FINDE: Neural Differential Equations for Finding and Preserving Invariant Quantities

Takashi Matsubara (Osaka University), Takaharu Yaguchi (Kobe University)

ICLR

Deep Learning with Laws of Physics

Deep learning (DL) for dynamical systems.

- The need for computer simulation is ubiquitous.
  - weather forecasting, drug discovery, vehicle design, etc.
- DL learns physical phenomena from observations, by learning the underlying differential equations like $\dot{x}=f(x)$.
- Physical systems often have first integrals (invariant quantities)
  - energy, momentum, holonomic constraints, mass, etc.
  - Incorporating first integrals into DL leads to a better accuracy.

No general method to preserve first integrals

- One method was proposed for each type of first integrals.
  - HNN [2], LieConv [3], CHNN [4], HNN++ [5], etc.
- Difficult to determine the type of first integral before learning.

Contributions of Proposal

- Finding and preserving any types of first integrals from observations.
- Available with previously proposed methods for some first integrals.
- Preserving first integrals without numerical errors in simulations.

Reference

[1] Chen et al., NeurIPS, 2018.
[2] Greydanus et al., NeurIPS, 2019.
[3] Finzi el al., ICML, 2020.
[4] Finzi el al., NeurIPS, 2020.
[5] Matsubara el al., NeurIPS, 2020.
[6] Hairer et al., Springer-Verlag, 2006.

Neural Networks for Finding and Preserving First Integrals

What are first integrals?

- A quantity $V(x)$ is a first integral of the system $\dot{x}=f(x)$ $\Leftrightarrow$ $V(x)$ is constant for any solution $x(t)$ (i.e., $\dot{V}(x(t))=0$)
- With first integrals $V_1,\dots,V_K$, any solution $x(t)$ is clamped to the submanifold $\mathcal{M}'=\{x\in\mathcal{M}: V_k(x(t))=V_k(x(0)) \text{ for } k=1,\dots,K\}$
- The time-derivative $\dot{x}$ is also clamped to the tangent space $T_x\mathcal{M}'=\{w\in\mathbb{R}^N: \nabla V_k(x)^\top w=0 \text{ for } k=1,\dots,K\}$

Numerical methods to preserve first integrals

- e.g., projection method [6]
  - Given a state x_n, predict the next $\hat{x}_{n+1}$ and project it onto $\mathcal{M}'$, that is, $\mathrm{argmin}_{x_{n+1}} \|x_{n+1}-\hat{x}_{n+1}\|$ s.t. $V_k(x_{n+1})=V_k(x_n)$
  - Need for solving an iterative optimization problem at every step.
  - Taking the limit as the time step $\Delta t\to 0$, we propose *First-Integral preserving Neural Differential Equation (FINDE).*

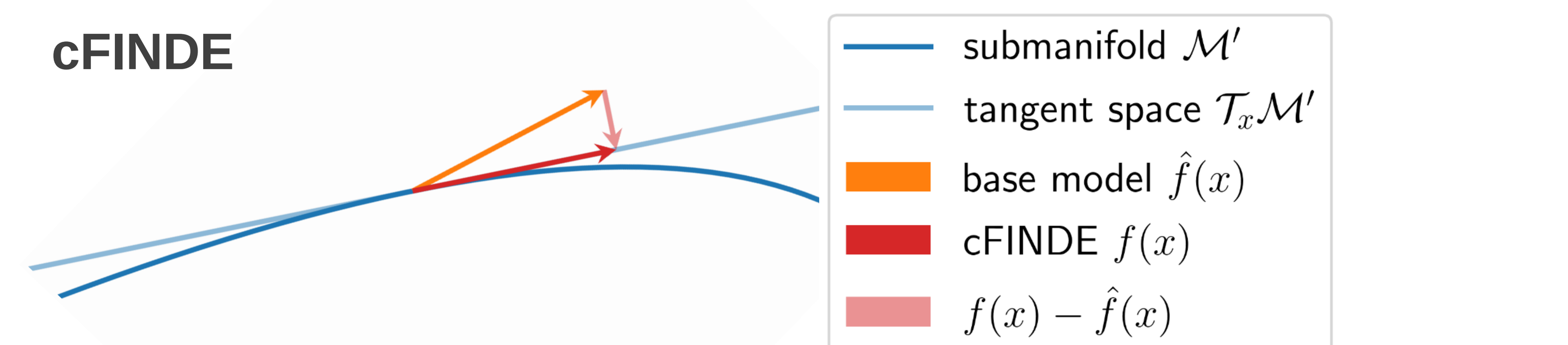
Ich FINDE dich!

- Learn the dynamics $\hat{f}(x)$ and first integrals $\boldsymbol{V}=(V_1 \dots V_K)^\top$ jointly, by projecting the former onto $T_x\mathcal{M}'$ defined by the latter.
- The continuous-time FINDE (cFINDE) $\dot{x}=f(x)$ preserves first integrals $\boldsymbol{V}$.
- Combined with numerical integrators, cFINDE $\dot{x}=f(x)$ no longer preserves first integrals $\boldsymbol{V}$ due to the temporal discretization errors.

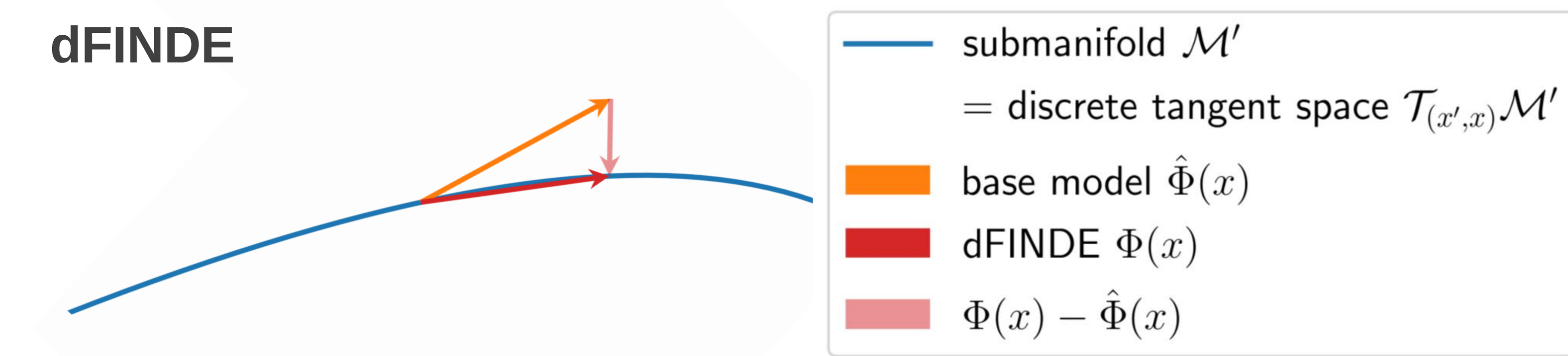
Overcoming temporal discretization errors

- We propose the discrete-time FINDE (dFINDE) .
  - Instead of the ODE $\dot{x}=\hat{f}(x)$,the difference equation $\frac{x'-x}{\Delta t}=\hat{\Phi}(x',x)$.
  - Instead of the gradient $\nabla V(x)$, the discrete gradient $\overline{\nabla} V(x',x)$.
  - Instead of $T_x\mathcal{M}'$, the discrete tangent space $T_{(x',x)}\mathcal{M}'=\{w\in\mathbb{R}^N: \overline{\nabla} V_k(x',x)^\top w=0 \text{ for } k=1,\dots,K\}$
- Then, project the discrete-time dynamics $\hat{\Phi}(x',x)$ onto $T_{(x',x)}\mathcal{M}'$.
- dFINDE $\frac{x'-x}{\Delta t}=\Phi(x',x)$ preserves the first integrals $\boldsymbol{V}$ in discrete time.
- No temporal discretization errors.
- No need for solving an optimization problem at the training phase.

cFINDE



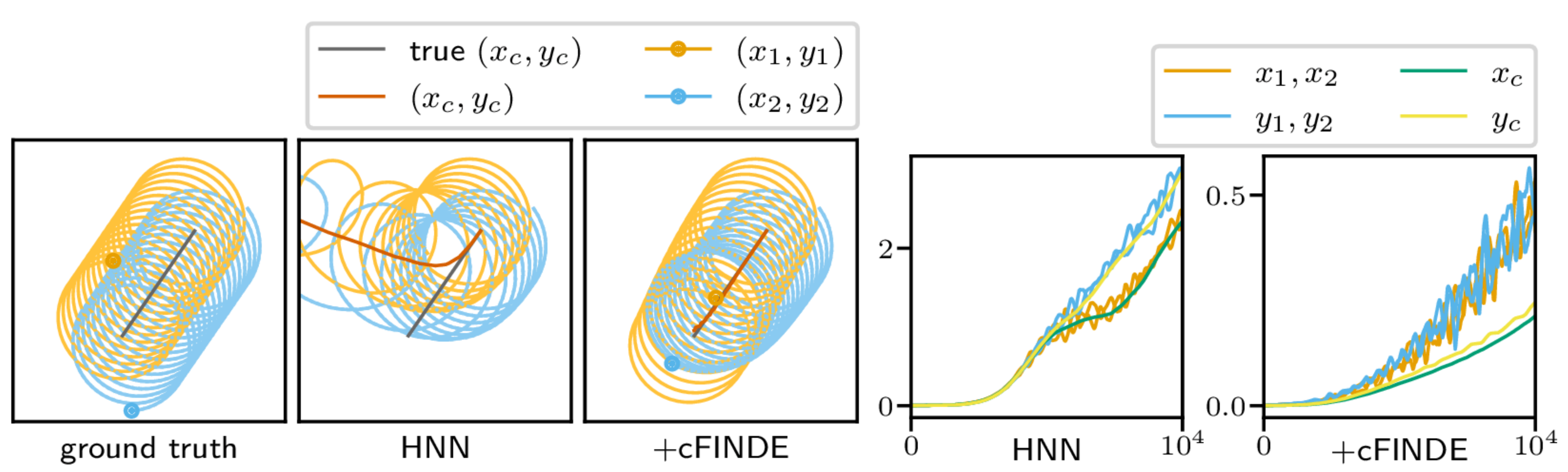
dFINDE



Experiments and Results

2D two-body problem

- Found and preserved the linear momenta in x- and y-directions
  - suggested by smaller errors of the center-of-mass x_c



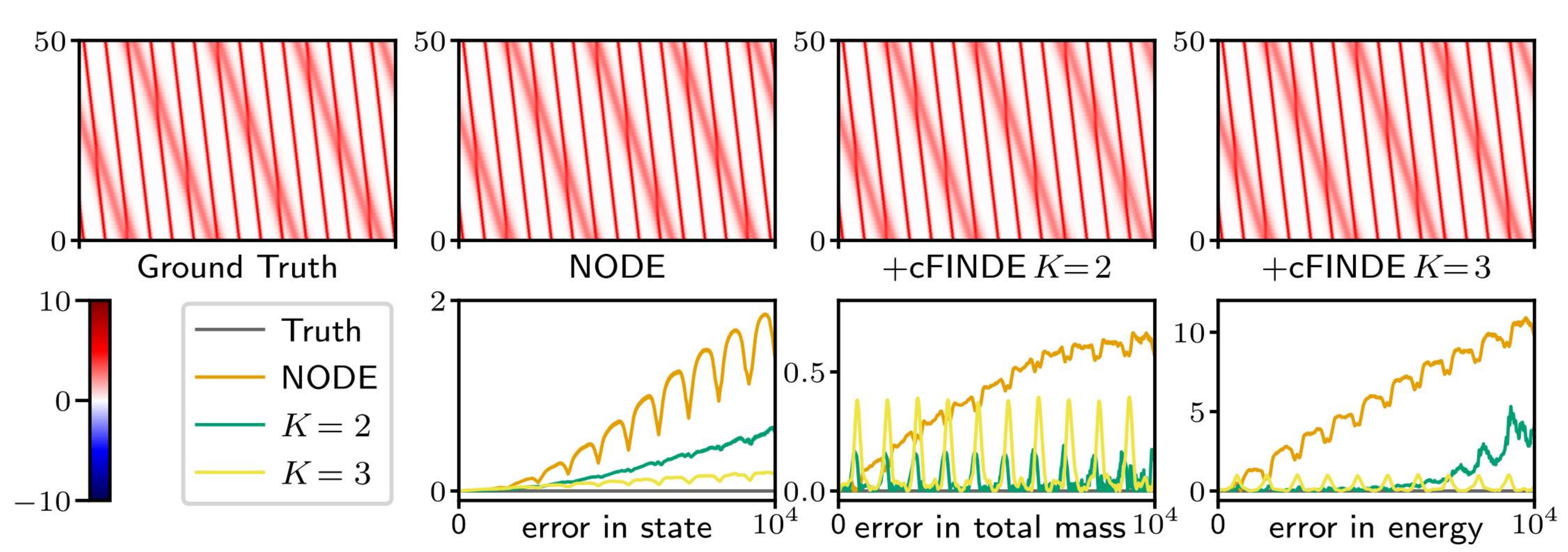
- Found first integrals were linear momenta
  - confirmed by symbolic regression

Trial	Training Data V_1	Training Data V_2	Test Data V_1	Test Data V_2
0	$v_{x1}+v_{x2}$	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}+\alpha$	$v_{y1}+v_{y2}$
1	$v_{x1}+v_{x2}$	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}$	$v_{y1}+v_{y2}$
2	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}$	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}$
3	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}$	$v_{y1}+v_{y2}$	$v_{x1}+v_{x2}$
4	$v_{x1}+v_{x2}-v_{y1}-v_{y2}$	$v_{x1}+v_{x2}+v_{y1}+v_{y2}$	$v_{x1}+v_{x2}-v_{y1}-v_{y2}$	$v_{x1}+v_{x2}+v_{y1}+v_{y2}$

We removed biases and scale factors. $\alpha=0.003(y_1+y_2)(v_{x2}+x_1+y_1(v_{x2}+y_1+y_2)+1.402)$.

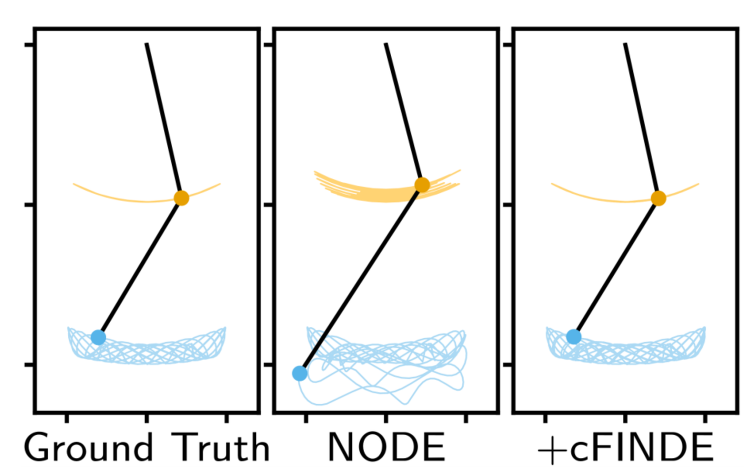
KdV equation

- preserved the total mass for $K\geq 2$ and the energy for $K\geq 3$.



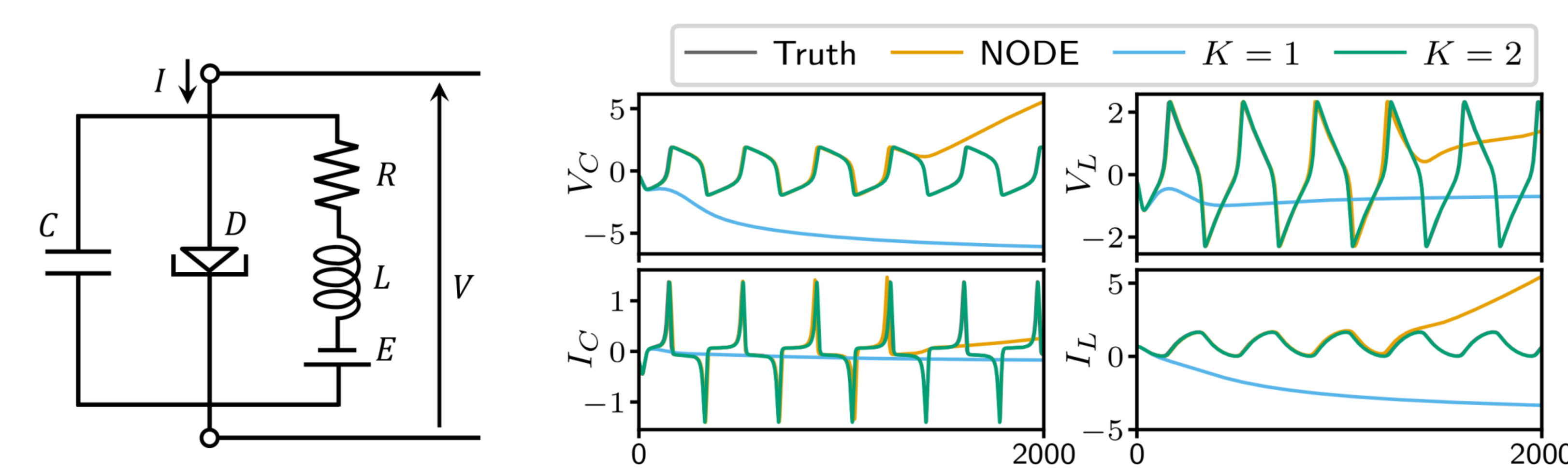
Double pendulum

- worked the best with $K=5$, suggesting all first integrals found.
  - energy
  - 2 holonomic constraints
  - 2velocity constraints



Circuit representation of FitzHugh-Nagumo model

- Kirchhoff's voltage and current laws produced two first integrals.
- Even in dissipative systems, FINDE found first integrals.



Better prediction accuracy

- Except for an unreasonably large number K of assumed first integrals.
- dFINDE further improves the accuracy (the valid prediction time).

	Two-body	KdV equation	double pendulum	FitzHugh-Nagumo
Base model	0.362	0.339	0.110	0.236
cFINDE	**0.450**	**0.730**	**0.585**	**0.437**
dFINDE	**0.475**	**0.780**	**0.591**	**0.455**