SympNets & PNNs: Intrinsic structure-preserving networks for identifying Hamiltonian & Poisson systems

Zhen Zhang
Division of Applied Mathematics
Brown University

joint with Pengzhan Jin, Aiqing Zhu, George Em Karniadakis, Yifa Tang

Sep. 7th, 2021
NUMDIFF-16

Outline

Introduction

Denote the d-by-d identity matrix by I_d, let

$$J := \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}.$$

In consideration of the Hamiltonian system

$$\begin{cases} \dot{y} = J^{-1}\nabla H(y) \\ y(t_0) = y_0 \end{cases}, \tag{1}$$

where $y(t) \in \mathbb{R}^{2d}$, and H is the Hamiltonian function, usually representing the energy of the system. Let $\phi_h(y)$ be the phase transformation of above system, i.e., $\phi_h(y_0) = y(t_0 + h)$. $\phi_h(y)$ in fact gives the next phase point after time step h with respect to the phase point y.

Introduction

Definition

A matrix $H \in \mathbb{R}^{2d \times 2d}$ is called symplectic if $H^T J H = J$.

Definition

A differentiable map $\Phi : U \to \mathbb{R}^{2d}$ (where $U \subset \mathbb{R}^{2d}$ is an open set) is called symplectic if the Jacobian matrix $\frac{\partial \Phi}{\partial x}$ is everywhere symplectic, i.e.,

$$\left(\frac{\partial \Phi}{\partial x}\right)^T J \left(\frac{\partial \Phi}{\partial x}\right) = J.$$

Introduction

The connection between Hamiltonian system and symplectic map:

In 1899, Poincare pointed out that the phase transformation of the Hamiltonian system is a symplectic map, i.e.,

$$\left(\frac{\partial \phi_h}{\partial x}\right)^T J \left(\frac{\partial \phi_h}{\partial x}\right) = J.$$

"It is natural to look forward to those discrete systems which preserve as much as possible the intrinsic properties of the continuous system." (Kang Feng, 1984)

Introduction

- Forward problem: solve the Hamiltonian system by **symplectic integrators**, such as the mid-point rule:

$$\bar{y} = y + hJ^{-1}\nabla H(\frac{\bar{y}+y}{2}).$$

(It is straightforward to verify that $\left(\frac{\partial \bar{y}}{\partial y}\right)^T J \left(\frac{\partial \bar{y}}{\partial y}\right) = J$)

- Inverse problem: how to discover the Hamiltonian system, given the data of phase points from a unknown Hamiltonian system?

We know less about the inverse problem, while the forward problem is well developed.

Introduction

Problem setup:

Given the dataset $\{(x_i, y_i)\}$ satisfying $\phi_h(x_i) = y_i$, to predict the phase flow $\phi_h(x), \phi_h^2(x), \phi_h^3(x) \cdots$, where $\phi_h^2(x)$ means $\phi_h \circ \phi_h(x)$.

Mostly, the dataset $\{(x_i, y_i)\}$ can be a series of phase points depending on time, i.e., $\{(x_i, x_{i+1})\}$ where $\phi_h(x_i) = x_{i+1}$.

Introduction

Hamiltonian neural networks (NIPS 2019)
by Samuel Greydanus, Misko Dzamba, Jason Yosinski.

Using a neural network to approximate the Hamiltonian H with loss

$$\|\frac{dy}{dt} - J^{-1}\nabla\widetilde{H}(y)\|,$$

and the time derivative is discretized by a numerical integrator (in fact should be a **symplectic integrator**, they did not mention this point, but it is indeed needed, proved by other later works). For example, with mid-point rule,

$$\|\frac{x_{i+1} - x_i}{h} - J^{-1}\nabla\widetilde{H}(\frac{x_i + x_{i+1}}{2})\|.$$

Finally computing the predicted flow with the trained network $\widetilde{H}$.

Introduction

Another strategy: directly learn the phase transformation $\phi_{\boldsymbol{h}}$.

To this end, we expect to construct the neural networks preserving the symplecticity. Our architecture design philosophy is based on the fact that the composition of symplectic maps is again symplectic.

Purpose

To construct the symplectic networks, we have to search for satisfactory unconstrained parametrization of symplectic matrices. We first show some existing modern factorizations.

Denote the collection of symplectic matrices by

$$SP = \{H \in \mathbb{R}^{2d \times 2d} | H^T J H = J\}.$$

The modern factorizations

- Most of the factorizations require cells like symplectic-orthogonal matrices, which are not elementary enough hence hard to be freely parameterized.
- Transvections factorization needs $\mathcal{O}(d)$ generators.
- We need a more elementary unconstrained factorization, since the deep learning techniques focus on the unconstrained optimization.

Unit triangular factorization

Unit triangular symplectic matrices

$$\begin{pmatrix} I & S \\ 0 & I \end{pmatrix}, \quad \begin{pmatrix} I & 0 \\ S & I \end{pmatrix}, \quad S^T = S$$

We can rewrite the symmetric S as $(W^T + W)$ in practice, then the W is unconstrained. Denote

$$L_n = \left\{ \begin{pmatrix} I & 0/S_n \\ S_n/0 & I \end{pmatrix} \cdots \begin{pmatrix} I & S_3 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ S_2 & I \end{pmatrix} \begin{pmatrix} I & S_1 \\ 0 & I \end{pmatrix} \middle| \right.$$
$$\left. S_i \in \mathbb{R}^{d\times d}, S_i^T = S_i, i = 1, 2, \cdots, n \right\},$$

where the unit upper triangular symplectic matrices and the unit lower triangular symplectic matrices appear alternately. And it is clear that $L_m \subset L_n \subset SP$ for all integers $1 \leq m \leq n$.

Unit triangular factorization

Theorem (Unit Triangular Factorization)

$SP = L_9$.

Thus any symplectic matrix can be factored into no more than 9 unit triangular symplectic matrices. **Only $\mathcal{O}(1)$ generators.**

Unit triangular factorization of the matrix symplectic group (2019, by Pengzhan Jin, Yifa Tang, Aiqing Zhu.)

SympNets

- **Linear Modules**

$$\mathcal{L}_n(x) = \begin{pmatrix} I & 0/S_n \\ S_n/0 & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ S_2 & I \end{pmatrix} \begin{pmatrix} I & S_1 \\ 0 & I \end{pmatrix} x + b, \quad x, b \in \mathbb{R}^{2d}, S_i^T = S_i.$$

Here $S_i = W_i^T + W_i \in \mathbb{R}^{d\times d}$, and W_i are the network parameters to be optimized.

- **Activation Modules**

$$\mathcal{N}_{up}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & \tilde{\sigma}_a \\ 0 & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} := \begin{pmatrix} p + diag(a)\sigma(q) \\ q \end{pmatrix}, \quad \mathcal{N}_{low}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & 0 \\ \tilde{\sigma}_a & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix}.$$

Here $a \in \mathbb{R}^d$ is the network parameter to be optimized.

- **Gradient Modules**

$$\mathcal{G}_{up}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & \hat{\sigma}_{K,a,b} \\ 0 & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix} := \begin{pmatrix} p + K^T diag(a)\sigma(Kq + b) \\ q \end{pmatrix},$$

$$\mathcal{G}_{low}\begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & 0 \\ \hat{\sigma}_{K,a,b} & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix}.$$

Here $K \in \mathbb{R}^{m\times d}, a, b \in \mathbb{R}^m$ are the network parameters to be optimized.

SympNets

Remark

The term "gradient module": we proved that $\hat{\sigma}_{K,a,b}$ can approximate any ∇f for $f : \mathbb{R}^d \to \mathbb{R}$.

$$\mathcal{M}_L = \{\psi | \psi \text{ is a linear module}\}.$$
$$\mathcal{M}_A = \{\psi | \psi \text{ is an activation module}\}.$$
$$\mathcal{M}_G = \{\psi | \psi \text{ is a gradient module}\}.$$

Definition

Consider $\{v_i\}_1^{k+1} \subset \mathcal{M}_L$, $\{w_i\}_1^k \subset \mathcal{M}_A$. Let

$$\psi = v_{k+1} \circ w_k \circ v_k \circ \cdots \circ w_1 \circ v_1.$$

ψ is called the **LA-SympNet**. We define the collection of all the LA-SympNets as

$$\Psi_{LA} = \{\psi | \psi \text{ is a LA-SympNet}\}.$$

SympNets

Definition

Consider $\{u_i\}_1^k \subset \mathcal{M}_G$. Let

$$\psi = u_k \circ u_{k-1} \circ \cdots \circ u_1.$$

ψ is called the **G-SympNet**. We define the collection of all the G-SympNets as

$$\Psi_G = \{\psi | \psi \textit{ is a G-SympNet}\}.$$

Definition

Consider $\{v_i\}_1^k \subset \mathcal{M}_L \cup \mathcal{M}_A \cup \mathcal{M}_G$, where $\mathcal{M}_L$, $\mathcal{M}_A$ and $\mathcal{M}_G$ are the set of linear, activation and gradient modules respectively. Let

$$\psi = v_k \circ v_{k-1} \circ \cdots \circ v_1.$$

ψ is called the **symplectic network (SympNet)**. Futhermore, we define the collection of all the SympNets as

$$\Psi = \{\psi | \psi \textit{ is a symplectic network}\}.$$

SympNets

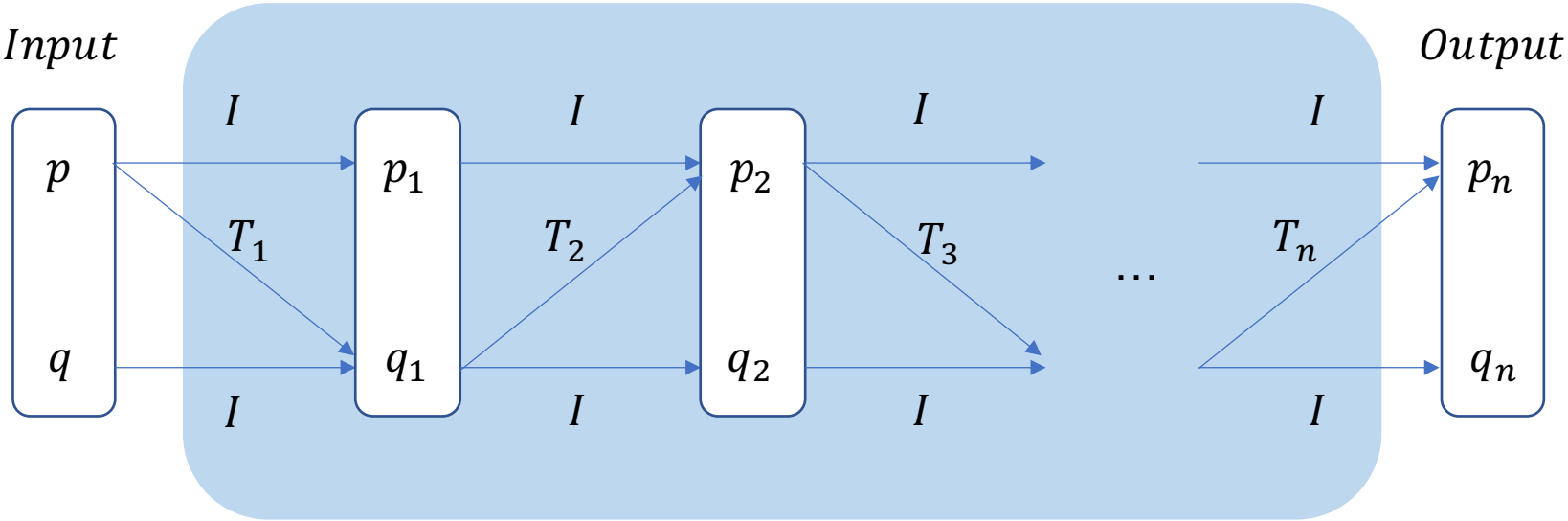

Figure 1: **Architecture of SympNet.** The SympNet can be seen as the neural network with a specific connection pattern, which gaurantees symplecticity. Here T_i can be chosen as S, $\tilde{\sigma}$ or $\hat{\sigma}$, depending on which type of module it belongs to.

Approximation theorems

$$\mathcal{SP}^r(U) = \left\{ \Phi \in C^r(U; \mathbb{R}^{2d}) \middle| \left(\frac{\partial \Phi}{\partial x}\right)^T J \left(\frac{\partial \Phi}{\partial x}\right) = J \right\}.$$

Definition

Let $r \in \{0\} \cup \mathbb{N}$ be given. σ is r-finite if $\sigma \in C^r(\mathbb{R})$ and $0 < \int |D^r(\sigma)| d\lambda < \infty$, where λ is the Lebesgue measure on $\mathbb{R}$.

Theorem (Approximation theorem for G-SympNet)

If the activation function σ is r-finite, then the set of G-SympNet Ψ_G is r-uniformly dense on compacta in $\mathcal{SP}^r(U)$.

Theorem (Approximation theorem for LA-SympNet)

If the activation function σ is r-finite, then the set of LA-SympNet Ψ_{LA} is r-uniformly dense on compacta in $\mathcal{SP}^r(U)$.

Three body problem

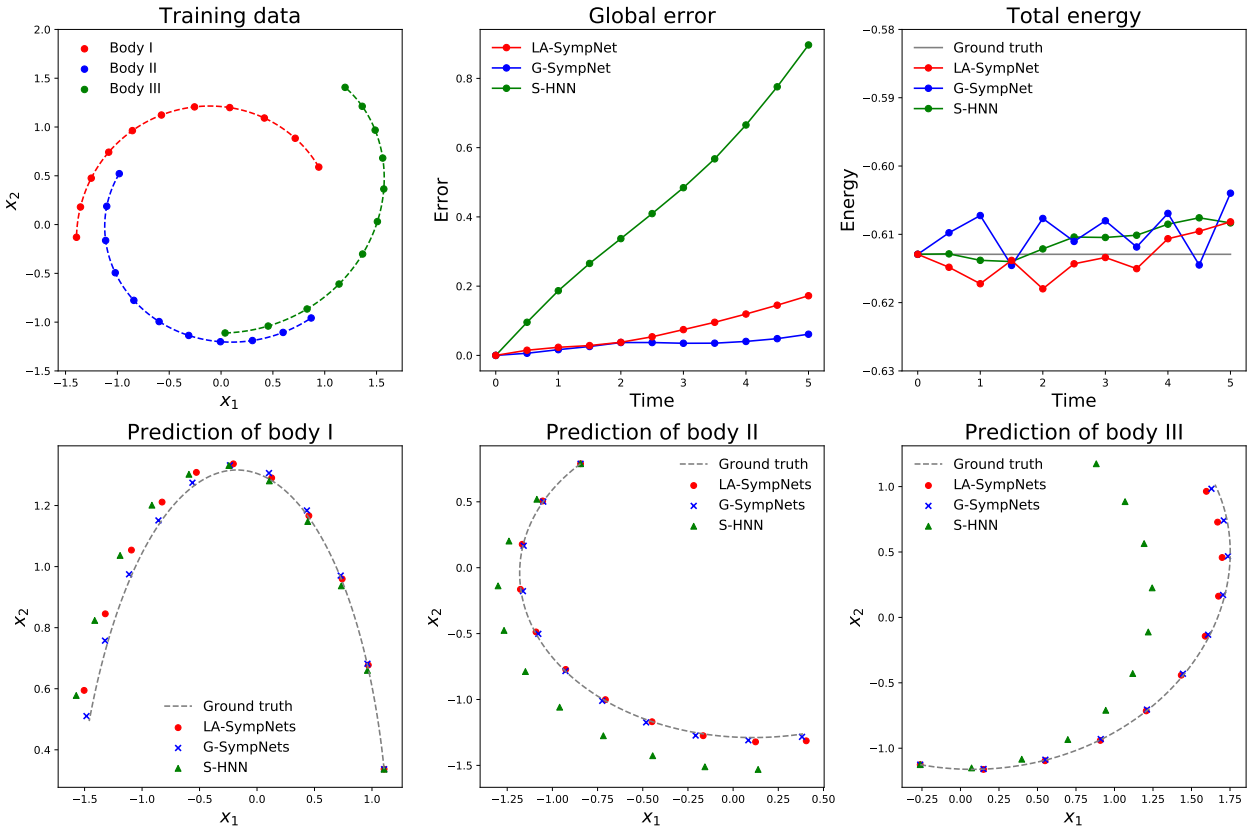

Figure 2: **Results for the three-body system.** (**Top-left**) One trajectory from the training dataset. (**Top-middle**) The global error versus time. (**Top-right**) The total energies for the predictions on the representative trajectory. (**Bottom**) Predicted position q on the representative trajectory.

Extension

Definition

If a matrix-valued function $B(y)$ is anti-commutative, bilinear and satisfies Leibniz's rule then

$$(\{F, G\}_B)(y) = \nabla F(y)^T B(y) \nabla G(y), \tag{2}$$

defines a Poisson bracket, and the corresponding differential system

$$\dot{y} = B(y)\nabla H(y)$$

is a Poisson system. Here H is still called a Hamiltonian.

Up to now, the Hamiltonian system and the Poisson system have been unified as

$$\dot{y}_i = \{y_i, H\}_B, \quad i = 1, \cdots, n, \tag{3}$$

for B satisfying the previous lemma, and the system becomes Hamiltonian when $B = J^{-1}$.

Darboux-Lie theorem

Theorem (Darboux 1882, Lie 1888)

Suppose that the matrix $B(y)$ defines a Poisson bracket and is of constant rank $n - q = 2d$ in a neighbourhood of $y_0 \in \mathbb{R}^n$. Then, there exist functions $P_1(y), \cdots, P_d(y)$, $Q_1(y), \cdots, Q_d(y)$, and $C_1(y), \cdots, C_q(y)$ such that $y \to (P_i(y), Q_i(y), C_k(y))$ constitutes a local change of coordinates to canonical form. With this change of coordinates, the Poisson system $\dot{y} = B(y)\nabla H(y)$ becomes

$$\dot{z} = B_0 \nabla K(z) \quad \textit{with} \quad B_0 = \begin{pmatrix} J^{-1} & 0 \\ 0 & 0 \end{pmatrix},$$

where $K(z) = H(y)$. Writing $z = (p, q, c)$, this system becomes

$$\dot{p} = -K_q(p, q, c), \quad \dot{q} = K_p(p, q, c), \quad \dot{c} = 0.$$

Poisson Neural Networks

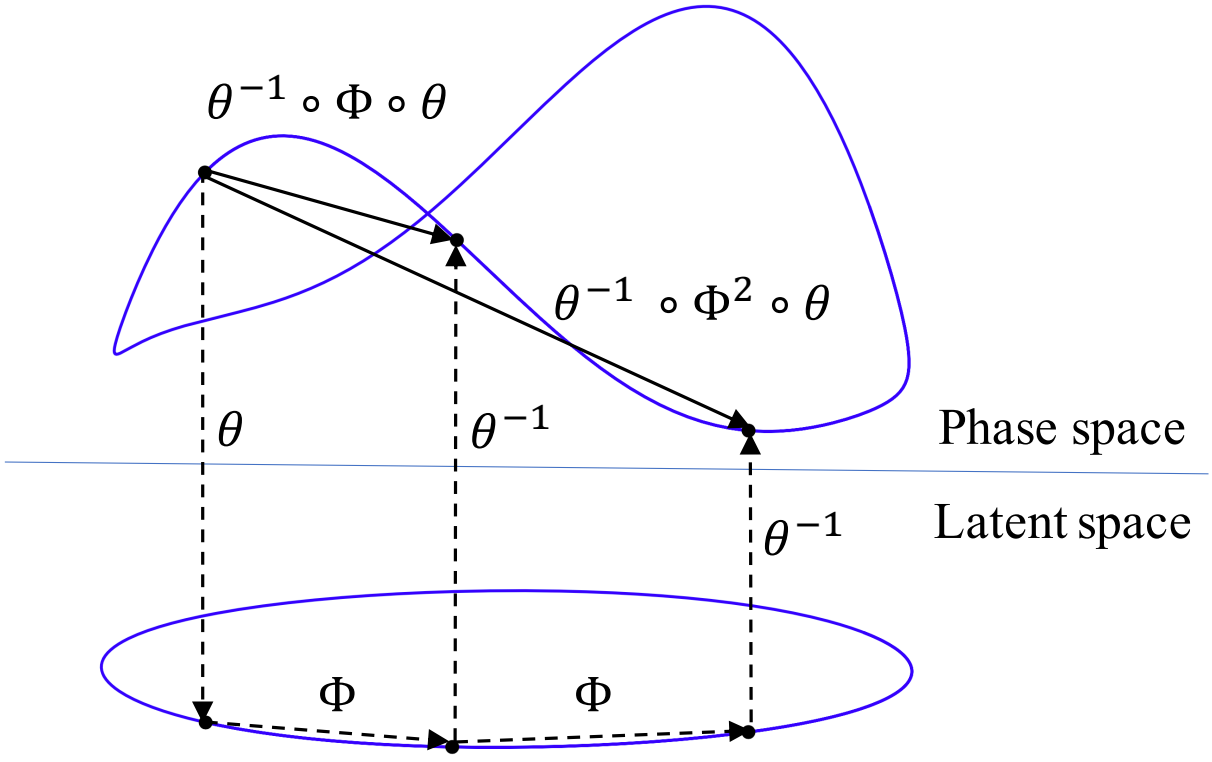

Figure 3: **Architecture of PNN.** PNNs are composed of three parts: (1) a transformation, (2) an extended symplectic map, and (3) the inverse of the transformation, denoted by θ, Φ, and θ^{-1} respectively.

Neural ODE, PNN and SympNet

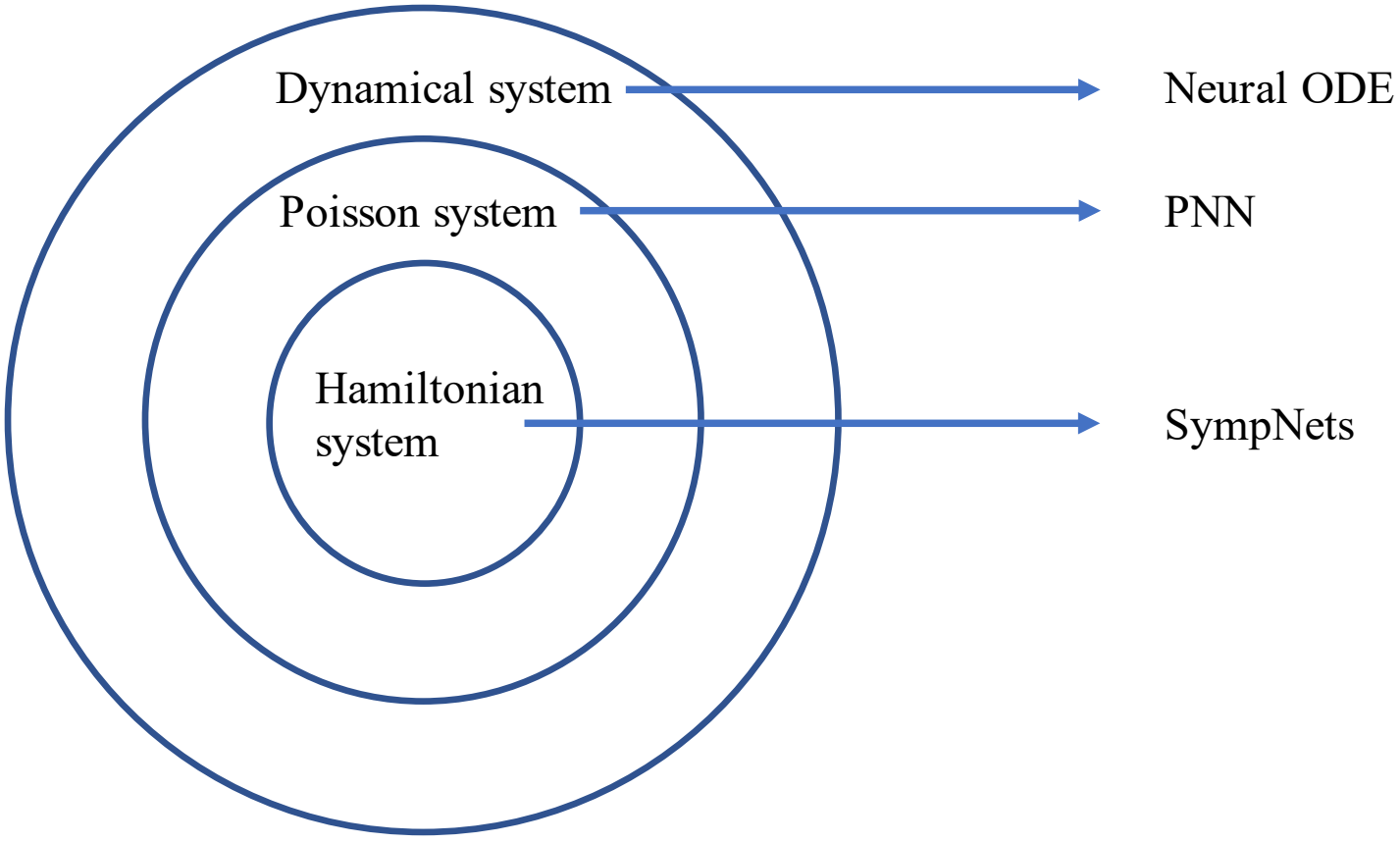

Figure 4: **Neural ODE, PNN and SympNet**. Prior knowledge used: SympNet $>$ PNN $>$ Neural ODE; Model expressivity: Neural ODE $>$ PNN $>$ SympNet.

Charged particle in an electromagnectic potential

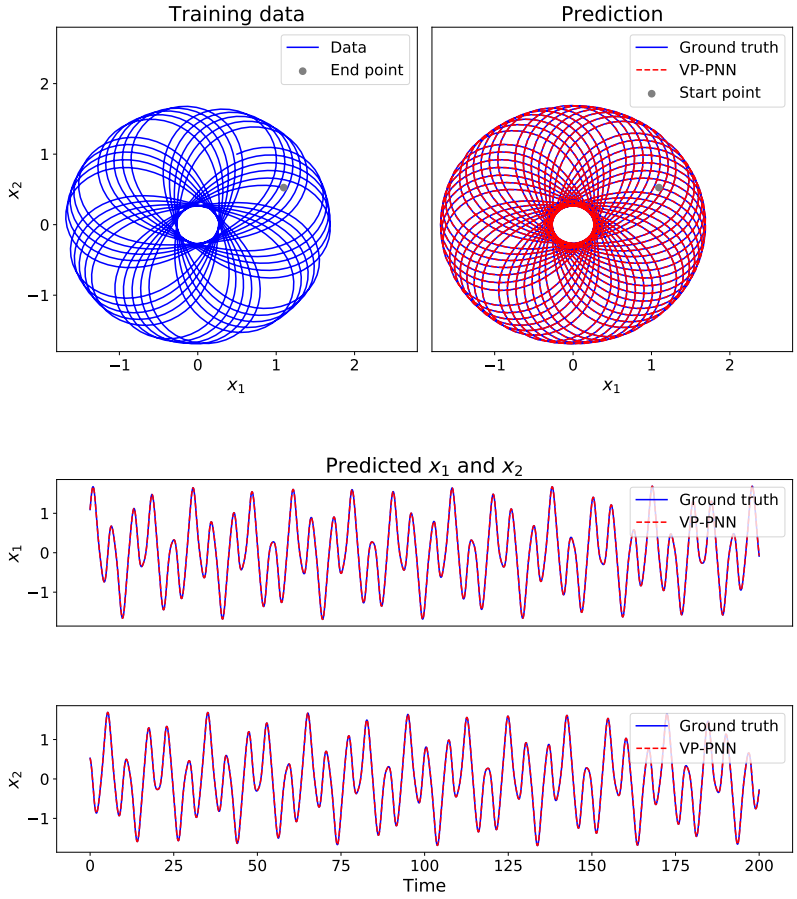

Figure 5: **Charged particle in the electromagnetic potential.** (**Top-left**) The position (x_1, x_2) of training flow. (**Top-right**) Prediction of the VP-PNN starting at the end point of training flow for 2000 steps. (**Bottom**) Prediction of the position of particle over time.

Nonlinear Schrödinger equation

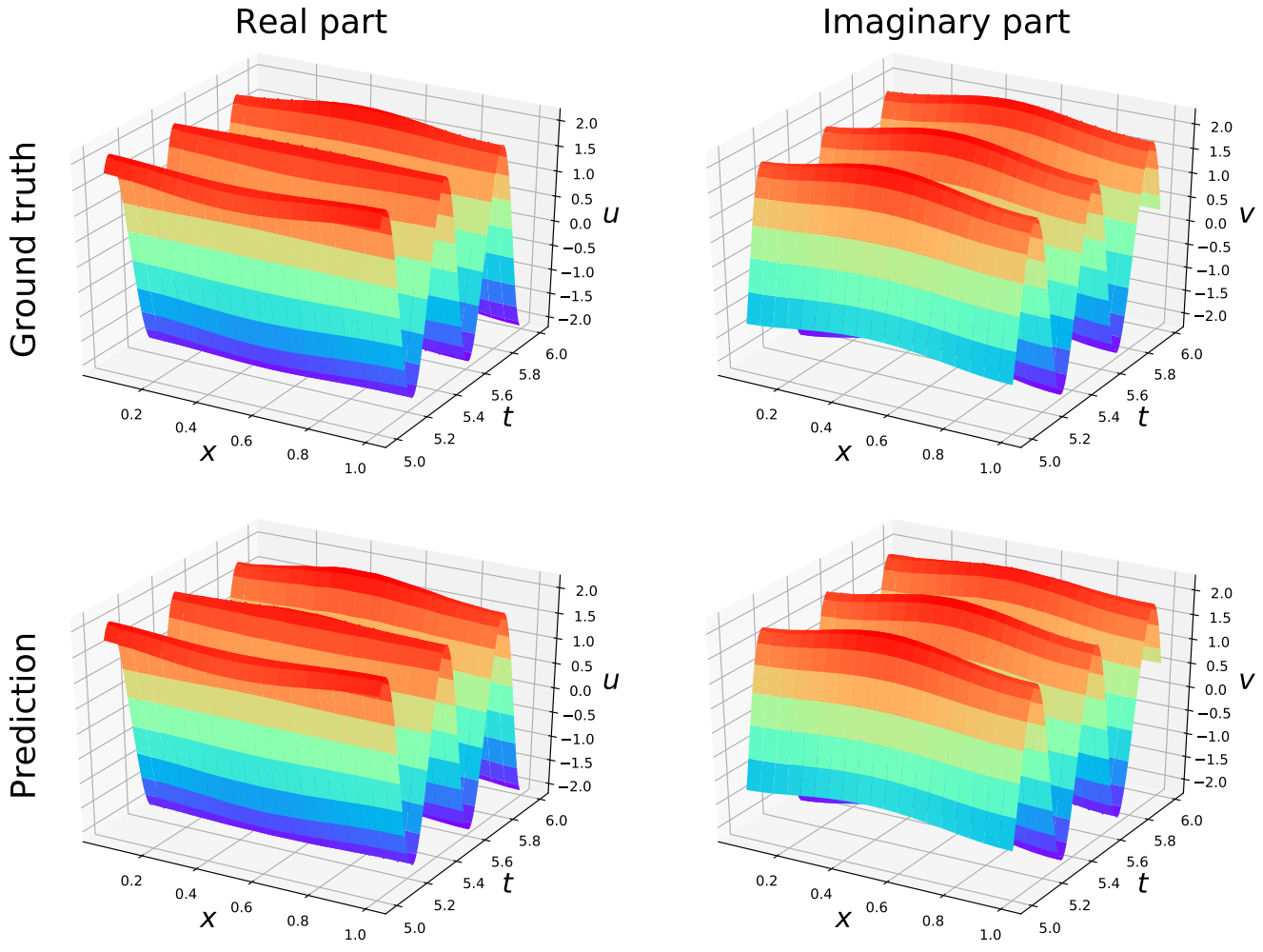

Figure 6: **Ablowitz–Ladik model of nonlinear Schrödinger equation.** Predictions of both real and imaginary parts match the ground truth well.

Pixel observations of two-body problem

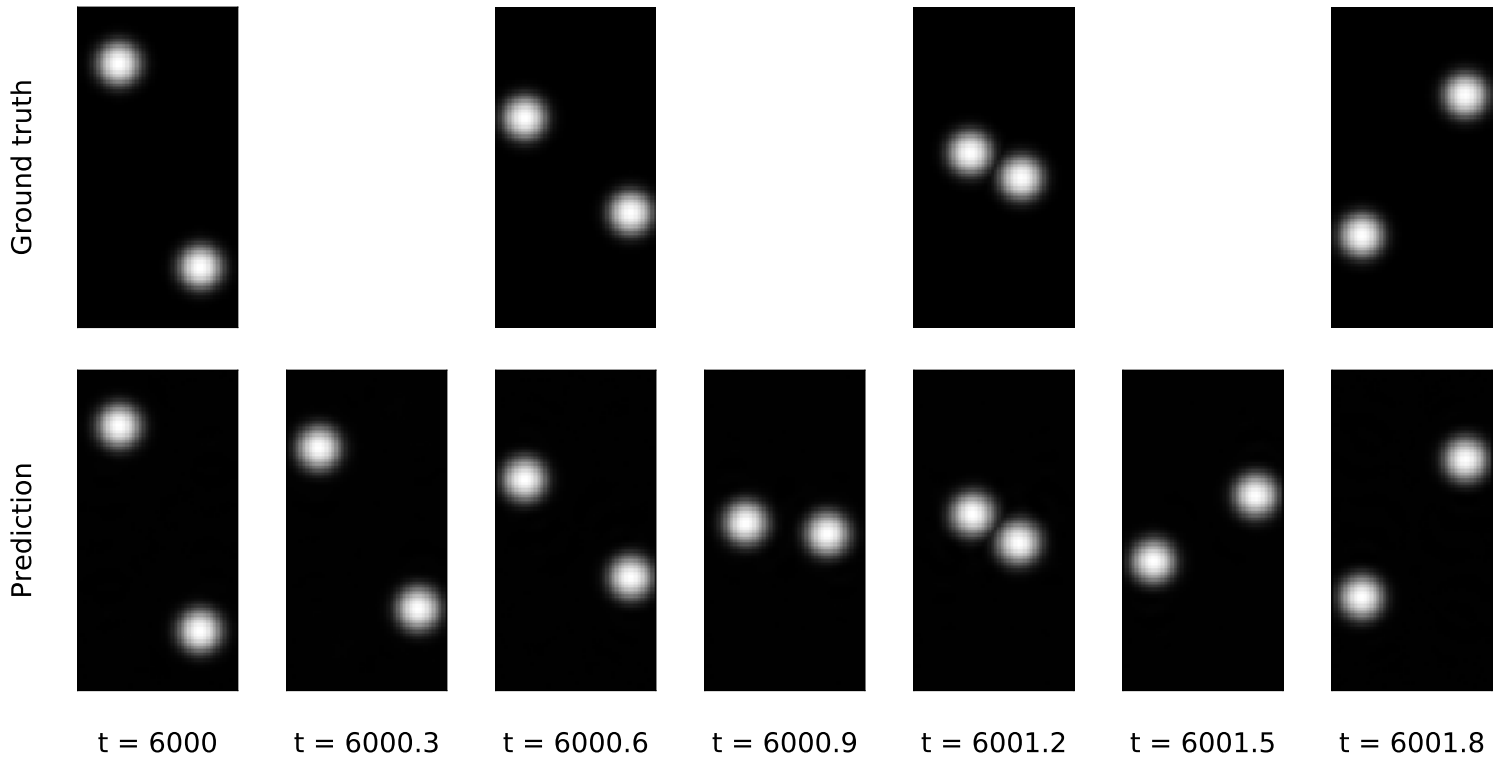

Figure 7: **Long time prediction and frame interpolation for two body images.** (**Top**) Ground truth, four consecutive points in the test dataset, after $t = 6000$ with step size $h = 0.6$. (**Bottom**) Predictions made by the PNN, with a finer step size $h = 0.3$. (**All**) PNNs can handle long-time integration and frame interpolation perfectly.

Summary

- We solve the inverse problem for Hamiltonian and Poisson systems using neural networks (SympNets and PNNs).
- Universality theorems are established.
- Applications to the prediction of ODEs, PDEs.
- Future work: Control problem, molecular dynamics.