Structure Preserving Neural Networks and Applications to Optimal Control Problems

Zhen Zhang Pengzhan Jin George Em Karniadakis

Division of Applied Mathematics, Brown University

BROWN

BROWN | Applied Mathematics

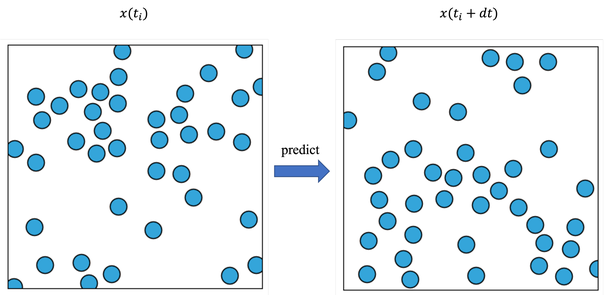


Figure 1. Problem setup. The state variable $x(t) = (p(t), q(t))$ includes information on momentum p and position q.

Problem setup: Given the state of a system at previous step $x(t_i - dt)$, we want to predict $x(t_i)$ using data-driven models. I.e., design a neural network ϕ such that $x(t_i) \approx \phi(x(t_i - dt))$ and and satisfies the following properties:

- Permutation equivariance
- conservation of energy
- volume-preserving in the phase space

For a continuously differentiable symplectic map ϕ, there exists a (locally) Hamiltonian system

$$\dot{x} = J^{-1}\nabla H(x)$$

- Conservation of energy $H(x) = H(\phi(x))$
- Volume-preserving in the phase space $|A| = |\phi(A)|$

Our first goal reduced to approximate a symplectic map with easily parameterizable model.

Symplectic networks (SympNets)

Approximate linear symplectic map

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

- L_n is symplectic. We will refer to it as symplectic linear module.
- **Theorem:** L_5 is the set of all linear symplectic maps. In other words, any symplectic matrix can be factorized into no more than 5 unit triangular matrices.

Approximate general symplectic map

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

- N_{up}, N_{low} are symplectic. We will refer to them as symplectic activation module.
- We will refer to the composition of L_n, N_{up} and N_{low} as LA-SympNet.
- **Theorem:** Under some mild conditions, LA-SympNets can approximate arbitrary symplectic maps and their derivatives.
- Gradient module (auxiliary):

$$\mathcal{G}_{up} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & \nabla_q V(\cdot) \\ 0 & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix}, \quad \mathcal{G}_{low} \begin{pmatrix} p \\ q \end{pmatrix} = \begin{bmatrix} I & 0 \\ \nabla_p T(\cdot) & I \end{bmatrix} \begin{pmatrix} p \\ q \end{pmatrix}$$

where T and V are standard fully-connected neural networks.

- We will refer to the composition of $\mathcal{G}_{up}$ and $\mathcal{G}_{low}$ as G-SympNet.
- **Theorem:** Under some mild conditions, G-SympNets can approximate arbitrary symplectic maps and their derivatives.

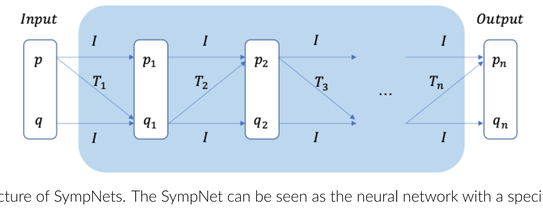


Figure 2. Architecture of SympNets. The SympNet can be seen as the neural network with a specific lower/upper triangular-type connection pattern, which guarantees symplecticity.

Symplectic graph neural networks (SGNNs)

The second goal is to embed permutation equivariance into the previous architecture via graph neural networks. Our strategy is to modify the existing linear and gradient module.

$$\varphi_n \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} I & 0/S_n \odot A \\ S_n \odot I/0 & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ S_2 \odot I & I \end{pmatrix} \begin{pmatrix} I & S_1 \odot A \\ 0 & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} + b, \tag{1}$$

$$\begin{aligned} \phi_{up} \begin{pmatrix} p \\ q \end{pmatrix} &= \begin{pmatrix} I & -\nabla_q V(\cdot, \boldsymbol{e}) \\ 0 & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}, \\ \phi_{low} \begin{pmatrix} p \\ q \end{pmatrix} &= \begin{pmatrix} I & 0 \\ \nabla_p T(\cdot, \boldsymbol{v}) & I \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix}, \end{aligned} \tag{2}$$

- For large particle systems it's more nature to exploit the graph structure in the data by parameterizing V or T as graph neural networks.
- Each node in the graph stores the position and momentum of a single particle.
- We build the graph based on the distance between particles: If the distance between two particles are within a cutoff radius r_c, we build an edge for them.
- A is the adjacency matrix and e are spring constants which can be either prefixed or trainable.
- Here V and T can be interpreted as the potential and kinetic energy of the system.

We consider argon particles governed by Lennard-Jones potential in 2d. We generate data with time step 10^{-14}s, then downsample to make $dt = 10^{-13}s$. Note if we simulate data directly with $dt = 10^{-13}$s, the energy of the system would explode using the same numerical integrator.

- Train with $x(0), x(dt), \cdots, x(1000dt)$. Goal: Predict $x(1001dt), x(1002dt), \cdots, x(2000dt)$
- NVE (microcarnonical ensemble) energy should be conserved.
- Number of particles = 2000.
- 19.7% percent reduction in prediction time compared to the numerical integrator.

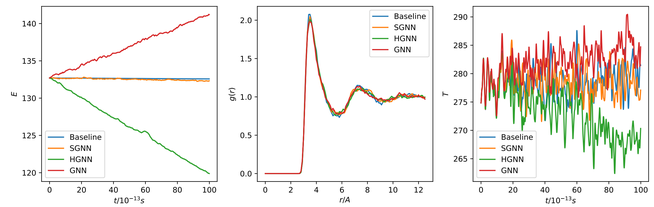
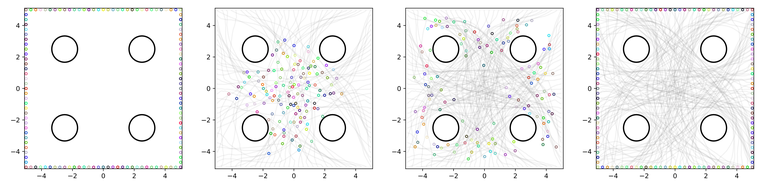


Figure 3. Statistical quantities of the predicted trajectories by GNN, HGNN and SGNN.

- Multi-agent optimal control formulation: $x(s)$ denotes the positions of all agents.

$$\min\{\int_0^T l(x(s), \dot{x}(s)) + \epsilon\beta_a(h(x(s)))ds : x(0) = x_0, x(T) = x_T\}$$

- Dimension is Mm ($M = \#$ agents, m is the dimension of the physical space)
- The optimal trajectory satisfies the Hamiltonian ODE with H directly computable from l via the Legendre transform.
- We apply SympNet to solve the Hamilton's ODE with initial and terminal conditions x_0 and x_T.



(a) $t = 0$ (b) $t = \frac{1}{3}$ (c) $t = \frac{2}{3}$ (d) $t = 1$

Figure 4. **Path planning of 128 drones with four obstacles.** We plot the predicted positions of 128 drones at time $t = 0, \frac{1}{3}, \frac{2}{3}, 1$. The four black circles represent the four obstacles, and the colored circles represent the drones.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., and Karniadakis, G. E. (2020). Sympnets: Intrinsic structure-preserving symplectic networks for identifying hamiltonian systems. *Neural Networks*, 132:166–179.

Meng, T., Zhang, Z., Darbon, J., and Karniadakis, G. (2022). Sympocnet: Solving optimal control problems with applications to high-dimensional multiagent path planning problems. *SIAM Journal on Scientific Computing*, 44(6):B1341–B1368.