Symplectic Neural Flows for Modelling and Discovery

Davide Murari

Department of Applied Mathematics and Theoretical Physics, University of Cambridge.

dm2011@cam.ac.uk

In collaboration with Priscilla Canizares, Carola-Bibiane Schönlieb, Ferdia Sherry, Zakhar Shumaylov



> The equations of motion of canonical Hamiltonian systems write

$$\begin{cases} \dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{1}$$

> The equations of motion of canonical Hamiltonian systems write

$$\begin{cases} \dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{1}$$

▶ Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), $\phi_{H,t}(\mathbf{x}_0) = \mathbf{x}(t)$, we have that

> The equations of motion of canonical Hamiltonian systems write

$$\begin{cases} \dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{1}$$

▶ Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), $\phi_{H,t}(\mathbf{x}_0) = \mathbf{x}(t)$, we have that

$$\frac{d}{dt} H(\phi_{H,t}(\mathbf{x}_0)) = \nabla H(\phi_{H,t}(\mathbf{x}_0))^\top \mathbb{J} \nabla H(\phi_{H,t}(\mathbf{x}_0)) = 0,$$

$$\left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right)^\top \mathbb{J} \left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right) = \mathbb{J},$$

> The equations of motion of canonical Hamiltonian systems write

$$\begin{cases} \dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x}) = X_H(\mathbf{x}) \in \mathbb{R}^{2n} \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}, \quad \mathbb{J} = \begin{bmatrix} 0 & I_n \\ -I_n & 0_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{1}$$

▶ Denoted with $\phi_{H,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ the exact flow of (1), $\phi_{H,t}(\mathbf{x}_0) = \mathbf{x}(t)$, we have that

$$\frac{d}{dt} H(\phi_{H,t}(\mathbf{x}_0)) = \nabla H(\phi_{H,t}(\mathbf{x}_0))^\top \mathbb{J} \nabla H(\phi_{H,t}(\mathbf{x}_0)) = 0,$$

$$\left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right)^\top \mathbb{J} \left(\frac{\partial \phi_{H,t}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right) = \mathbb{J},$$

• the flow preserves the canonical volume form of \mathbb{R}^{2n} .

Symplectic numerical methods

A one-step numerical method $\varphi^h : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is symplectic if and only if when applied to a Hamiltonian system the map φ^h is symplectic, i.e.,

$$\left(\frac{\partial \varphi^{h}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\top} \mathbb{J}\left(\frac{\partial \varphi^{h}(\mathbf{x})}{\partial \mathbf{x}}\right) = \mathbb{J}.$$

Symplectic numerical methods

A one-step numerical method $\varphi^h : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is symplectic if and only if when applied to a Hamiltonian system the map φ^h is symplectic, i.e.,

$$\left(rac{\partial arphi^h(\mathbf{x})}{\partial \mathbf{x}}
ight)^ op \mathbb{J}\left(rac{\partial arphi^h(\mathbf{x})}{\partial \mathbf{x}}
ight) = \mathbb{J}.$$

Symplectic and energy preserving methods

Let $\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x})$ be a Hamiltonian system with Hamiltonian H and no conserved quantities other than H. Let φ^h be a symplectic and energy-preserving method for the Hamiltonian system. Then φ^h reproduces the exact solution up to a time re-parametrisation.

Symplectic numerical methods

A one-step numerical method $\varphi^h : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is symplectic if and only if when applied to a Hamiltonian system the map φ^h is symplectic, i.e.,

$$\left(rac{\partial arphi^h(\mathbf{x})}{\partial \mathbf{x}}
ight)^ op \mathbb{J}\left(rac{\partial arphi^h(\mathbf{x})}{\partial \mathbf{x}}
ight) = \mathbb{J}.$$

Symplectic and energy preserving methods

Let $\dot{\mathbf{x}} = \mathbb{J}\nabla H(\mathbf{x})$ be a Hamiltonian system with Hamiltonian H and no conserved quantities other than H. Let φ^h be a symplectic and energy-preserving method for the Hamiltonian system. Then φ^h reproduces the exact solution up to a time re-parametrisation.

Informal theorem

A symplectic method almost conserves the Hamiltonian for an exponentially long time.

Davide Murari (DAMTP)

Example: simple harmonic oscillator



Davide Murari (DAMTP)

Symplectic Neural Flows for Modelling and Discovery

Forward invariant subset of the phase space

• Suppose $\mathbf{x}(t) \in \Omega \subset \mathbb{R}^{2n}$, whenever $\mathbf{x}(0) \in \Omega$, for any $t \ge 0$.

Forward invariant subset of the phase space

▶ Suppose $\mathbf{x}(t) \in \Omega \subset \mathbb{R}^{2n}$, whenever $\mathbf{x}(0) \in \Omega$, for any $t \ge 0$.

▶ By the group property of the flow map, we know that

$$\phi_{H,n\Delta t+\delta t} = \phi_{H,\delta t} \circ \underbrace{\phi_{H,\Delta t} \circ \ldots \circ \phi_{H,\Delta t}}_{n \text{ times}}, \ n \in \mathbb{N}, \ \delta t \in (0,\Delta t).$$

As a consequence, to approximate $\phi_{H,t} : \Omega \to \Omega$ for any $t \ge 0$, we only have to approximate it for $t \in [0, \Delta t]$.



Figure 1: Neural network trained to approximate $\phi_{H,t}$ for $t \in [0, \Delta t = 1]$ and applied up to T = 100.

Davide Murari (DAMTP)

Unsupervised solution of the Hamiltonian equations

Approximate the flow map $\phi_{H,t}: \Omega \to \Omega$, for any $t \ge 0$, on a compact forward invariant set $\Omega \subset \mathbb{R}^{2n}$, given the Hamiltonian energy $H: \mathbb{R}^{2n} \to \mathbb{R}$.

Unsupervised solution of the Hamiltonian equations

Approximate the flow map $\phi_{H,t}: \Omega \to \Omega$, for any $t \ge 0$, on a compact forward invariant set $\Omega \subset \mathbb{R}^{2n}$, given the Hamiltonian energy $H: \mathbb{R}^{2n} \to \mathbb{R}$.

Supervised approximation of an unknown Hamiltonian flow map

Approximate the flow map $\phi_{H,t}: \Omega \to \Omega$, for any $t \ge 0$, on a compact forward invariant set $\Omega \subset \mathbb{R}^{2n}$, given trajectory segments $\{(\mathbf{x}_0^n, \mathbf{y}_1^n, ..., \mathbf{y}_M^n)\}_{n=1}^N$, $\mathbf{y}_m^n \approx \phi_{H,t_m^n}(\mathbf{x}_0^n)$.

Unsupervised solution of the Hamiltonian equations

Approximate the flow map $\phi_{H,t}: \Omega \to \Omega$, for any $t \ge 0$, on a compact forward invariant set $\Omega \subset \mathbb{R}^{2n}$, given the Hamiltonian energy $H: \mathbb{R}^{2n} \to \mathbb{R}$.

Supervised approximation of an unknown Hamiltonian flow map

Approximate the flow map $\phi_{H,t}: \Omega \to \Omega$, for any $t \ge 0$, on a compact forward invariant set $\Omega \subset \mathbb{R}^{2n}$, given trajectory segments $\{(\mathbf{x}_0^n, \mathbf{y}_1^n, ..., \mathbf{y}_M^n)\}_{n=1}^N$, $\mathbf{y}_m^n \approx \phi_{H,t_m^n}(\mathbf{x}_0^n)$.

Remark: Given the several known qualitative properties of $\phi_{H,t}$, we want to leverage them when designing the approximating map.

The SympFlow

▶ We now build a neural network that approximates $\phi_{H,t} : \Omega \to \Omega$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.

The SympFlow

- ▶ We now build a neural network that approximates $\phi_{H,t} : \Omega \to \Omega$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.
- ▶ We rely on two building blocks, which applied to $(\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{2n}$ write:

$$\phi_{\mathbf{p},t}((\mathbf{q},\mathbf{p})) = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - (\nabla_{\mathbf{q}}V(t,\mathbf{q}) - \nabla_{\mathbf{q}}V(0,\mathbf{q})) \end{bmatrix}, \ \phi_{\mathbf{q},t}((\mathbf{q},\mathbf{p})) = \begin{bmatrix} \mathbf{q} + (\nabla_{\mathbf{p}}K(t,\mathbf{p}) - \nabla_{\mathbf{p}}K(0,\mathbf{p})) \\ \mathbf{p} \end{bmatrix}$$

The SympFlow

- ▶ We now build a neural network that approximates $\phi_{H,t} : \Omega \to \Omega$ for a forward invariant set $\Omega \subset \mathbb{R}^{2n}$, and $t \in [0, \Delta t]$, while reproducing the qualitative properties of $\phi_{H,t}$.
- ▶ We rely on two building blocks, which applied to $(\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{2n}$ write:

$$\phi_{\mathbf{p},t}((\mathbf{q},\mathbf{p})) = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - (\nabla_{\mathbf{q}}V(t,\mathbf{q}) - \nabla_{\mathbf{q}}V(0,\mathbf{q})) \end{bmatrix}, \ \phi_{\mathbf{q},t}((\mathbf{q},\mathbf{p})) = \begin{bmatrix} \mathbf{q} + (\nabla_{\mathbf{p}}K(t,\mathbf{p}) - \nabla_{\mathbf{p}}K(0,\mathbf{p})) \\ \mathbf{p} \end{bmatrix}$$

The SympFlow architecture is defined as

$$\mathcal{N}_{\theta}\left(t,(\mathbf{q}_{0},\mathbf{p}_{0})\right) = \phi_{\mathbf{p},t}^{L} \circ \phi_{\mathbf{q},t}^{L} \circ \cdots \circ \phi_{\mathbf{p},t}^{1} \circ \phi_{\mathbf{q},t}^{1}((\mathbf{q}_{0},\mathbf{p}_{0})),$$

with

$$\begin{split} V^{i}(t,\mathbf{q}) &= \ell_{\theta_{3}^{i}} \circ \sigma \circ \ell_{\theta_{2}^{i}} \circ \sigma \circ \ell_{\theta_{1}^{i}} \left(\begin{bmatrix} \mathbf{q} \\ t \end{bmatrix} \right), \ \mathcal{K}^{i}(t,\mathbf{p}) = \ell_{\rho_{3}^{i}} \circ \sigma \circ \ell_{\rho_{2}^{i}} \circ \sigma \circ \ell_{\rho_{1}^{i}} \left(\begin{bmatrix} \mathbf{p} \\ t \end{bmatrix} \right) \\ \ell_{\theta_{k}^{i}}(x) &= \mathcal{A}_{k}^{i}x + \mathcal{A}_{k}^{i}, \ \ell_{\rho_{k}^{i}}(x) = \mathcal{B}_{k}^{i}x + \mathcal{B}_{k}^{i}, \ k = 1, 2, 3, \ i = 1, ..., L. \end{split}$$

Properties of the SympFlow

► The SympFlow is symplectic for every time t ∈ R. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\begin{split} \phi_{\mathbf{p},t}^{i}((\mathbf{q},\mathbf{p})) &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}}V^{i}(t,\mathbf{q}) - \nabla_{\mathbf{q}}V^{i}(0,\mathbf{q})\right) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \nabla_{\mathbf{q}}\left(\int_{0}^{t}\partial_{s}V^{i}(s,\mathbf{q})ds\right) \end{bmatrix} = \phi_{\widetilde{V}^{i},t}((\mathbf{q},\mathbf{p})), \end{split}$$

with $\widetilde{V}^{i}(t,(\mathbf{q},\mathbf{p})) = \partial_{t}V^{i}(t,\mathbf{q}).$

Properties of the SympFlow

► The SympFlow is symplectic for every time t ∈ R. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\begin{split} \phi_{\mathbf{p},t}^{i}((\mathbf{q},\mathbf{p})) &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}} V^{i}(t,\mathbf{q}) - \nabla_{\mathbf{q}} V^{i}(0,\mathbf{q}) \right) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \nabla_{\mathbf{q}} \left(\int_{0}^{t} \partial_{s} V^{i}(s,\mathbf{q}) ds \right) \end{bmatrix} = \phi_{\widetilde{V}^{i},t}((\mathbf{q},\mathbf{p})), \end{split}$$

with $\widetilde{V}^{i}(t,(\mathbf{q},\mathbf{p})) = \partial_{t}V^{i}(t,\mathbf{q}).$

▶ The SympFlow is volume preserving.

Properties of the SympFlow

► The SympFlow is symplectic for every time t ∈ R. The building blocks we compose are exact flows of time-dependent Hamiltonian systems:

$$\begin{split} \phi_{\mathbf{p},t}^{i}((\mathbf{q},\mathbf{p})) &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \left(\nabla_{\mathbf{q}}V^{i}(t,\mathbf{q}) - \nabla_{\mathbf{q}}V^{i}(0,\mathbf{q})\right) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{q} \\ \mathbf{p} - \nabla_{\mathbf{q}}\left(\int_{0}^{t}\partial_{s}V^{i}(s,\mathbf{q})ds\right) \end{bmatrix} = \phi_{\widetilde{V}^{i},t}((\mathbf{q},\mathbf{p})), \end{split}$$

with $\widetilde{V}^{i}(t,(\mathbf{q},\mathbf{p})) = \partial_{t}V^{i}(t,\mathbf{q}).$

- The SympFlow is volume preserving.
- ▶ The SympFlow is the exact solution of a time-dependent Hamiltonian system.

Theorem (The Hamiltonian flows are closed under composition)

Let $H^1, H^2 : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}$ be continuously differentiable functions. Then, the map $\phi_{H^2,t} \circ \phi_{H^1,t} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is the exact flow of the time-dependent Hamiltonian system defined by the Hamiltonian function

$$egin{aligned} \mathcal{H}^3(t,\mathbf{x}) &= \mathcal{H}^2(t,\mathbf{x}) + \mathcal{H}^1\left(t,\phi_{H^2,t}^{-1}(\mathbf{x})
ight). \end{aligned}$$

▶ This theorem implies that there is a Hamiltonian function $\mathcal{H}(\mathcal{N}_{\theta}) : \mathbb{R} \times \mathbb{R}^{2n} \to \mathbb{R}$ such that

$$\mathcal{N}_{ heta}\left(t, \mathsf{x}
ight) = \phi_{\mathcal{H}\left(\mathcal{N}_{ heta}
ight), t}(\mathsf{x})$$

for every $t \geq 0$ and $\mathbf{x} \in \mathbb{R}^{2n}$.

¹Leonid Polterovich. *The Geometry of the Group of Symplectic Diffeomorphisms*. Lectures in Mathematics ETH Zürich. Basel: Springer Basel AG, 2001. ISBN: 978-3-7643-6432-8.

Davide Murari (DAMTP)

Symplectic Neural Flows for Modelling and Discovery

Extension of the SympFlow outside of $[0, \Delta t]$

▶ Once we have trained \mathcal{N}_{θ} to be reliable for $t \in [0, \Delta t]$, we extend it for longer times as

$$\psi\left(t, \mathsf{x}_{0}
ight) := ar{\psi}_{t-\Delta t \lfloor t/\Delta t
floor} \circ \left(ar{\psi}_{\Delta t}
ight)^{\lfloor t/\Delta t
floor} \left(\mathsf{x}_{0}
ight),$$

for $t\in [0,+\infty)$ and $\mathbf{x}_0\in \Omega\subset \mathbb{R}^{2n}$, where

$$\begin{split} \bar{\psi}_{s}\left(\mathbf{x}_{0}\right) &:= \mathcal{N}_{\theta}\left(s, \mathbf{x}_{0}\right), \ s \in [0, \Delta t), \\ \left(\bar{\psi}_{\Delta t}\right)^{k} &:= \underbrace{\bar{\psi}_{\Delta t} \circ \cdots \circ \bar{\psi}_{\Delta t}}_{k \text{ times}}, \ k \in \mathbb{N}. \end{split}$$

Extension of the SympFlow outside of $[0, \Delta t]$

▶ Once we have trained \mathcal{N}_{θ} to be reliable for $t \in [0, \Delta t]$, we extend it for longer times as

$$\psi\left(t, \mathbf{x}_{0}\right) := \bar{\psi}_{t-\Delta t \lfloor t/\Delta t \rfloor} \circ \left(\bar{\psi}_{\Delta t}\right)^{\lfloor t/\Delta t \rfloor} \left(\mathbf{x}_{0}\right),$$

for $t\in [0,+\infty)$ and $\mathbf{x}_0\in \Omega\subset \mathbb{R}^{2n}$, where

$$\begin{split} \bar{\psi}_{s}\left(\mathbf{x}_{0}\right) &:= \mathcal{N}_{\theta}\left(s, \mathbf{x}_{0}\right), \ s \in [0, \Delta t), \\ \left(\bar{\psi}_{\Delta t}\right)^{k} &:= \underbrace{\bar{\psi}_{\Delta t} \circ \cdots \circ \bar{\psi}_{\Delta t}}_{k \text{ times}}, \ k \in \mathbb{N}. \end{split}$$

▶ $\psi(t, \cdot) = \phi_{\widetilde{H},t}$ for the piecewise continuous Hamiltonian

$$\widetilde{H}\left(t, \mathbf{x}
ight) := \mathcal{H}\left(\mathcal{N}_{ heta}
ight) \left(t - \Delta t \lfloor t / \Delta t
floor, \mathbf{x}
ight).$$

Theorem

Let $H : \mathbb{R} \times \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^{2n}$ compact, be a continuously differentiable function. For any $\varepsilon > 0$, there is a SympFlow $\overline{\psi}_t$ such that

$$\sup_{\substack{t\in[0,\Delta t]\\\mathbf{x}\in\Omega}} \left\|\bar{\psi}_t(\mathbf{x}) - \phi_{H,t}(\mathbf{x})\right\|_{\infty} < \varepsilon.$$

Training of the SympFlow to solve $\dot{\mathbf{x}}(t) = X_H(\mathbf{x}(t))$

- ▶ The SympFlow is based on modelling the scalar-valued potentials $\widetilde{V}^i, \widetilde{K}^i : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ with feed-forward neural networks.
- \blacktriangleright To train the overall model \mathcal{N}_{θ} we minimise the loss function

$$\mathcal{L}(\theta) = \underbrace{\frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \frac{d}{dt} \mathcal{N}_{\theta} \left(t, \mathbf{x}_0^i \right) \right\|_{t=t_i}^{t=t_i} - \mathbb{J} \nabla H \left(\mathcal{N}_{\theta} \left(t_i, \mathbf{x}_0^i \right) \right) \right\|_2^2}_{\text{Residual term}} + \underbrace{\frac{1}{N_m} \sum_{j=1}^{N_m} \left(\mathcal{H}(\mathcal{N}_{\theta})(t_j, \mathbf{x}^j) - H(\mathbf{x}^j) \right)^2}_{\text{Hamiltonian matching}},$$

where we sample $t_i, t_j \in [0, \Delta t]$, and $\mathbf{x}_0^i, \mathbf{x}^i \in \Omega \subset \mathbb{R}^{2n}$.

Davide Murari (DAMTP)

Supervised training of the SympFlow to approximate $\phi_{H,t}$

▶ To train the overall model \mathcal{N}_{θ} we minimise the loss function

$$\mathcal{L}(heta) = rac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} \|\mathcal{N}_{ heta}\left(t_m^n, \mathbf{x}_0^n
ight) - \mathbf{y}_m^n\|_2^2,$$

where $\mathbf{x}_0^n \in \Omega \subset \mathbb{R}^{2n}$, and $\mathbf{y}_m^n \approx \phi_{H,t_m^n}(\mathbf{x}_0^n)$.



Simple Harmonic Oscillator (unsupervised)

Equations of motion

$$\dot{x} = p, \ \dot{p} = -x.$$





Simple Harmonic Oscillator (supervised)



(b) Fixed N = 100 and M = 50.

Equations of motion

$$\dot{x} = p_x, \ \dot{y} = p_y, \ \dot{p}_x = -x - 2xy, \ \dot{p}_y = -y - (x^2 - y^2).$$



Figure 3: **Unsupervised experiment** — **Hénon–Heiles:** Comparison of the Poincaré sections and the energy behaviour up to time T = 1000.

- Improve the efficiency of the method by replacing gradients of MLPs with some other alternatives (Topic of a Summer Project in Cambridge that Zak and I proposed).
- Extend the approach to capture parametric dependencies, and apply this procedure for parameter identification.
- ▶ Improve our theoretical understanding of the dynamics exactly solved by the SympFlow.
- Apply the method to higher dimensional systems.

THANK YOU FOR THE ATTENTION

davidemurari.com/sympflow to read the paper

Physics-informed neural networks

▶ We introduce a parametric map $\mathcal{N}_{\theta}(\cdot, \mathbf{x}_0) : [0, T] \to \mathbb{R}^d$ such that $\mathcal{N}_{\theta}(0, \mathbf{x}_0) = \mathbf{x}_0$, and choose its weights so that

$$\mathcal{L}(heta) := rac{1}{C} \sum_{c=1}^{C} \left\| rac{d}{dt} \mathcal{N}_{ heta}\left(t, \mathbf{x}_{0}
ight)
ight|_{t=t_{c}} - \mathcal{F}\left(\mathcal{N}_{ heta}\left(t_{c}, \mathbf{x}_{0}
ight)
ight)
ight\|_{2}^{2} o \min$$

for some collocation points $t_1, \ldots, t_C \in [0, T]$.

Physics-informed neural networks

▶ We introduce a parametric map $\mathcal{N}_{\theta}(\cdot, \mathbf{x}_0) : [0, T] \to \mathbb{R}^d$ such that $\mathcal{N}_{\theta}(0, \mathbf{x}_0) = \mathbf{x}_0$, and choose its weights so that

$$\mathcal{L}(heta) := rac{1}{C} \sum_{c=1}^{C} \left\| rac{d}{dt} \mathcal{N}_{ heta}\left(t, \mathbf{x}_{0}
ight)
ight|_{t=t_{c}} - \mathcal{F}\left(\mathcal{N}_{ heta}\left(t_{c}, \mathbf{x}_{0}
ight)
ight)
ight\|_{2}^{2} o \min$$

for some collocation points $t_1, \ldots, t_C \in [0, T]$.

▶ Then, $t \mapsto \mathcal{N}_{\theta}(t, \mathbf{x}_{0})$ will solve a different IVP

$$egin{split} \left\{ \dot{\mathbf{y}}\left(t
ight) = \mathcal{F}\left(\mathbf{y}\left(t
ight)
ight) + \left(rac{d}{dt}\mathcal{N}_{ heta}\left(t,\mathbf{x}_{0}
ight)
ight|_{t=t} - \mathcal{F}\left(\mathbf{y}\left(t
ight)
ight)
ight) \in \mathbb{R}^{d}, \ \mathbf{y}\left(0
ight) = \mathbf{x}_{0} \in \mathbb{R}^{d}, \end{split}$$

where hopefully the residual $\frac{d}{dt}\mathcal{N}_{\theta}(t,\mathbf{x}_{0})|_{t=t} - \mathcal{F}(\mathbf{y}(t))$ is small in some sense.

Training issues with neural network

- Solving a single IVP on [0, T] with a neural network can take long training time.
- The obtained solution can not be used to solve the same ordinary differential equation with a different initial condition.



Figure 4: Solution comparison after reaching a loss value of 10^{-5} . The training time is of 87 seconds (7500 epochs with 1000 new collocation points randomly sampled at each of them).

Training issues with neural network

▶ It is hard to solve initial value problems over long time intervals.

