^2\right], \qquad \varepsilon = 10^{-4}.$$

Regularised learnt Lotka-Volterra and the DVI

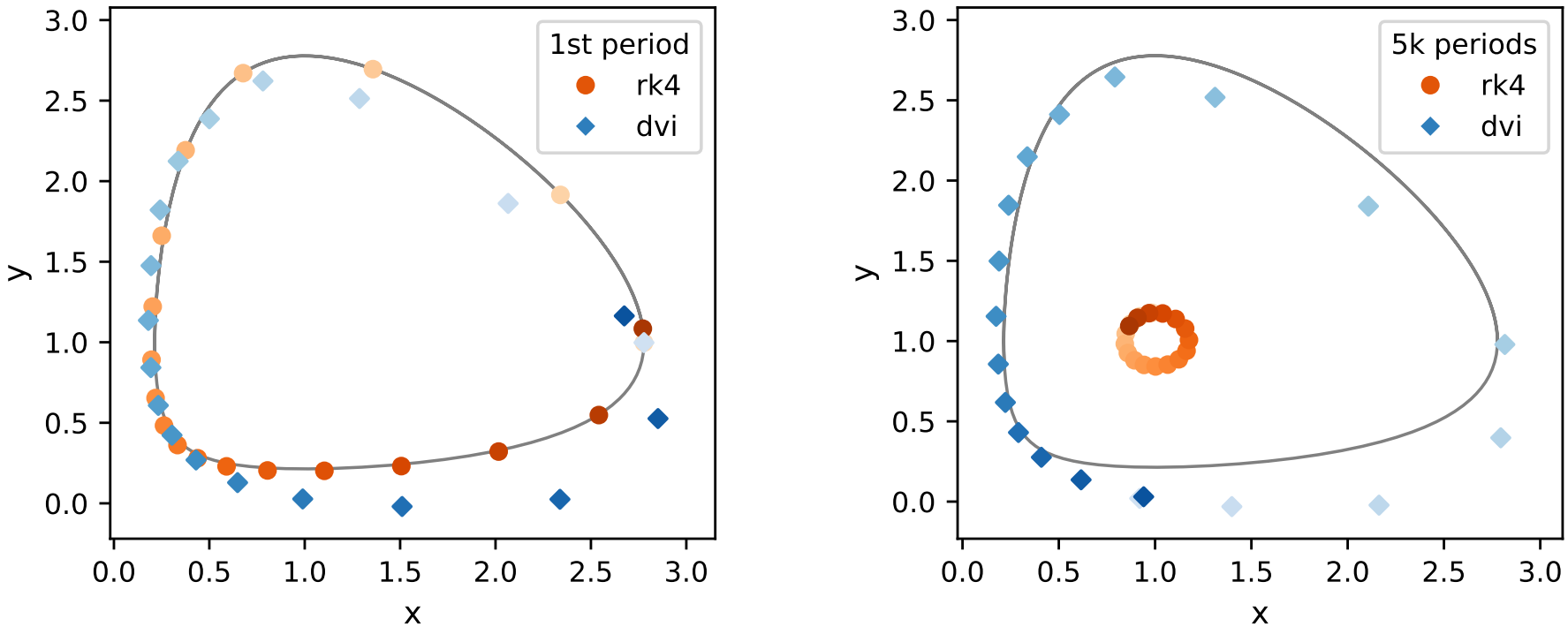


Figure: Simulation of the **learnt** dynamics fitted with the regularised loss, in short time (left) and long time (5k periods, right), using RK4 and the DVI, $\Delta t = 0.4$.

Outline

Why learn the exact vector field?

Since the end goal is simulation, we could instead learn the mapping

$$z(0) \xrightarrow[t \mapsto \varphi_t]{\varphi_{\Delta t}} z(\Delta t) \xrightarrow[t \mapsto \varphi_t]{\varphi_{\Delta t}} z(2\Delta t)$$

Why learn $f(\,\cdot\,;\Theta) \approx f$ when we could learn $\Phi_{\Delta t}(\,\cdot\,;\Theta) \approx \varphi_{\Delta t}$?

Snapshot dataset

$$\Big\{ \big(z, \varphi_{\Delta t}(z), \varphi_{2\Delta t}(z)\big), \quad z \text{ sampled randomly} \Big\}$$

- high-precision short-time simulation
- Δt chosen at the limit of stability

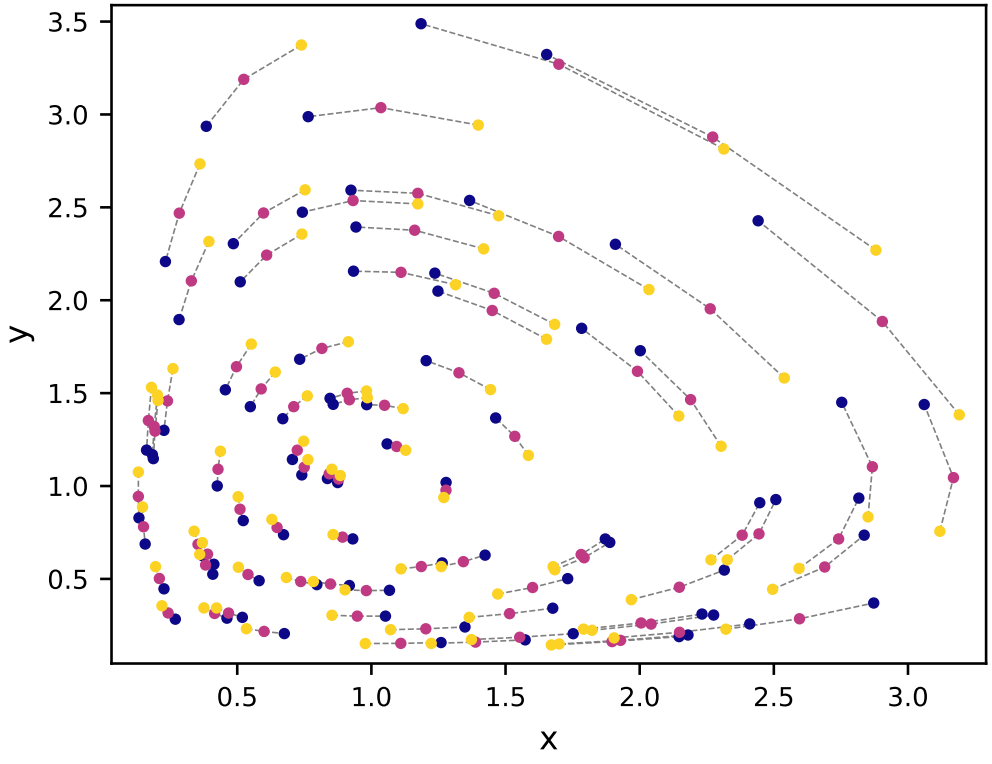


Figure: Sample of the snapshot dataset for LV

Strategies for **canonical** dynamics

The flow $(t, u) \mapsto \varphi_t(u)$ is *symplectic*, meaning that for all (t, u),

$$\left(D_u\varphi_t(u)\right)^\mathsf{T} J\, D_u\varphi_t(u) = J$$

1. **Impose it weakly** by adding a term to the loss.
   - 👎 Will drift over time.
2. **Use another architecture** e.g. a SympNet[Jin et al. 2020] or a HénonNet [Burby, Q. Tang, and Maulik 2021]

$$\varphi_{\Delta t}(u) = \varphi^{[L]} \circ \ldots \circ \varphi^{[1]}(u).$$

   - 👎 Not known for non-canonical systems
3. **Use a numerical integrator** i.e. learn a Hamiltonian such that the dynamics are recovered numerically, e.g. for the midpoint

$$\varphi_{\Delta t}(u) = u + \Delta t\, J^{-1} \nabla \widetilde{H}_{\Delta t}\left(\tfrac{1}{2}\left[u + \varphi_{\Delta t}(u)\right]\right)$$

[David and Méhats 2021]

Fit a numerical integrator?

Use a numerical integrator and learn the associated *modified* vector field, e.g.

$$\textit{Explicit Euler}: \quad \varphi_{\Delta t}(z) = z + \Delta t\, \widetilde{f}_{\Delta t}(z)$$

$$\textit{Implicit Euler}: \quad \varphi_{\Delta t}(z) = z + \Delta t\, \widetilde{f}_{\Delta t}\big(\varphi_{\Delta t}(z)\big)$$

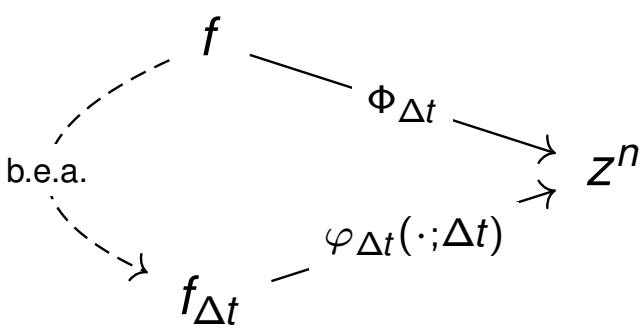
- Does this modified vector field exist?
- Can the structure be preserved?
- Can the original vector field be recovered?

These questions are the topic of *backward error analysis*!
[Hairer, Lubich, and Wanner 2006, Chap. IX]

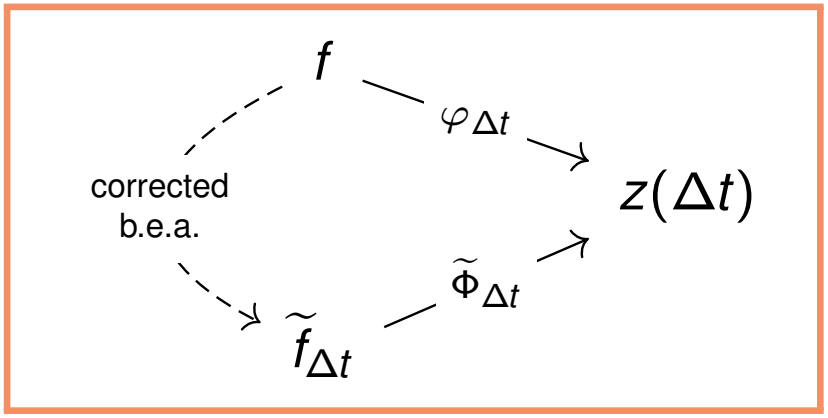
With NNs: Poli et al. 2020; Mathiesen, Yang, and Hu 2022; Bouchereau et al. 2023

Backward error analysis in the linear case

the numerical solution is the exact solution of a perturbed vf



the exact solution is the numerical solution of a modified vf



For a linear problem $\dot{z} = Mz$,

$$\left.\begin{aligned} \varphi_{\Delta t} &= e^{\Delta t\, M} \\ \widetilde{\Phi}_{\Delta t} &= \mathrm{id} + \Delta t\, \widetilde{M}_{\Delta t} \end{aligned}\right\} \quad \Longrightarrow \quad \widetilde{M}_{\Delta t} = \frac{1}{\Delta t}\left(e^{\Delta t\, M} - \mathrm{id}\right).$$

With non-linearities, some error is expected,

$$\|\varphi_{\Delta t} - \widetilde{\Phi}_{\Delta t}\| \lesssim e^{-1/\Delta t}$$

[Hairer, Lubich, and Wanner 2006, Thm IX.7.6]

Error convergence without structure

$$\text{err} = \left\| e^{TM} - \left(\text{id} + \delta t\, \widetilde{M}_{\Delta t}\right)^{T/\delta t} \right\|, \qquad M = J^{-1} I_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

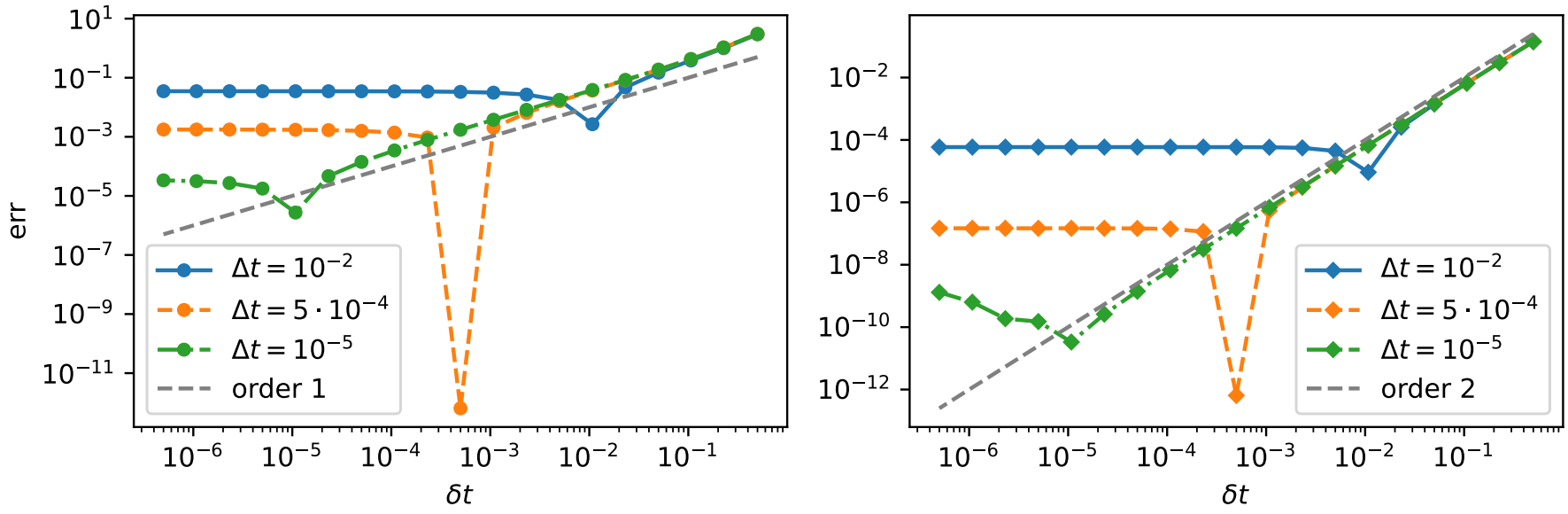


Figure: Error at a time $T = 5$ for different learning time-steps Δt and simulation time-steps δt using the explicit Euler method (left) and the midpoint method (right).

simulate with the same time-step as for learning

Error convergence with structure

$$\text{err} = \left\| e^{TM} - \left(\text{id} + \delta t\, J^{-1} \widetilde{H}_{\Delta t}\right)^{T/\delta t} \right\|, \qquad \widetilde{H}_{\Delta t} = \frac{1}{2}\left(J\widetilde{M}_{\Delta t} + (J\widetilde{M}_{\Delta t})^{\mathsf{T}}\right)$$

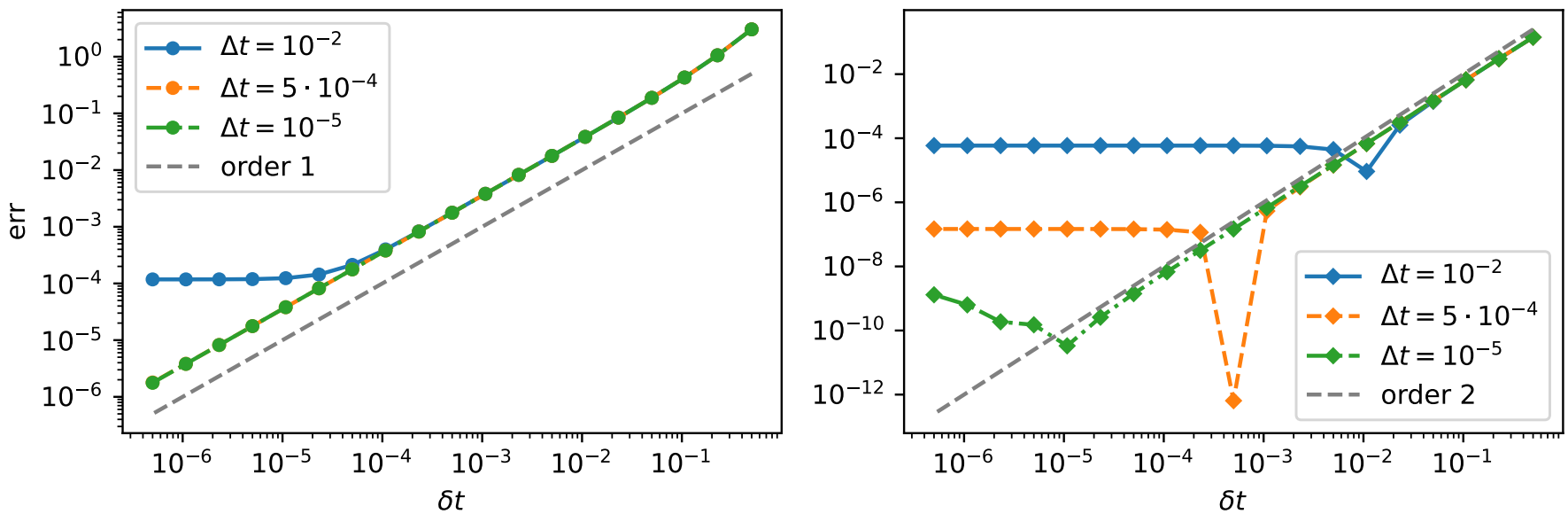


Figure: Error at a time $T = 5$ for different learning time-steps Δt and simulation time-steps δt using the explicit Euler method (left) and the midpoint method (right).

use a scheme appropriate for the structure

Scheme learning the DVI

Training phase minimize Newton residual: setting $J_{\Delta t} = D_2 S_{\Delta t}\big(z, \varphi_{\Delta t}(z)\big)$,

$$\mathcal{C}(\Theta) = \mathbb{E}_z\left[\left\| J_{\Delta t}^{-1} S_{\Delta t}\big(z, \varphi_{\Delta t}(z)\big)\right\|_{\mathrm{sch}}^2\right] + \varepsilon \log(\kappa_{\mathrm{sch}}(J_{\Delta t}))$$

The vector and matrix norms are weighted with the Gram matrix

$$\mathbb{E}_z\left[(z - \varphi_{\Delta t}(z))^{\mathsf{T}}(z - \varphi_{\Delta t}(z))\right].$$

Other losses are possible, see e.g. [Offen and Ober-Blöbaum 2024], but these are not easily compatible with the Gram norm.

Performing simulations

$$\underset{z^{n+1}}{\text{solve}} \quad S_{\Delta t}(z^n, z^{n+1}) = 0$$

initial guess with a basic NN s.t. $\mathrm{NN}(z) \approx \varphi_{\Delta t}(z)$, then Newton steps.

Result on Lotka-Volterra

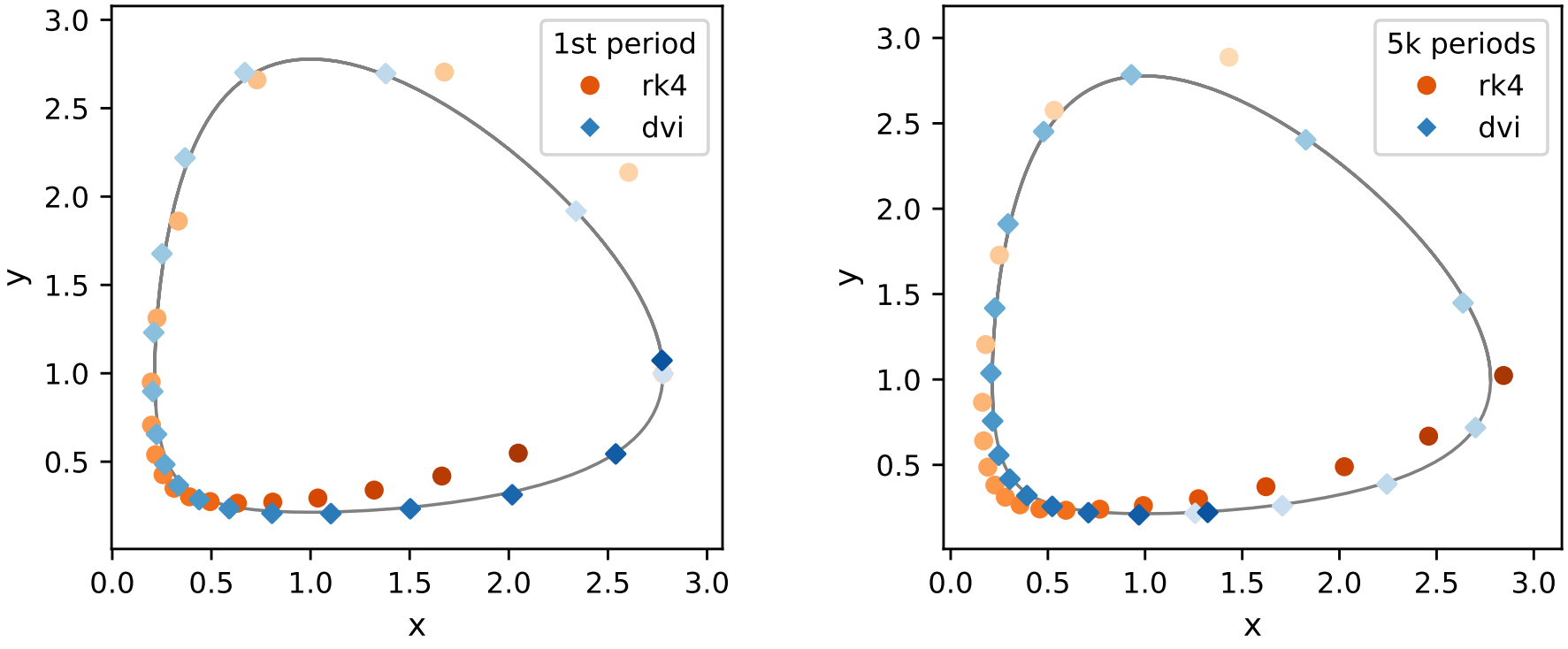


Figure: Simulation of the **scheme-learnt** dynamics in short time (left) and long time (5k periods, right), using RK4 and the DVI.

- the DVI is more accurate than the 4th-order method

Result on the guiding center – Barely passing

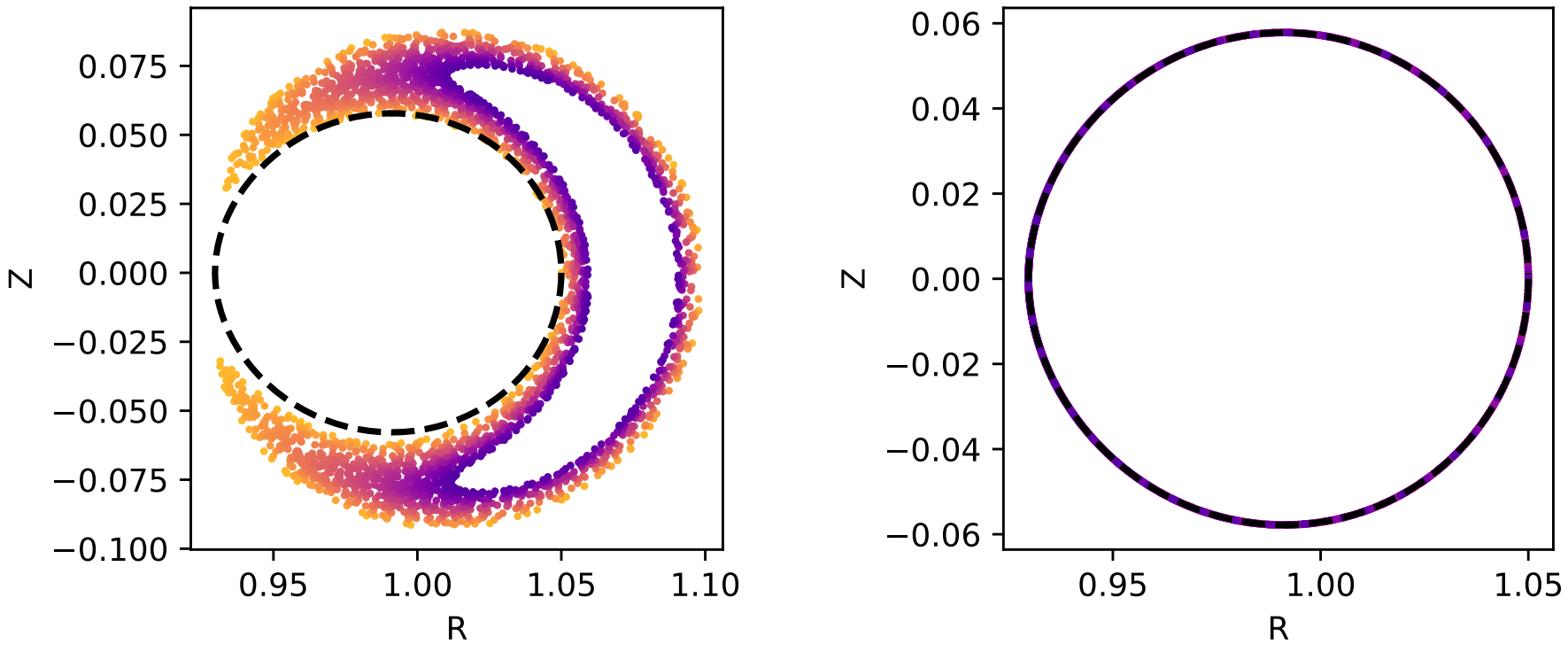


Figure: Barely passing particle, simulated using the **scheme-learnt** model with RK4 (left) and DVI (right), for 1300 periods with about 15 time-steps per period. Points are drawn every 11 time-steps.

- again, the error of the DVI compensates the error of the dynamics

Result on the guiding center – Energy evolution

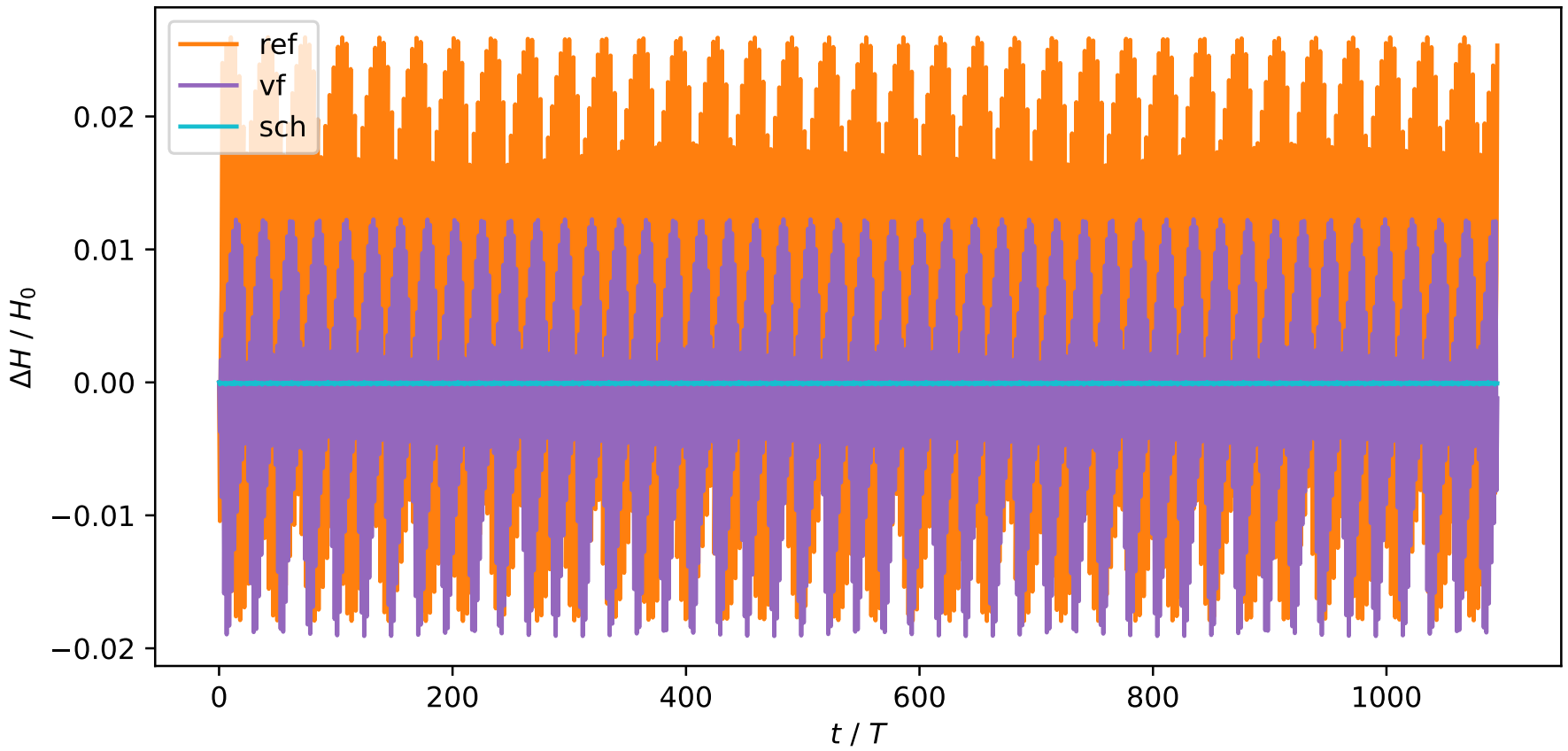


Figure: Relative energy evolution when applying the DVI on the different models (plotted every 11 time-steps) for the barely passing GC.

- with scheme-learning, the (relative) error is $\sim 10^{-4}$

Summary

Quick overview

- Hamiltonian dynamics and reminders on the guiding center
- the importance of encoding structure in learning
- two distinct learning strategies

Vector field learning

- simple approach
- accurate to physics
- adapt loss for numerics

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed (but large) time-step

What's left

- time-dependent problems, e.g. magnetic field lines
- is the "nice" split $z = (x, y)$ always possible?
- thorough backward error analysis of the DVI

Summary

Quick overview

- Hamiltonian dynamics and reminders on the guiding center
- the importance of encoding structure in learning
- two distinct learning strategies

Vector field learning

- simple approach
- accurate to physics
- adapt loss for numerics

Scheme learning

- goal-oriented loss
- long-time simulation
- fixed (but large) time-step

What's left

- time-dependent problems, e.g. magnetic field lines
- is the "nice" split $z = (x, y)$ always possible?
- thorough backward error analysis of the DVI

Thank you for you attention!

APPENDICES

References I

Bouchereau, Maxime et al. (Apr. 18, 2023). *Machine Learning Methods for Autonomous Ordinary Differential Equations*. arXiv: 2304.09036 [cs, math]. URL: http://arxiv.org/abs/2304.09036 (visited on 07/21/2023). Pre-published.

Burby, J W, Q Tang, and R Maulik (Feb. 1, 2021). "Fast Neural Poincaré Maps for Toroidal Magnetic Fields". In: *Plasma Physics and Controlled Fusion* 63.2, p. 024001. ISSN: 0741-3335, 1361-6587.

Chen, Yuhan, Takashi Matsubara, and Takaharu Yaguchi (2021). "Neural Symplectic Form: Learning Hamiltonian Equations on General Coordinate Systems". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 16659–16670.

David, Marco and Florian Méhats (June 22, 2021). *Symplectic Learning for Hamiltonian Neural Networks*. arXiv: 2106.11753 [cs, math]. Pre-published.

Dong, Suchuan and Naxian Ni (June 15, 2021). "A Method for Representing Periodic Functions and Enforcing Exactly Periodic Boundary Conditions with Deep Neural Networks". In: *Journal of Computational Physics* 435, p. 110242. ISSN: 0021-9991.

Ellison, C. L. et al. (May 4, 2018). "Degenerate Variational Integrators for Magnetic Field Line Flow and Guiding Center Trajectories". In: *Physics of Plasmas* 25.5, p. 052502. ISSN: 1070-664X.

References II

Gnanasambandam, Raghav et al. (Apr. 29, 2022). *Self-Scalable Tanh (Stan): Faster Convergence and Better Generalization in Physics-informed Neural Networks*. arXiv: 2204.12589. Pre-published.

Greydanus, Samuel, Misko Dzamba, and Jason Yosinski (2019). "Hamiltonian Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Hairer, Ernst, Christian Lubich, and Gerhard Wanner (2006). *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. 2nd ed. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer-Verlag. ISBN: 978-3-540-30663-4.

Jin, Pengzhan et al. (Dec. 2020). "SympNets: Intrinsic Structure-Preserving Symplectic Networks for Identifying Hamiltonian Systems". In: *Neural Networks* 132, pp. 166–179. ISSN: 08936080.

Mathiesen, Frederik Baymler, Bin Yang, and Jilin Hu (June 28, 2022). "Hyperverlet: A Symplectic Hypersolver for Hamiltonian Systems". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.4 (4), pp. 4575–4582. ISSN: 2374-3468.

Offen, Christian and Sina Ober-Blöbaum (Jan. 8, 2024). "Learning of Discrete Models of Variational PDEs from Data". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34.1, p. 013104. ISSN: 1054-1500.

References III

Poli, Michael et al. (2020). “Hypersolvers: Toward Fast Continuous-Depth Models”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 21105–21117.

Qin, Hong, Xiaoyin Guan, and William M. Tang (Apr. 21, 2009). “Variational Symplectic Algorithm for Guiding Center Dynamics and Its Application in Tokamak Geometry”. In: *Physics of Plasmas* 16.4, p. 042510. ISSN: 1070-664X.

Vermeeren, Mats (July 27, 2018). *Modified Equations for Variational Integrators Applied to Lagrangians Linear in Velocities*. arXiv: 1709.09567. Pre-published.

Outline

The guiding center equations

The magnetic potential $A = (0, A_\theta, A_\phi)$ is given by

$$A_\theta(r,\theta) = \frac{B_0 R_0^2}{\cos^2(\theta)} \left(\frac{r\cos(\theta)}{R_0} - \log\left(1 + \frac{r\cos(\theta)}{R_0}\right)\right), \qquad A_\phi(r) = -\frac{B_0 r^2}{2q_0}.$$

The parameters are $R_0 = 1$ the radial position of the magnetic axis; $B_0 = 1$ the magnitude of the magnetic field at R_0; $q_0 = 2$ the (dimensionless) safety factor, regarded as constant. To avoid division by zero, we use the integral formulation $A_\theta(r,\theta) = B_0 R_0 r^2 \int_0^1 \frac{t\,\mathrm{d}t}{R_0 + tr\cos(\theta)}$.

The Hamiltonian is a combination of kinetic and magnetic energies,

$$H(r,\theta,u) = \frac{1}{2}u^2 + \mu B(r,\theta), \qquad B(r,\theta) = \frac{B_0}{1 + \frac{r\cos(\theta)}{R_0}} \sqrt{1 + \left(\frac{r}{q_0 R_0}\right)^2},$$

where the additional parameter μ is the (constant) magnetic moment of the particle and $B = \|\mathbf{B}\|$ is the magnitude of the magnetic field.

Guiding center – Barely passing

Adapted from [Qin, Guan, and W. M. Tang 2009], with $\mu_{\text{ref}} = 2.448 \times 10^{-6}$ and $u_{\text{ref}}(0) = -8.117 \times 10^{-4}$. We wish to use $\mu = 2.25 \times 10^{-6}$, and therefore look for $u(0)$ to satisfy a proportionality rule:

$$H = \frac{\mu}{\mu_{\text{ref}}} H_{\text{ref}}, \quad \text{i.e.} \quad \frac{1}{2}u(0)^2 + \mu B(0) = \frac{\mu}{2\mu_{\text{ref}}} u_{\text{ref}}(0)^2 + \mu B_{\text{ref}}(0).$$

Since the magnetic field is independent of μ and u, it is the same on both sides, $B(0) = B_{\text{ref}}(0)$, therefore we find

$$u(0) = \sqrt{\frac{\mu}{\mu_{\text{ref}}}} u_{\text{ref}}(0).$$

This yields $u(0) = -7.782 \times 10^{-4}$.

DVI - The error term

Using the Einstein summation convention,

$$\begin{aligned} \vartheta_{j,\alpha}\, r^j &= -(\vartheta_{j,\alpha,k}\dot{x}^j - H_{,\alpha,k})\dot{z}^k, \\ \vartheta_{i,\beta}\, r^\beta &= -(\vartheta_{i,j} + \vartheta_{j,i})r^j - (\vartheta_{j,i,k}\dot{x}^j - H_{,i,k})\dot{z}^k. \end{aligned} \tag{1}$$

Remember that we use the Einstein summation convention, with summation indices j, β and k here: the first line of (1) would read

$$\sum_{j=1}^{d} \frac{\partial \vartheta_j}{\partial y^\alpha} r^j = -\sum_{j=1}^{d}\sum_{k=1}^{2d} \left(\frac{\partial^2 \vartheta_j}{\partial y^\alpha \partial z^k}\dot{x}^j - \frac{\partial^2 H}{\partial y^\alpha \partial z^k} \right) \dot{z}^k.$$

The indices i, j are reserved for the x-component, α, β for y and k for z.

When used in learning, we first compute $(r^j)_j$ using the vector-field training data for $\dot{x}^j, \dot{z}^k$ and auto-differentiate through the network for $\vartheta_{j,\alpha}, \vartheta_{j,\alpha,k}, H_{,\alpha,k}$. We then compute $(r^\beta)_\beta$ in the same way, using the known r^j.

Outline

The multi-layer perceptron (MLP)

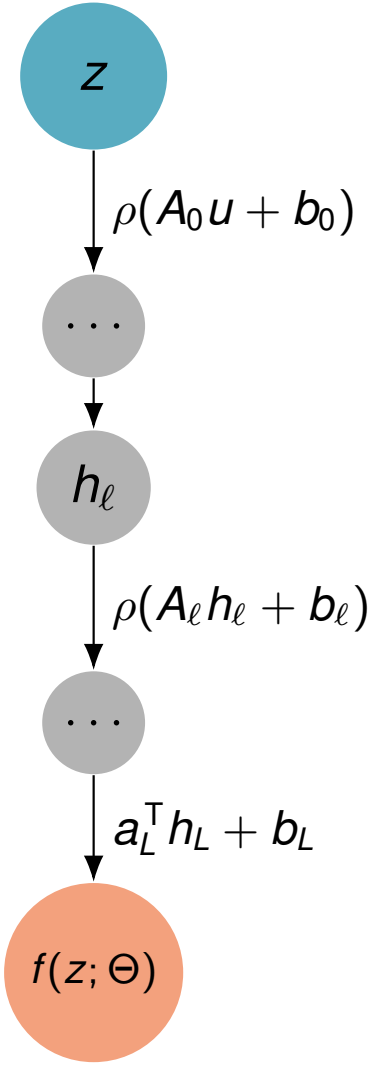
MLPs are a class of *parametric approximation functions* based on compositions.

$$f(z;\Theta) = \underbrace{\mathcal{F}_{\Theta_{L+1}} \circ \rho_{\mu_L} \circ \mathcal{F}_{\Theta_L} \circ \ldots \circ \underbrace{\rho_{\mu_1} \circ \mathcal{F}_{\Theta_1}(z)}_{h_1}}_{h_L}$$

- $\Theta = (\Theta_1, \mu_2, \Theta_2, \ldots, \mu_L, \Theta_{L+1})$ are the parameters;
- $h_1, \ldots, h_L$ are the "hidden variables"
- $\mathcal{F}_{\Theta_\ell}(h) = A_\ell h + b_\ell$ with $\Theta_\ell = (A_\ell, b_\ell)$;
- "self-scalable $\tanh$" activation from Gnanasambandam et al. 2022,

$$\rho_{\mu_\ell}(h) = \tanh_\odot(h) + \mu_\ell \odot h \odot \tanh_\odot(h),$$

multiplications and $\tanh$ applied componentwise.



The learning / fitting strategy

"Learning" consists in minimizing an objective function, the "loss"

$$\min_{\Theta} \mathcal{L}(\Theta) := \frac{1}{N} \sum_{b=1}^{N} d\left(f\left(z^{(b)}, \Theta\right),\ f\left(z^{(b)}\right) \right)$$

Some important precisions

1. The gradient w.r.t. Θ is computed using automatic differentiation;
2. This is done using stochastic gradient descent, i.e. *batches*;
   - the dataset $\left\{ \left(z^{(b)}, f(z^{(b)})\right) \right\}$ is too large otherwise
   - reduces overfitting, as a sort of annealing term
3. Sometimes the objective function is not the immediate output of the neural network.
   - fit the *differentials* of the output
   - penalisation on the parameters $\|\Theta\|$
   - regularisation using e.g. finite differences