Pacific Northwest
NATIONAL LABORATORY

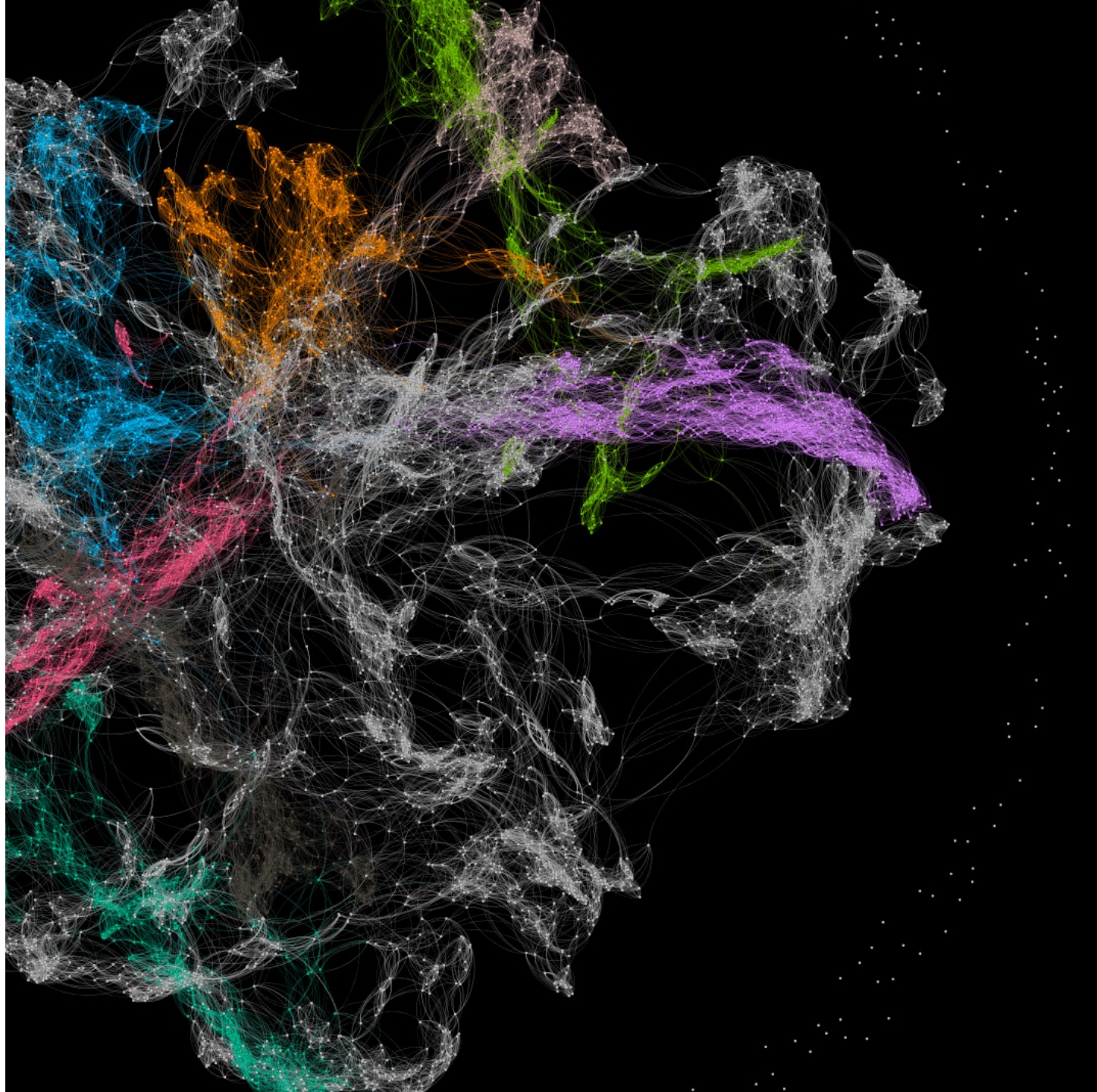
Neuromancer:
Differentiable Programming Library for Modeling, Control, and Optimization

The Sixth Autonomous Energy Systems, NREL Workshop, Golden, CO

September 18, 2023

Ján Drgoňa

Core developers: Aaron Tuor, James Koch, Madelyn Shapiro, Draguna Vrabie



U.S. DEPARTMENT OF ENERGY BATTELLE

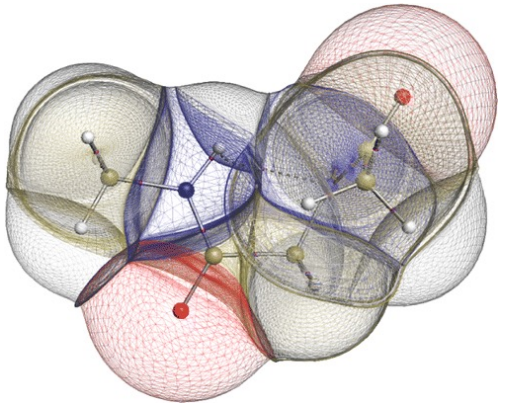
PNNL is operated by Battelle for the U.S. Department of Energy

Challenges of Dynamical Systems Modeling and Control

- **Simulations** are crucial for many areas of decision-making and scientific discovery
- **Need**: Improve computational efficiency and scalability for heterogenous scientific simulations
- **Challenges**:
  - Physically-consistent data-driven modeling
  - Fast simulation of complex systems
  - Optimal control and design of complex systems
- **Emerging solution**:
  - Scientific Machine Learning connecting physics and AI domains

Latest Neural Nets Solve World's Hardest Equations Faster Than Ever Before

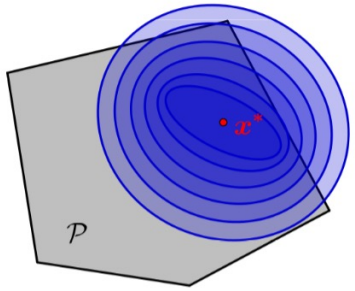
Quantamagazine



Landscape of Solution Methods

Constrained optimization

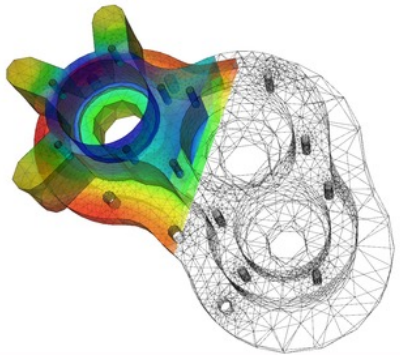
$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & b(x) \geq 0 \\ & c(x) = 0. \end{aligned}$$



- Requires prior knowledge of objective function and constraints

Differential equations

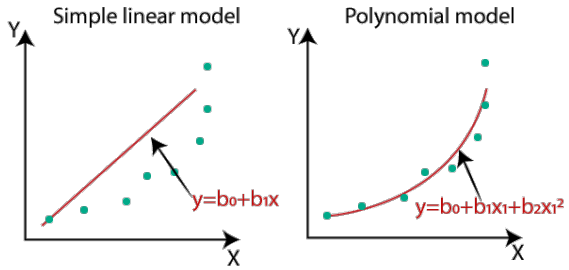
$$\frac{dy}{dx} = f(x)$$



- Requires prior knowledge of the physics to be modeled

Supervised Learning

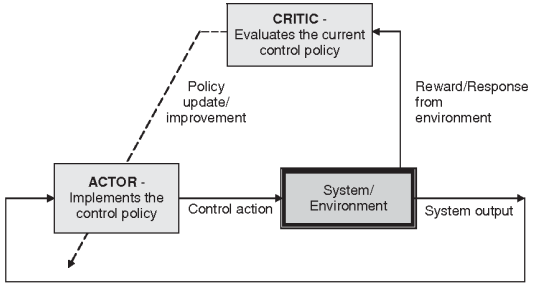
$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y^i, f(x^i, \theta))$$



- Requires large labeled datasets

Reinforcement Learning

$$\begin{aligned} \min_{\Theta} \quad & \sum_{i=1}^{m} \mathbf{r}(\mathbf{x}, \Theta) \\ \text{s.t.} \quad & \mathbf{Bellman}(\mathbf{x}, \Theta) = \mathbf{0}, \\ & \mathbf{environment}(\mathbf{x}, \Theta) = \mathbf{0} \\ & \mathbf{x} \in \Xi \end{aligned}$$



- Requires environment model to sample

More domain knowledge

Less domain knowledge

Pacific Northwest NATIONAL LABORATORY

Landscape of Solution Tools

Constrained optimization

Differential Equations

Supervised Learning

Reinforcement Learning

PYOMO

JUMP

GEKKO DYNAMIC OPTIMIZATION

CVXPY

CasADi

GUROBI OPTIMIZATION

DifferentialEquations.jl

MATLAB®

SciPy

PETSc TAO

NWCHEM

PyTorch

TensorFlow

flux

Gym

More domain knowledge

Less domain knowledge

Pacific Northwest NATIONAL LABORATORY

Landscape of Solution Tools

Online optimization

PYOMO

JUMP

GEKKO DYNAMIC OPTIMIZATION

CVXPY

CasADi

GUROBI OPTIMIZATION

Differential Equations

DifferentialEquations.jl

MATLAB®

SciPy

PETSc TAO

NWCHEM

Supervised Learning

PyTorch

TensorFlow

flux

Reinforcement Learning

Gym

What comes next? … Differentiable programming (DP): a unifying approach for data-driven modeling and optimization of complex systems based on automatic differentiation (AD)

Differentiable Programming Enables Scientific Machine Learning

- **Differentiable Programming**
  - M. Innes, et al., *A Differentiable Programming System to Bridge Machine Learning and Scientific Computing*, 2019
- **Physics-informed Neural Networks**
  - M. Raissi, et al., *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, 2019
- **Neural Differential Equations**
  - R. T. Q. Chen, et al., *Neural Ordinary Differential Equations*, 2019
  - C. Rackauckas , et al., *Universal Differential Equations for Scientific Machine Learning*, 2021
- **Differentiable Optimization**
  - A. Agrawal, et al., Differentiable Convex Optimization Layers, 2019
  - P. Donti, et al., *DC3: A learning method for optimization with hard constraints*, 2021
  - S. Gould, et al., *Deep Declarative Networks: A New Hope*, 2020
  - J. Kotary, et al., *End-to-End Constrained Optimization Learning: A Survey*, 2021
- **Differentiable Control**
  - B. Amos, et al., *Differentiable MPC for End-to-end Planning and Control*, 2019
  - S. East, et al., *Infinite-Horizon Differentiable Model Predictive Control*, 2020

NeuroMANCER: a Scientific Machine Learning Library

1. Mathematical formulation

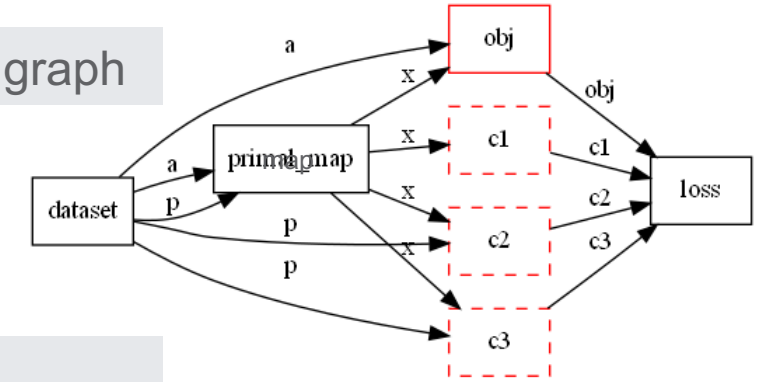
$$\min_{\Theta} (1-\mathbf{x})^2 + \boldsymbol{p}(\mathbf{y}-\mathbf{x}^2)^2$$
$$\text{s.t. } (\boldsymbol{p}/2)^2 \leq \mathbf{x}^2 + \mathbf{y}^2 \leq \boldsymbol{p}^2, \quad \mathbf{x} \geq \mathbf{y}$$
$$\mathbf{x} = \boldsymbol{\pi}_{\Theta}(\boldsymbol{p})$$

2. Python code interface

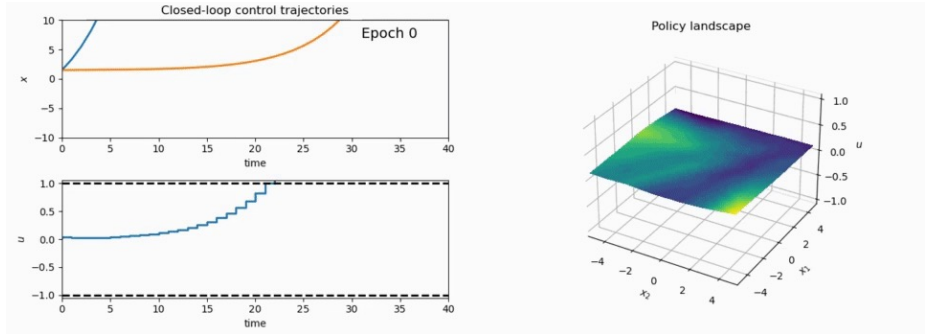
PyTorch

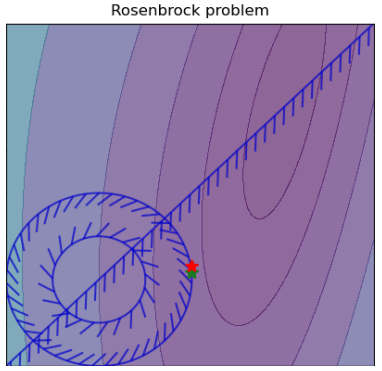
```
import neuromancer as nm

p = nm.variable('p')
x = nm.variable('x')
y = nm.variable('y')

obj = ((1-x)**2 + p*(y-x**2)**2).minimize(weight=1.0, name='obj')
c1 = (p/2)**2 <= x**2 + y**2
c2 = x**2 + y**2 <= p**2
c3 = x >= y

net = nm.MLP(insize=2, outsize=2, hsizes=[80]*4)
map = nm.Node(net, input_keys=['p'], output_keys=['x','y'])

loss = nm.PenaltyLoss([obj], [c1, c2, c3])
problem = nm.Problem([map], loss)
optimizer = torch.optim.AdamW(problem.parameters())
trainer = nm.Trainer(problem,data,optimizer)
best_model = trainer.train()
```

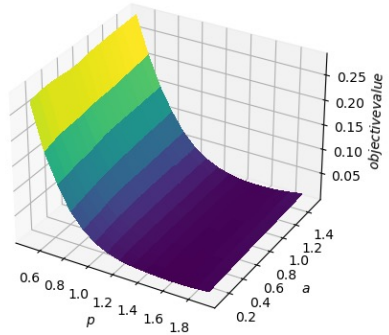
3. Problem graph



4. Results

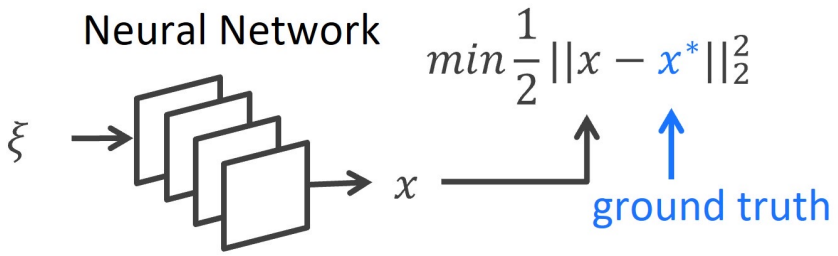




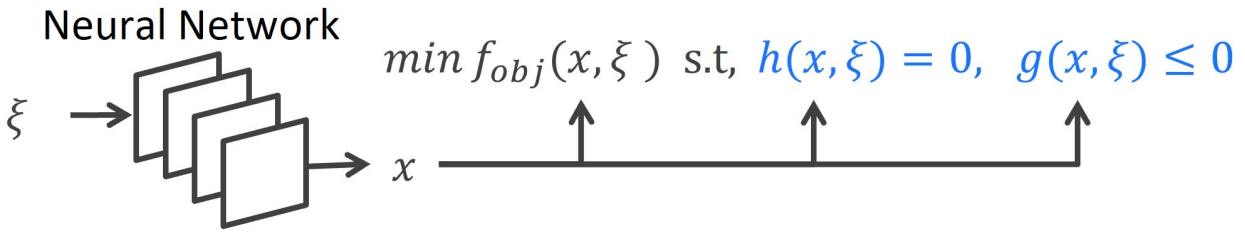


Parametric Constrained Optimization Capabilities in Neuromancer

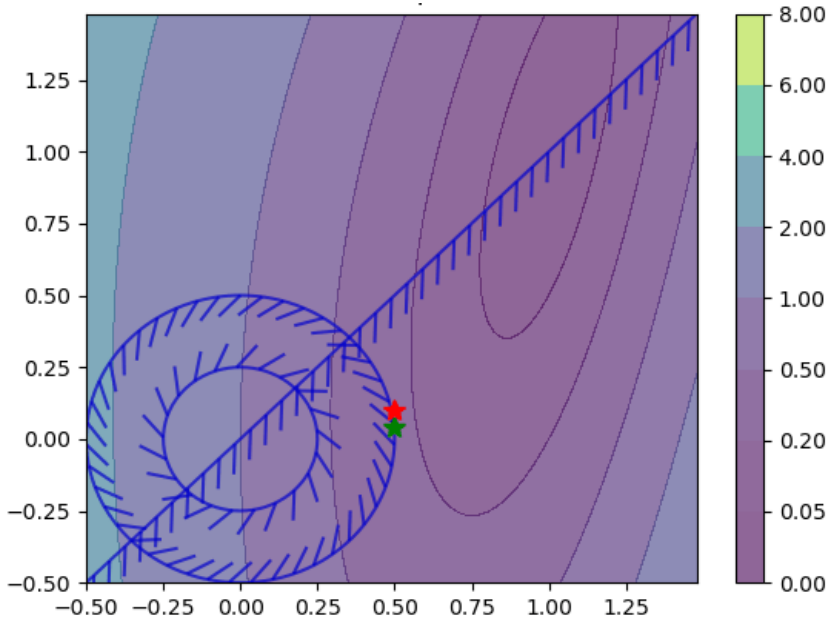
Imitation Learning



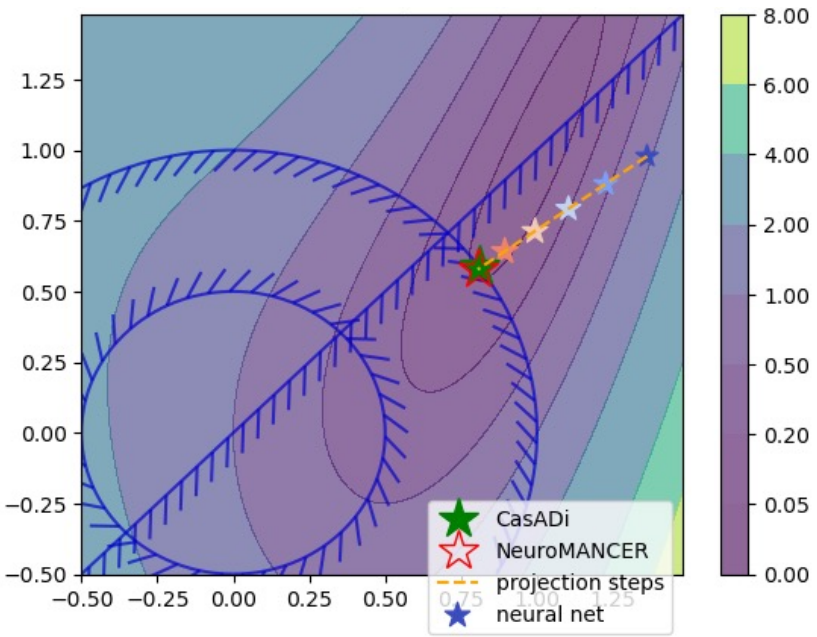
Differentiable Parametric Programming
or, Constrained Deep Learning, End-to-end NN



Training neural networks as explicit solutions

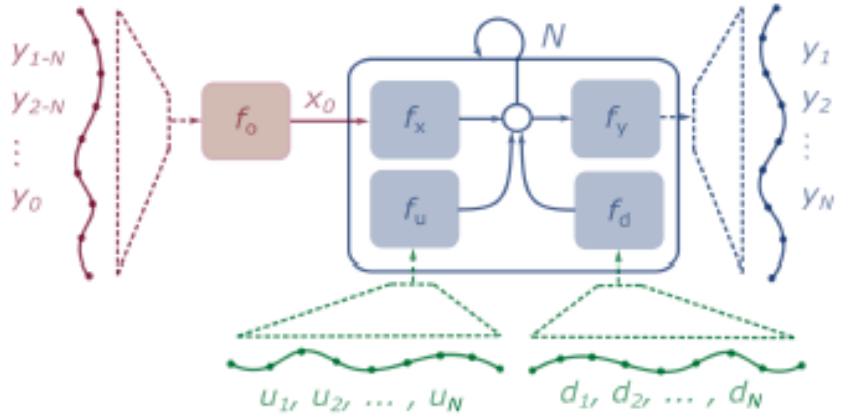


Feasibility restoration with implicit layers

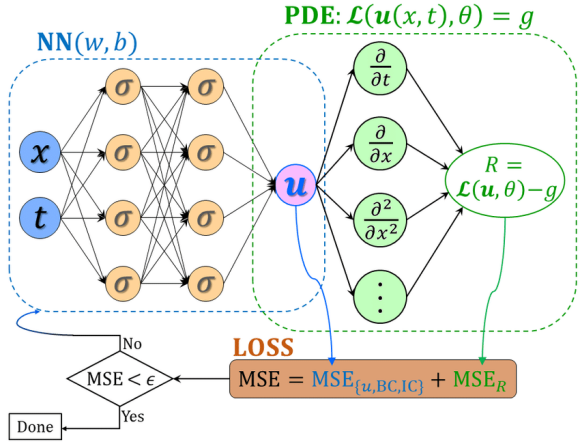


Data-driven Modeling Capabilities in Neuromancer

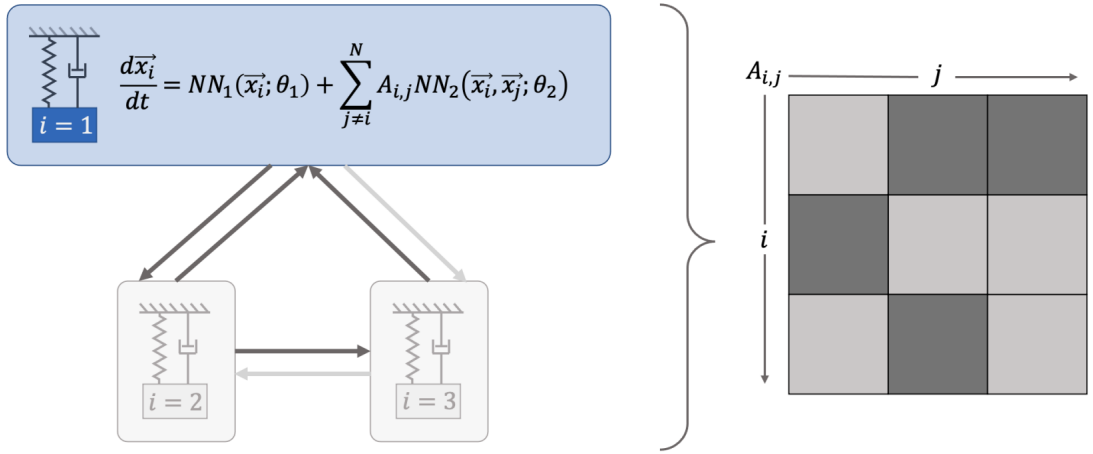
Component-based Physics-informed Machine Learning



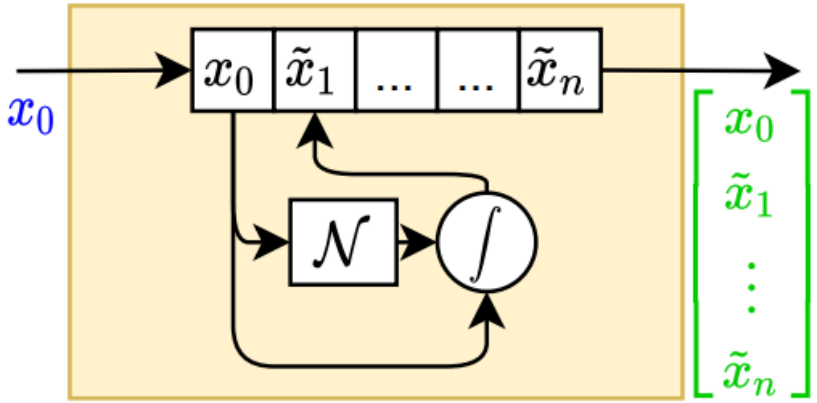
Physics-Informed Neural Networks



Networked Dynamical systems

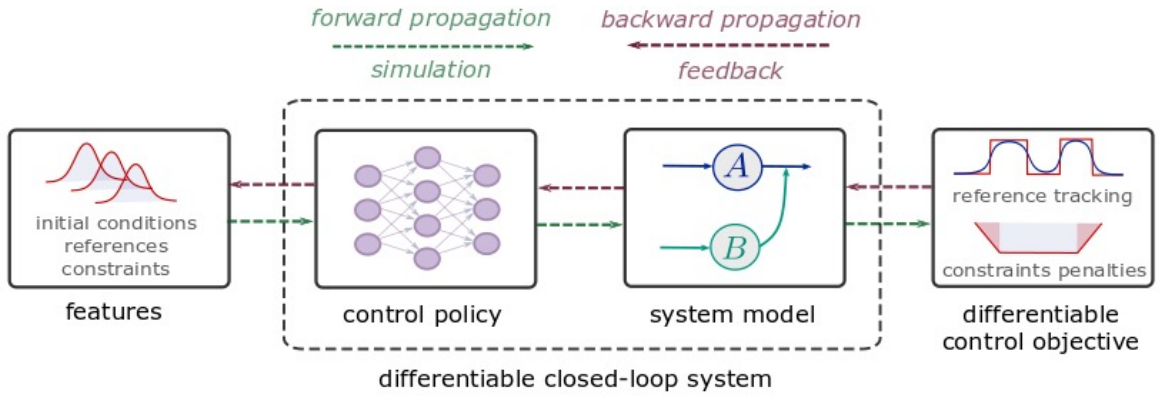


Neural differential equations

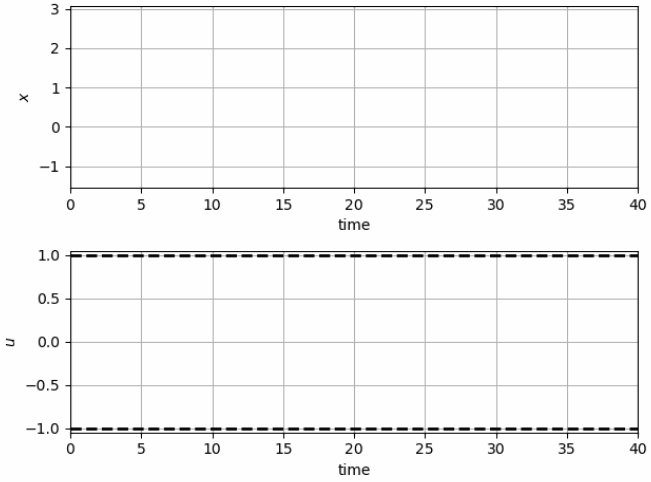


Optimal Control Capabilities in Neuromancer

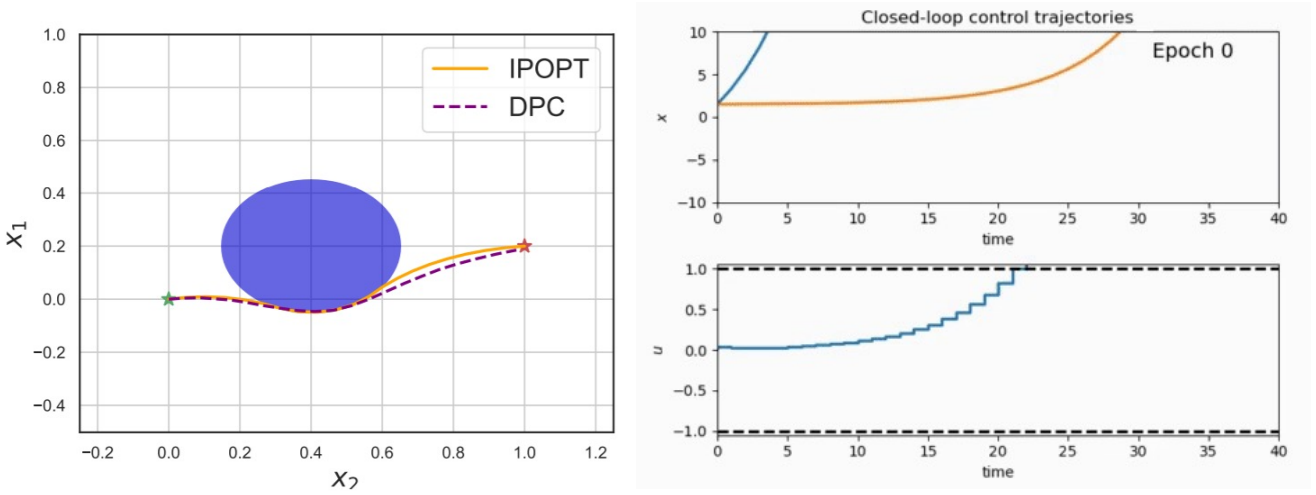
Learning constrained model-based control policies with differentiable control methods



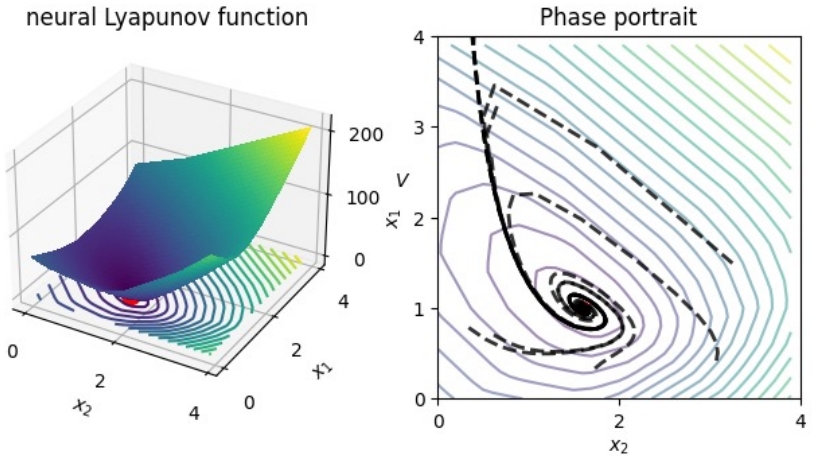
Learning stabilizing controllers



Trajectory optimization and obstacle avoidance



Learning Neural Lyapunov Functions



NeuroMANCER

Open-source scientific machine learning (**SciML**) toolbox in PyTorch for integrating deep learning, constrained optimization, and physics-based modeling

- **Physics-informed machine learning**
- **Data-driven modeling of dynamical systems**
- **Model-based policy optimization**
- **Parametric constrained optimization**

github.com/pnnl/neuromancer

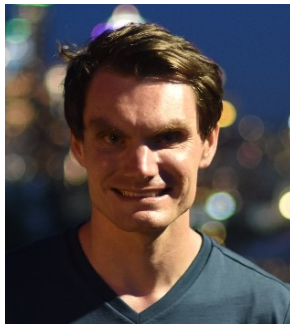
Pacific Northwest
NATIONAL LABORATORY

Acknowledgements

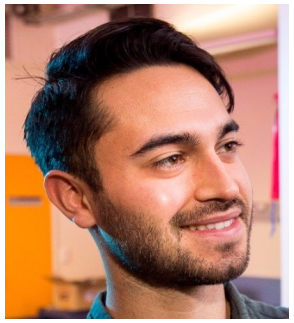

Aaron Tuor


James Koch


Madelyn Shapiro


Stefan Dernbach


Christian M. Legaard


Wenceslao Shaw Cortez


Ethan King


Shrirang Abhyankar


Mahantesh Halappanavar


Draguna Vrabie

U.S. DEPARTMENT OF ENERGY

Pacific Northwest
NATIONAL LABORATORY