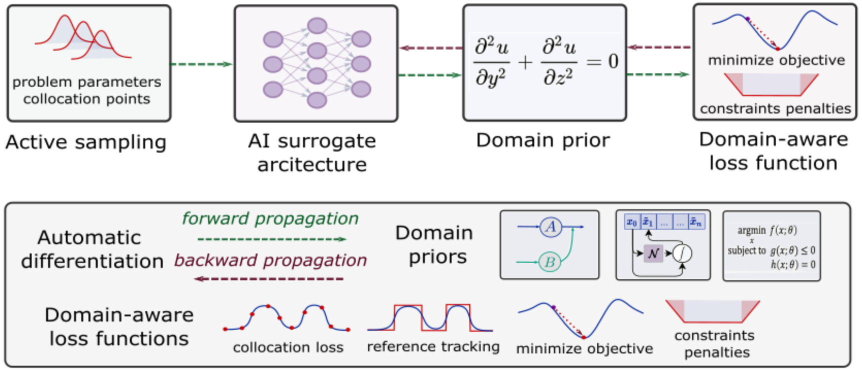


Tutorial on Scientific Machine Learning for Modeling, Optimization, and Control

Thomas Beckers, Truong X. Nghiem, Ján Drgoňa

8th Annual Learning for Dynamics & Control Conference
University of Southern California
June 17

Organizers

Thomas Beckers

VANDERBILT UNIVERSITY

Truong X. Nghiem

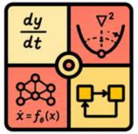
University of Central Florida

Ján Drgoňa

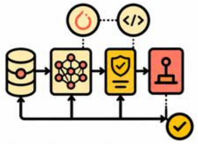
JOHNS HOPKINS UNIVERSITY

Goals

Understand the SciML paradigm as a unified framework that integrates physical laws, optimization, and control principles into machine learning models



Explore four key learning paradigms: learning to solve, optimize, model, and control, and understand their theoretical foundations and practical applications



Develop hands-on skills for building SciML learning pipelines using NeuroMANCER, PyTorch, and JAX

All slides and code examples are shared on
https://github.com/SciML-L4DC2026

Scientific Machine Learning (SciML)

What?

- SciML systematically integrates ML methods with mathematical models and algorithms developed in various scientific and engineering domains

Why?

- Scientific applications are governed by fundamental principles and physical constraints
- Purely data-driven "black box" ML methods cannot satisfy underlying physics

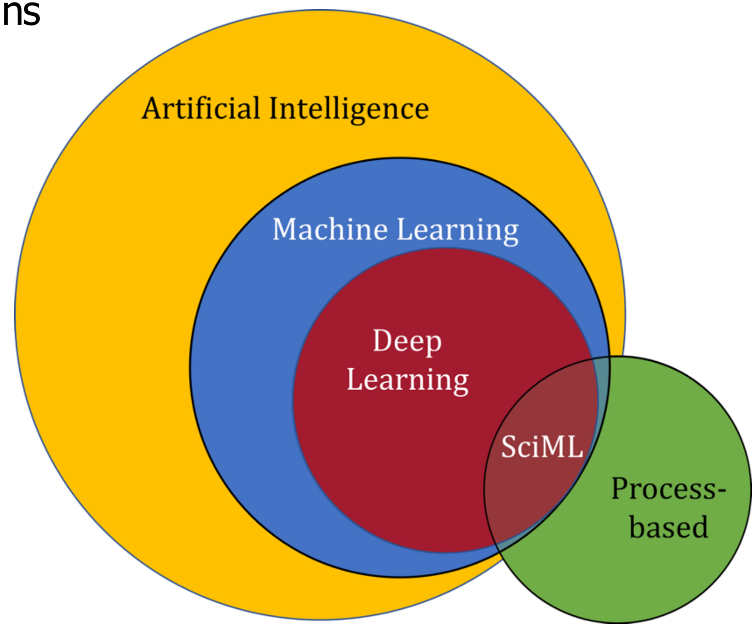
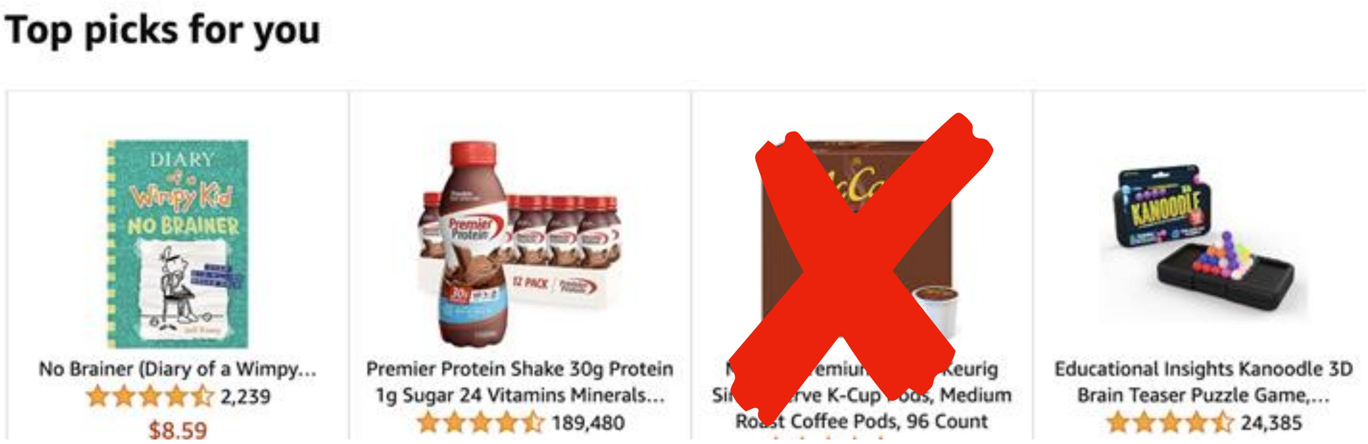


Image source: https://sciml.wur.nl/reviews/sciml/sciml.html

Why… for us?



But we need more for

Autonomous driving

[Electric Motor Engineering]

Human-robot collaboration

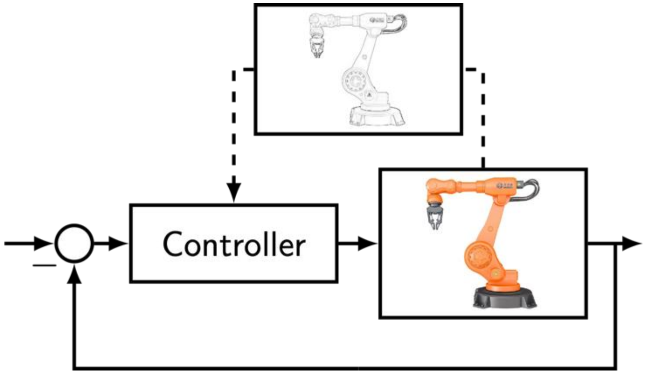
[Ars Electronica]

Power grids

[Chris Hunkeler]

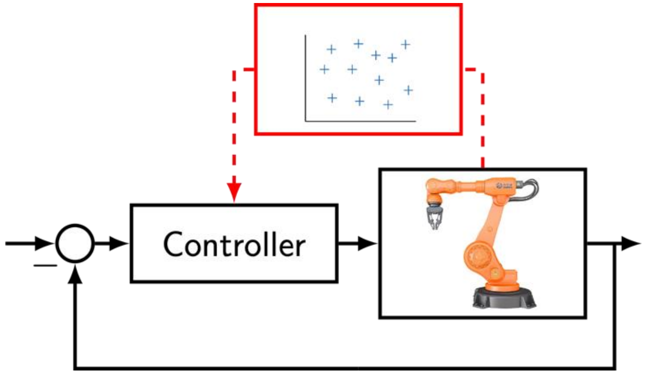
Modeling, optimization, and control of physical systems

Traditional



- ✓ Generalization
- ✓ Stability and performance
- ✓ No or little data
- ✗ Model selection
- ✗ Time-consuming (modeling and solving)

Purely data-driven

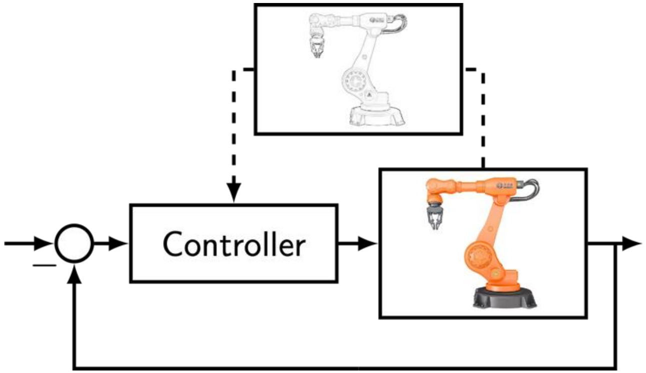


- ✓ Highly expressive
- ✓ Minimal expert knowledge
- ✗ Generalization and interpretability
- ✗ Amount of data
- ✗ Stability, performance, ...

Combining the best of both worlds

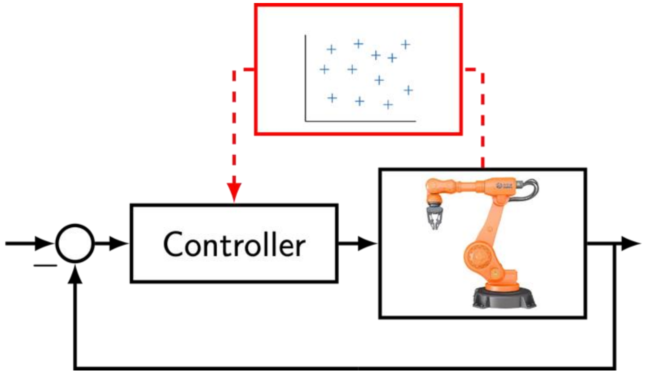
Scientific Machine Learning

Traditional



+

Purely data-driven



SciML

- Integrates physical knowledge, dynamical models, and data
- Enables learning with limited or noisy observations
- Improves sample efficiency and generalization
- Preserves physical consistency and interpretability
- Accelerates simulation, optimization, and control

Do we need different methods?

Complexity

LTI
system

Nonlinear, time-varying
system

Multi-scale, multi-physics,
nonlinear, …

Amount of data

Sparse identification

Bayesian models

Neural networks

Purpose

Identifying unknown physics/
governing equations?

Data efficiency

Interpretability

Modeling/Control/Optimization

Acceleration

There is not “the one” approach

Opportunities in control

Optimization

Constructing surrogate SciML models for the hard-to-optimize physics-based optimization problems or cost functions

Modeling & Control

Modeling of high dimensional and distributed physics in multi-physics systems.

Generation of structured SciML controllers for multicomponent systems

Assurances

Providing safety and performance guarantees for a broad class of learning-based control methods.

T. X. Nghiem *et al.*, "Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems," *2023 American Control Conference (ACC)*, 2024
J. Drgona *et al.*, "Safe Physics-Informed Machine Learning for Dynamics and Control," *2025 American Control Conference (ACC)*, 2025

Challenges

How to

- unify the terminology in SciML? (enhanced, informed, constrained, guided, encoded)
- avoid training failures of SciML models by getting stuck in local optima
- detecting & removing inappropriate physics bias
- verify methods for SciML to be scaled up for large-scale systems
- reduce the computational requirements without sacrificing accuracy

- balance between physics-driven and data-driven based modeling and optimization
- quantify minimal data requirements for training of SciML models and controllers
- quantify the uncertainty and modeling errors for SciML-based approaches
- effectively select representative training data for sampling-based SciML approaches
- guarantee stability and safety of a real-world system in closed-loop with SciML-based controllers

T. X. Nghiem *et al.*, "Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems," *2023 American Control Conference (ACC)*, 2024
J. Drgona *et al.*, "Safe Physics-Informed Machine Learning for Dynamics and Control," *2025 American Control Conference (ACC)*, 2025

Brief History of SciML: Early Foundations

Neural solvers for ODEs and PDEs: *represent and learn solutions with NNs.*

- E. Lagaris et al., *"Artificial neural networks for solving ordinary and partial differential equations,"* in IEEE Trans. Neural Networks, 1998
- McFall, K. S. & Mahan, J. R., *"Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions,"* in IEEE Trans. Neural Networks, 2009
- Beidokhti, R. S. & Malek, A., *"Solving initial-boundary value problems for systems of partial differential equations using neural networks and optimization techniques,"* in J. Franklin Inst., 2009
- Weinan E & B. Yu, *"The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems,"* in Commun. Math. Stat. 6, 2018
- J. Sirignano et al., *"DGM: A deep learning algorithm for solving partial differential equations,"* in Journal of computational physics, 2018

Data-driven equation discovery: *discover governing equations directly from data.*

- [Symbolic reg.] J. Bongard & H. Lipson, *"Automated reverse engineering of nonlinear dynamical systems,"* in Proc. NAS, 2007
- [Symbolic reg.] M. Schmidt & H. Lipson, *"Distilling free-form natural laws from experimental data,"* in Science, 2009
- [SINDy] S. Brunton et al., *"Discovering governing equations from data by sparse identification of nonlinear dynamical systems,"* in PNAS, 2016
- [SINDy for PDEs] S. H. Rudy et al., *"Data-driven discovery of partial differential equations,"* in Science Advances, 2017

Learning for optimization: *approximate expensive optimization routines with ML*

- [Algo. Unrolling] K. Gregor & Y. LeCun, *"Learning Fast Approximations of Sparse Coding,"* in ICML, 2010
- [Unrolling, Diff. Opt] J. Domke, *"Generic Methods for Optimization-Based Modeling,"* in AISTATS, 2012
- [Diff. Opt] D. Maclaurin et al., *"Gradient-Based Hyperparameter Optimization through Reversible Learning,"* in ICML, 2015

Brief History of SciML: Physics Enters ML

2017-2020 – Three major ideas emerged:

- Physics-informed learning (PINNs)
- Differentiable differential equations (Neural ODEs)
- Differentiable optimization layers

Physics-informed learning for ODEs and PDEs

Learn solutions, enforcing equations and constraints.

- M Raissi et al., *“Numerical Gaussian processes for time-dependent and nonlinear partial differential equations,”* in SIAM Journal on Scientific Computing, 2018
- M Raissi, *“Deep hidden physics models: Deep learning of nonlinear partial differential equations,”* in JMLR, 2018
- M Raissi, et al., *“Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear PDEs,”* in Journal of Computational Physics, 2019

Neural Differential Equations

Deep networks as continuous-time differential equations, differentiable ODE solvers.

- [Neural ODEs] R. T. Q. Chen et al., *“Neural ordinary differential equations,”* in NeurIPS, 2018
- [Neural SDEs] X. Li et al., *“Scalable Gradients for Stochastic Differential Equations,”* in AISTATS, 2020

Differentiable Optimization Layers

Optimization solver as end-to-end trainable differentiable layer

- [Diff. Opt] B. Amos & J. Z. Kolter, *"OptNet: Differentiable Optimization as a Layer in Neural Networks,"* in ICML, 2017
- [Diff. Opt] A. Agrawal et al., *"Differentiable Convex Optimization Layers,"* in NeurIPS, 2019

Brief History of SciML: Modern SciML

Since 2020, modern SciML has seen many developments, such as

Hybrid Physics + Machine Learning

Embed ML inside mechanistic differential equations, unifying mechanistic modeling and ML.

- [UDEs] C. Rackauckas et al., *"Universal differential equations for scientific machine learning,"* in arXiv:2001.04385, 2020
- [NDAEs] James Koch, et al., *"Learning Neural Differential Algebraic Equations via Operator Splitting*", CDC, 2025

Operator Learning

Learn maps between function spaces.

- [DeepONet] L. Lu, et al., *"Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators,"* in Nature Machine Intelligence, 2021
- [FNO] Z. Li et al., *"Fourier Neural Operator for Parametric Partial Differential Equations,"* in ICLR, 2021

Learning to Optimize

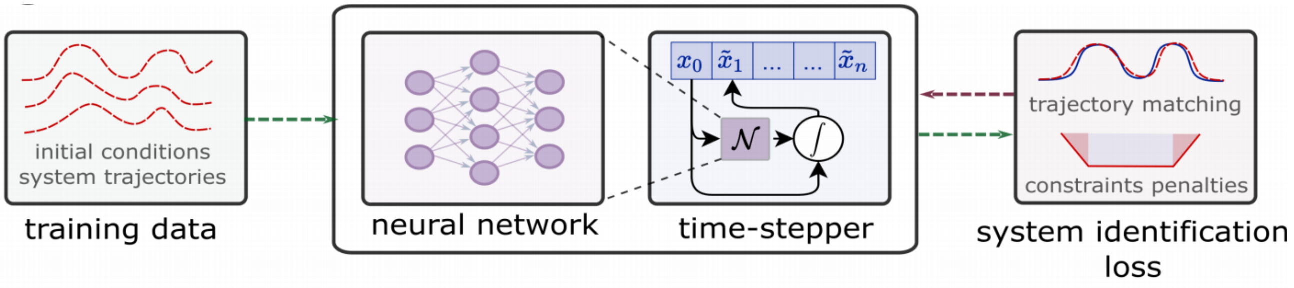
Learn maps from problem parameters to optimal solutions

- P. Donti et al., *"DC3: A learning method for optimization with hard constraints,"* in ICLR, 2021
- J. Kotary et al., *"End-to-End Constrained Optimization Learning: A Survey,"* in IJCAI, 2021

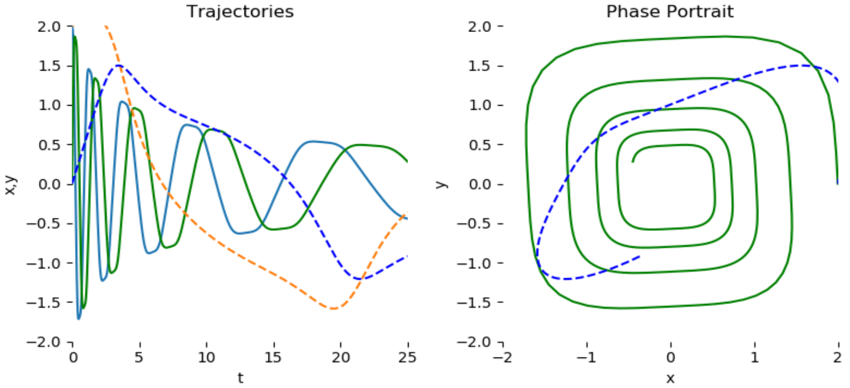
SciML emerged from the convergence of these foundations: neural ODE/PDE solvers, equation discovery, PIML, differentiable optimization, differentiable dynamical systems, operator learning,…

Enable today's SciML vision of *Learning to Model (L2M), Learning to Solve (L2S), Learning to Optimize (L2O), and Learning to Control (L2C)*.

Learning to Model (L2M) Dynamical Systems



Neural models for nonlinear system identification



Application: climate modeling

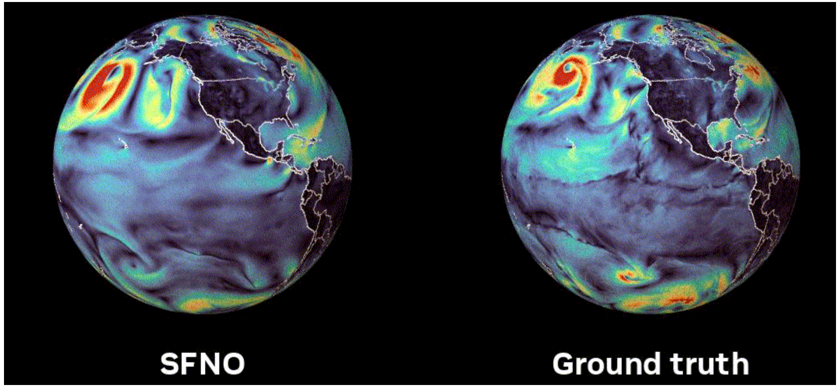
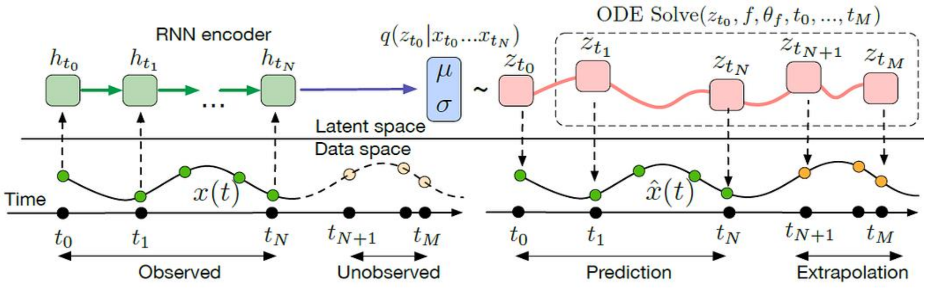


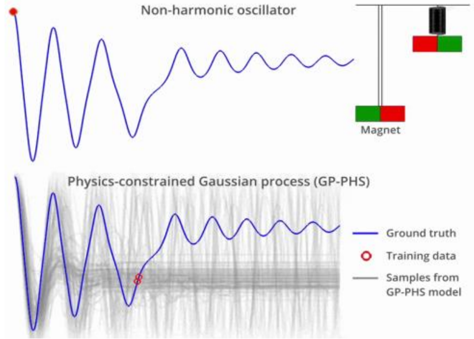
Image: NVIDIA FourCastNet

R. T. Q. Chen, et al., Neural ordinary differential equations. NeurIPS, 2018
C. Rackauckas, et al., Universal Differential Equations for Scientific Machine Learning, 2021
Z. Li et al., Fourier Neural Operator for Parametric Partial Differential Equations, in ICLR, 2021

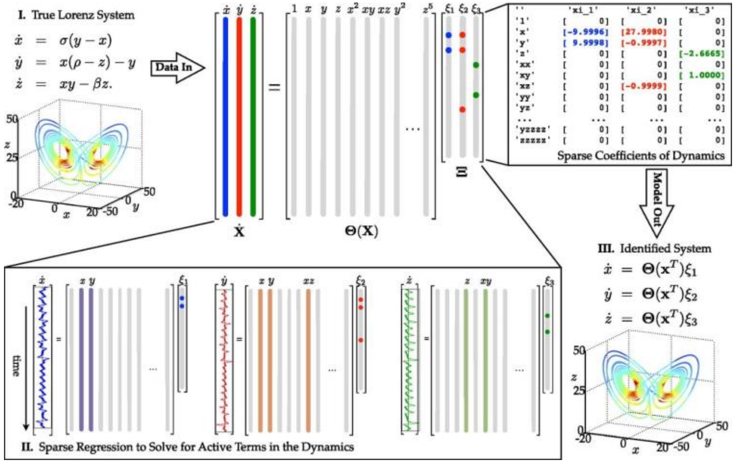
Learning to Model (L2M) Dynamical Systems



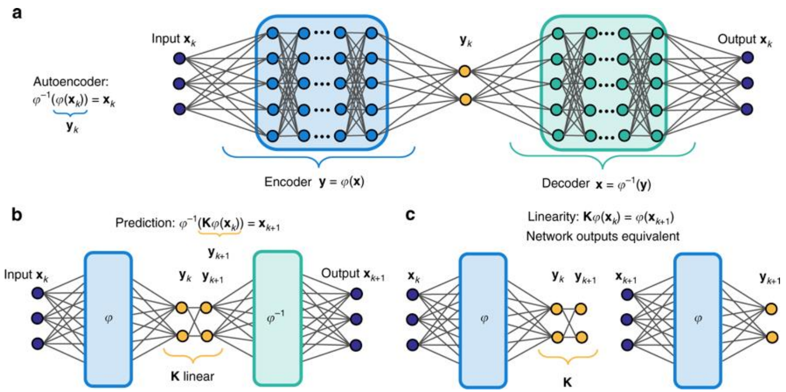
R. T. Q. Chen et al., *Neural Ordinary Differential Equations*, NeurIPS, 2019



T. Beckers et al., *Gaussian Process Port-Hamiltonian Systems: Bayesian Learning with Physics Prior*, CDC, 2022

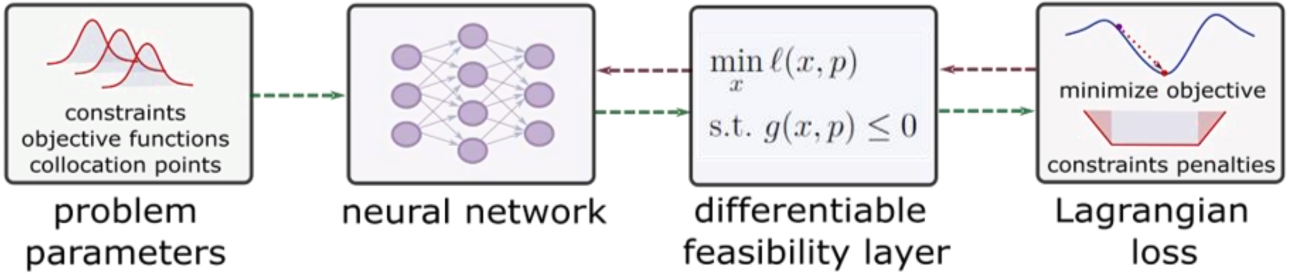


S. Brunton et al., *"Discovering governing equations from data by sparse identification of nonlinear dynamical systems,"* PNAS, 2016

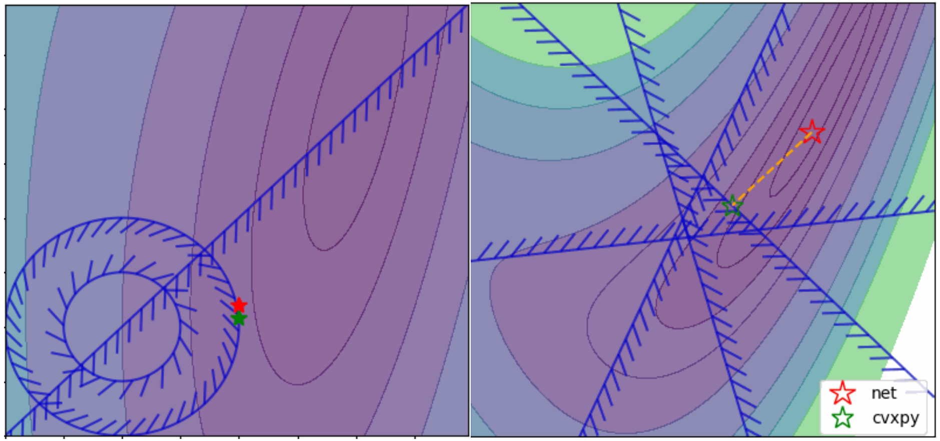


B. Lusch, et al., *Deep learning for universal linear embeddings of nonlinear dynamics*, Nature Comm., 2018

Learning to Optimize (L2O) with Constraints



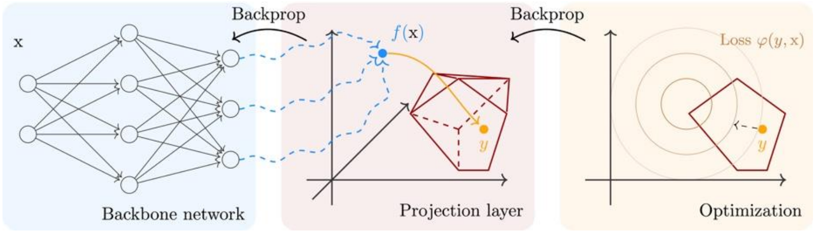
Training neural networks as optimization solutions



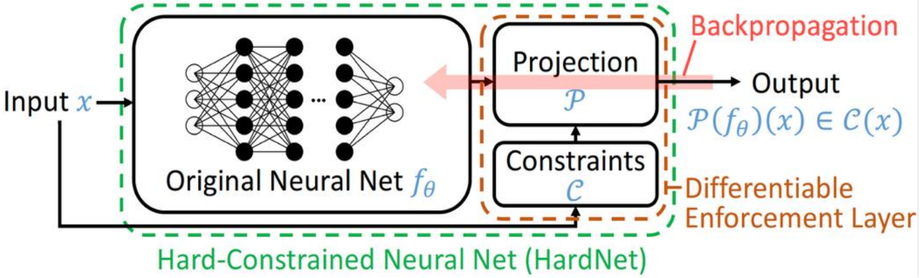
Application: solving optimal power flow

James Kotary, et al., *End-to-End Constrained Optimization Learning: A Survey*, IJCAI, 2021
Pascal Van Hentenryck, Optimization learning, arXiv:2501.03443, 2025

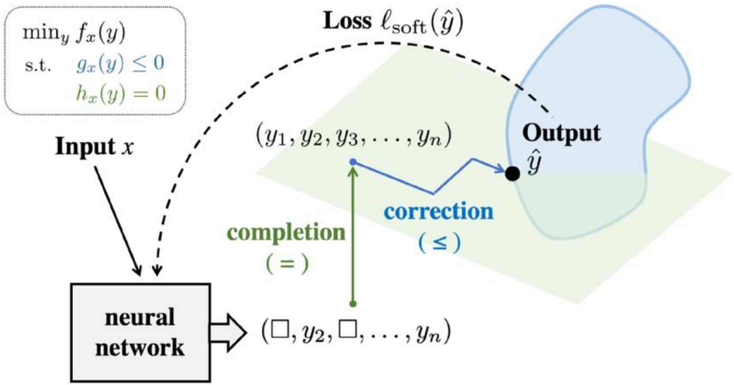
Learning to Optimize (L2O) with Constraints



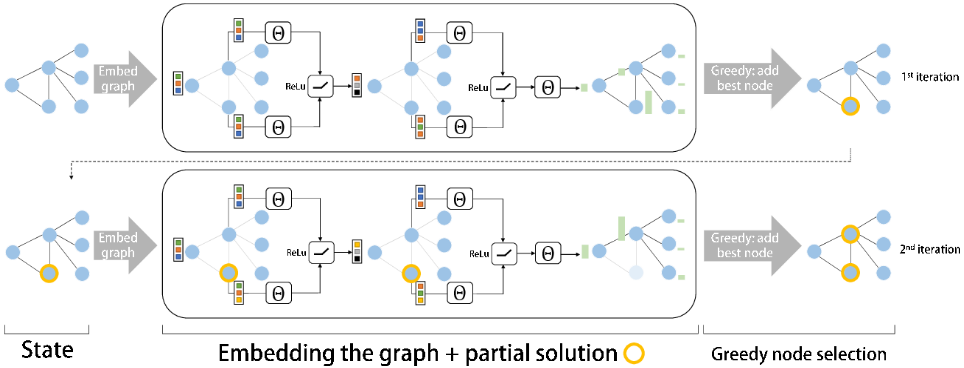
PD Grontas et al., *"Pinet: Optimizing hard-constrained neural networks with orthogonal projection layers,"* arXiv, 2025



Y. Min & N. Azizan, *"HardNet: Hard-Constrained Neural Networks with Universal Approximation Guarantees,"* JMLR, 2025

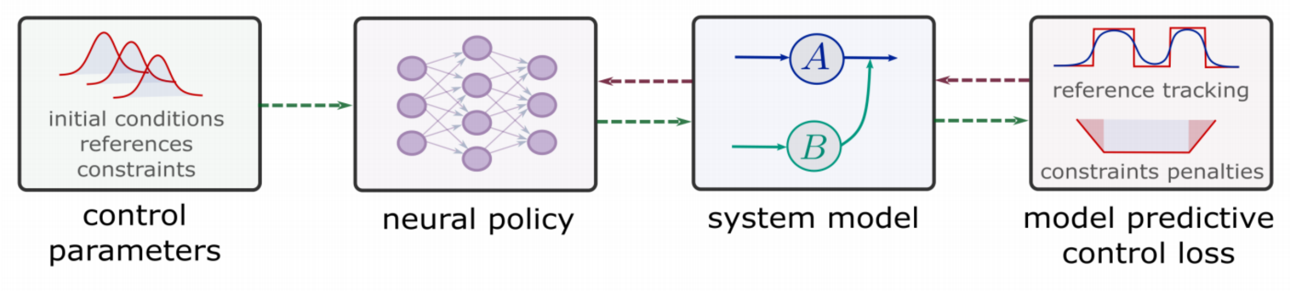


P. Donti, et al., "*DC3: A learning method for optimization with hard constraints,*" 2021

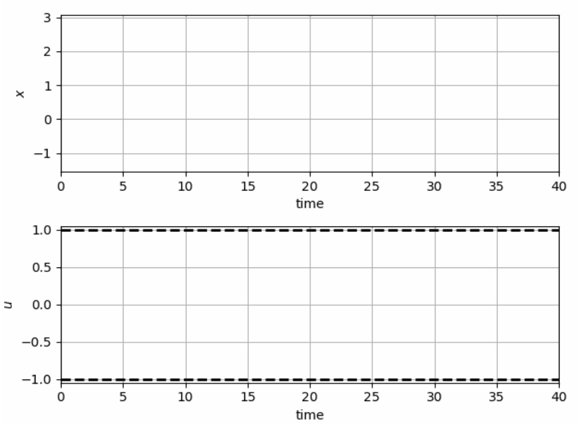


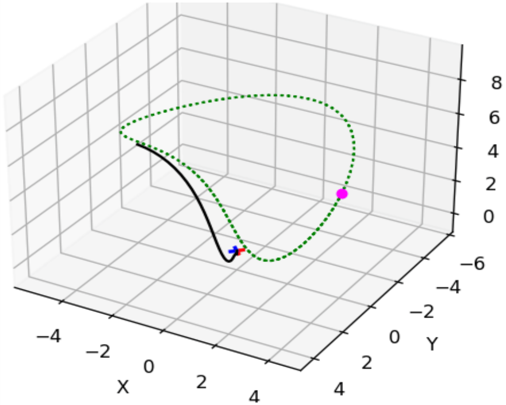
E. Khalil et al., *"Learning combinatorial optimization algorithms over graphs,"* NeurIPS, 2017

Learning to Control (L2C)

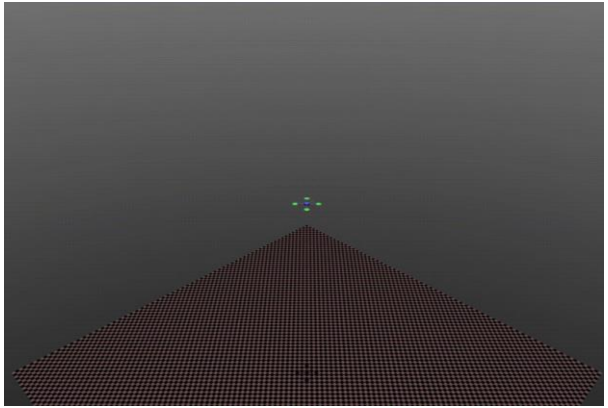


Trajectory optimization for dynamical systems





Application: autonomous systems



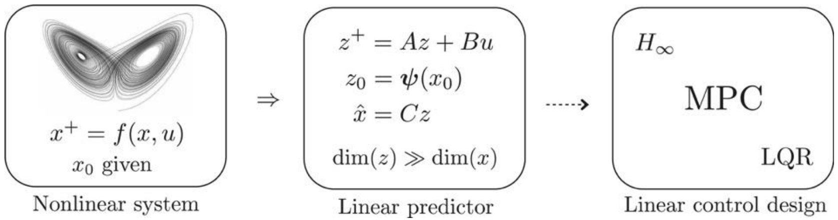
L. Hewing, et al., Learning-based model predictive control: Toward safe learning in control, Annual Review of Control, Robotics, and Autonomous Systems, 2020

L Brunke, et al., Safe learning in robotics: From learning-based control to safe reinforcement learning, Annual Review of Control, Robotics, and Autonomous Systems, 2021

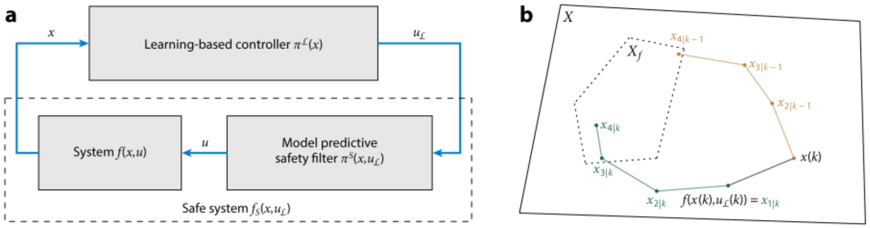
T. X. Nghiem et al., Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems, American Control Conference (ACC), 2023

J. Drgoňa et al., Safe Physics-informed Machine Learning for Dynamics and Control, American Control Conference (ACC), 2025

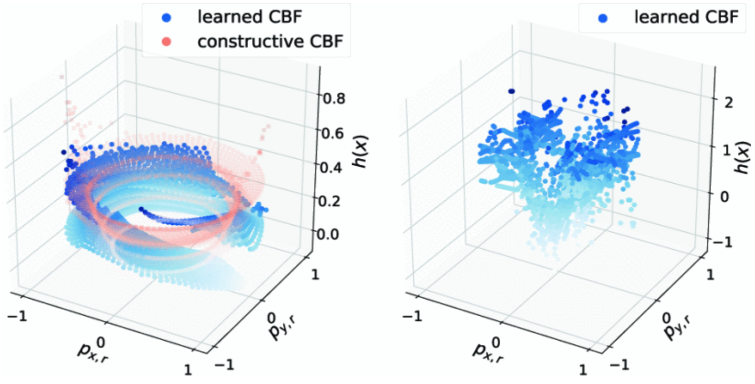
Learning to Control (L2C)



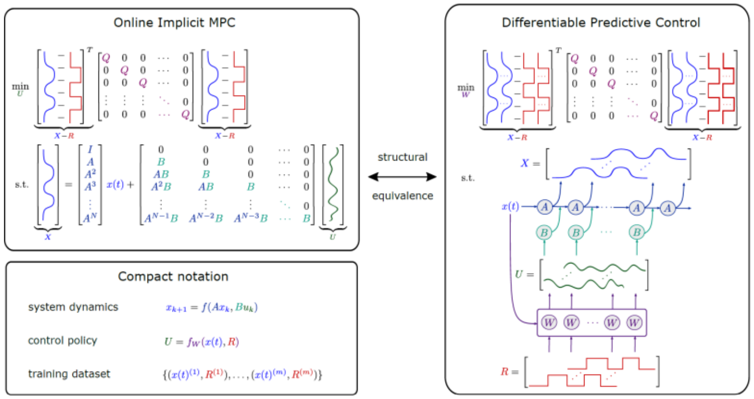
M. Korda & Igor Mezić, *"Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,"* Automatica, 2018



L. Hewing et al., *"Learning-based model predictive control: Toward safe learning in control,"* Ann. Rev. of Con., Rob. and Aut. Sys., 2020

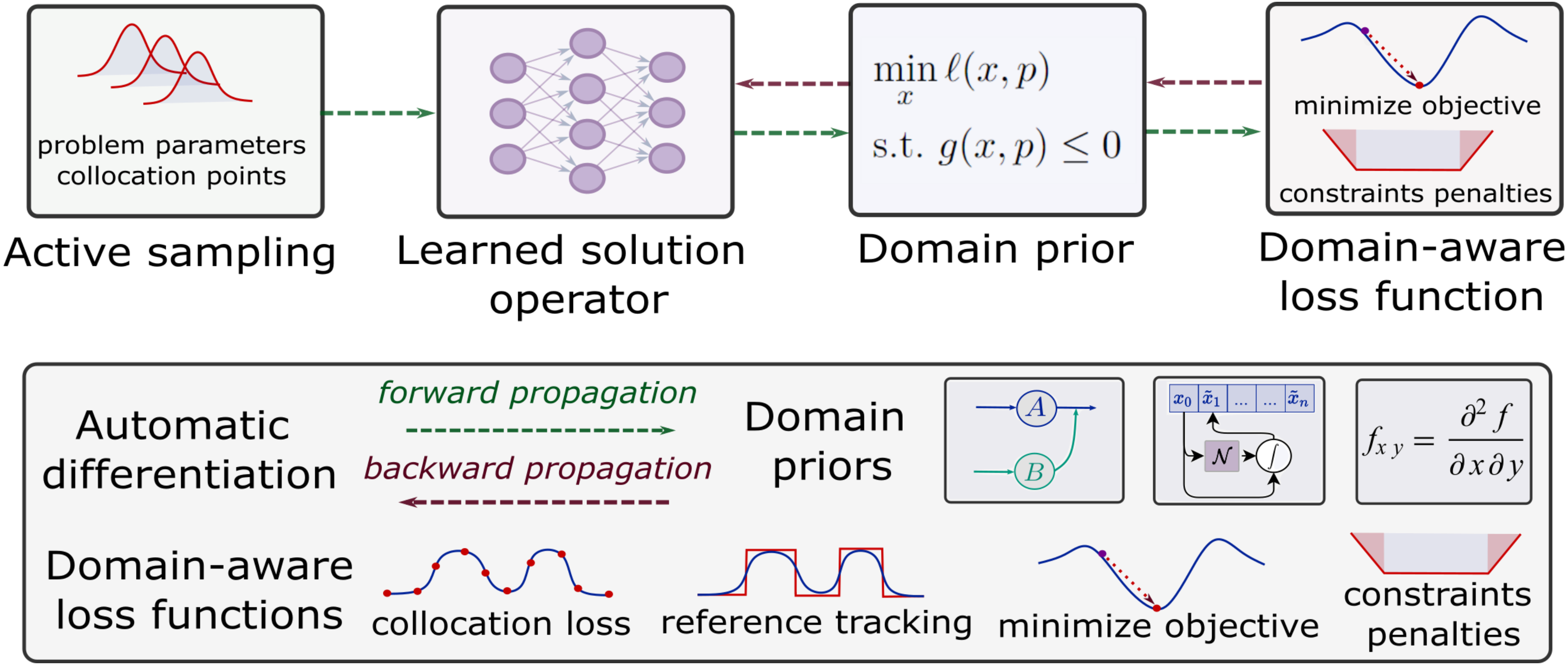


A Robey et al., *"Learning control barrier functions from expert demonstrations,"* CDC, 2020



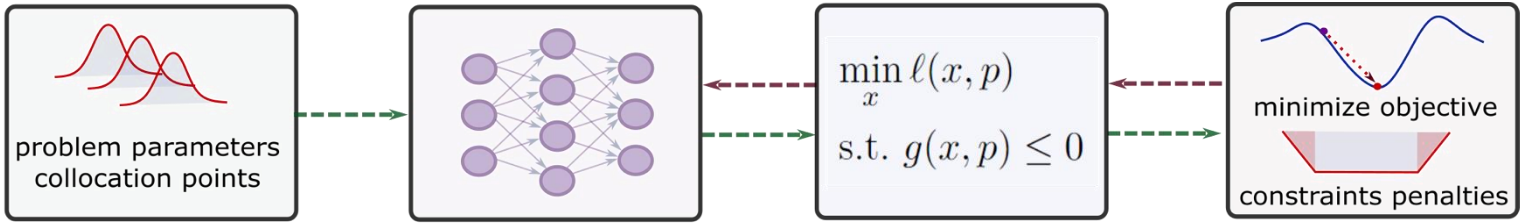
J. Drgoňa et al., *"Learning Constrained Parametric Differentiable Predictive Control Policies With Guarantees,"* in IEEE TSMC: Systems, 2024

Components of Scientific Machine Learning



Karniadakis, G.E., Kevrekidis, I.G., Lu, L. et al. Physics-informed machine learning. Nat Rev Phys 3, 2021.
Thiyagalingam, J., Shankar, M., Fox, G. et al. Scientific machine learning benchmarks. Nature Reviews Physics 4, 413–420, 2022.
Nghiem T., Drgona J., et al. Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems, ACC, 2023.

Generalized Scientific Machine Learning Problem



Generalized Scientific Machine Learning Problem

Given parameters, data, or scenarios $\xi \in \Omega$, find

$$u^\star(\cdot;\xi) = \arg\min_{u \in \mathcal{U}} \mathcal{J}(u;\xi)$$

subject to governing equations and auxiliary constraints

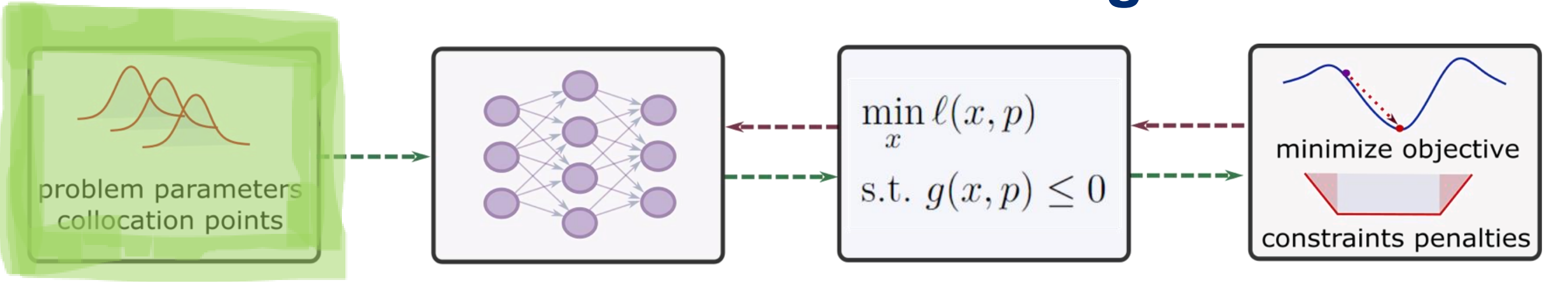
$$\mathcal{N}(u;\xi) = 0, \qquad \mathcal{B}(u;\xi) \leq 0.$$

The mapping

$$\mathcal{S}_\theta : \xi \mapsto u^\star(\cdot;\xi)$$

defines the parametric solution operator with trainable parameters θ.

Generalized Scientific Machine Learning Problem



Generalized Scientific Machine Learning Problem

Given parameters, data, or scenarios $\xi \in \Omega$, find

$$u^\star(\cdot;\xi) = \arg\min_{u\in\mathcal{U}} \mathcal{J}(u;\xi)$$

subject to governing equations and auxiliary constraints

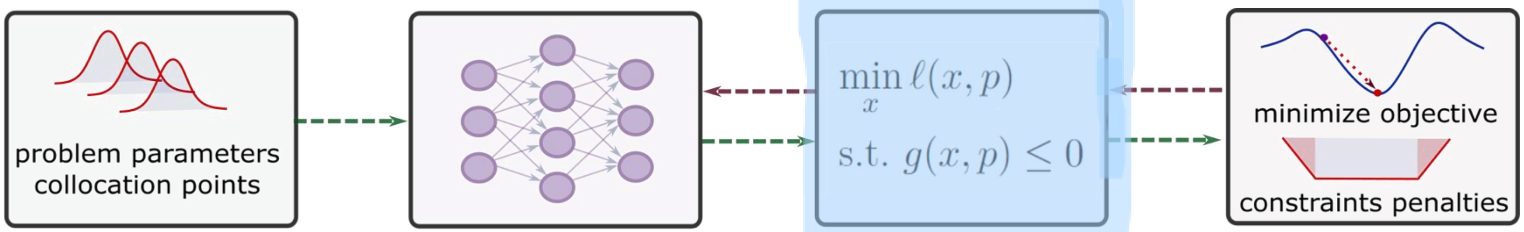
$$\mathcal{N}(u;\xi) = 0, \qquad \mathcal{B}(u;\xi) \leq 0.$$

The mapping

$$\mathcal{S}_\theta : \xi \mapsto u^\star(\cdot;\xi)$$

defines the parametric solution operator with trainable parameters θ.

Generalized Scientific Machine Learning Problem



Generalized Scientific Machine Learning Problem

Given parameters, data, or scenarios $\xi \in \Omega$, find

$$u^\star(\cdot;\xi) = \arg\min_{u \in \mathcal{U}} \mathcal{J}(u;\xi)$$

subject to governing equations and auxiliary constraints

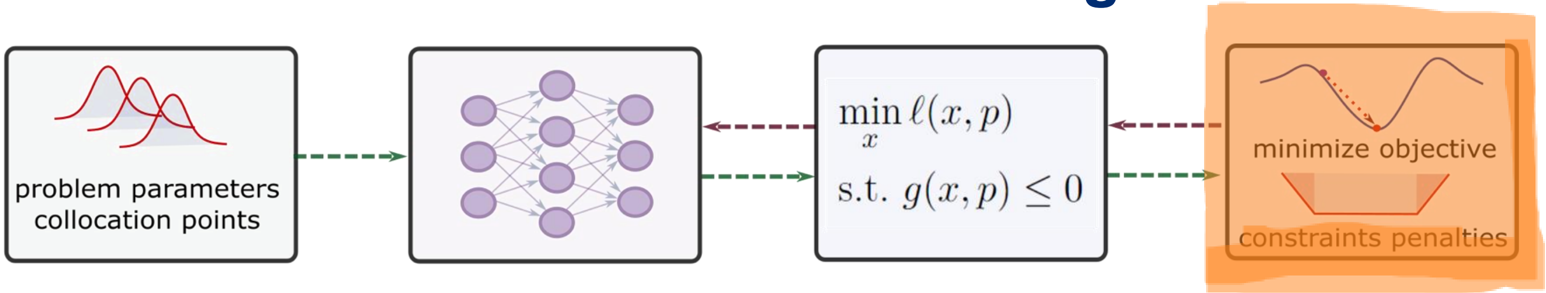
$$\mathcal{N}(u;\xi) = 0, \qquad \mathcal{B}(u;\xi) \leq 0.$$

The mapping

$$\mathcal{S}_\theta : \xi \mapsto u^\star(\cdot;\xi)$$

defines the parametric solution operator with trainable parameters θ.

Generalized Scientific Machine Learning Problem



Generalized Scientific Machine Learning Problem

Given parameters, data, or scenarios $\xi \in \Omega$, find

$$u^\star(\cdot;\xi) = \arg\min_{u \in \mathcal{U}} \mathcal{J}(u;\xi)$$

subject to governing equations and auxiliary constraints

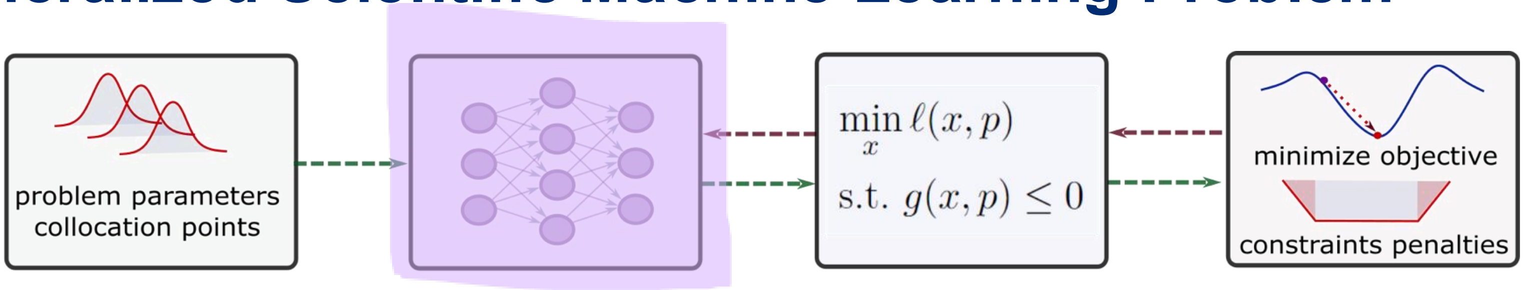
$$\mathcal{N}(u;\xi) = 0, \qquad \mathcal{B}(u;\xi) \leq 0.$$

The mapping

$$\mathcal{S}_\theta : \xi \mapsto u^\star(\cdot;\xi)$$

defines the parametric solution operator with trainable parameters θ.

Generalized Scientific Machine Learning Problem



Generalized Scientific Machine Learning Problem

Given parameters, data, or scenarios $\xi \in \Omega$, find

$$u^\star(\cdot;\xi) = \arg\min_{u \in \mathcal{U}} \mathcal{J}(u;\xi)$$

subject to governing equations and auxiliary constraints

$$\mathcal{N}(u;\xi) = 0, \qquad \mathcal{B}(u;\xi) \leq 0.$$

The mapping

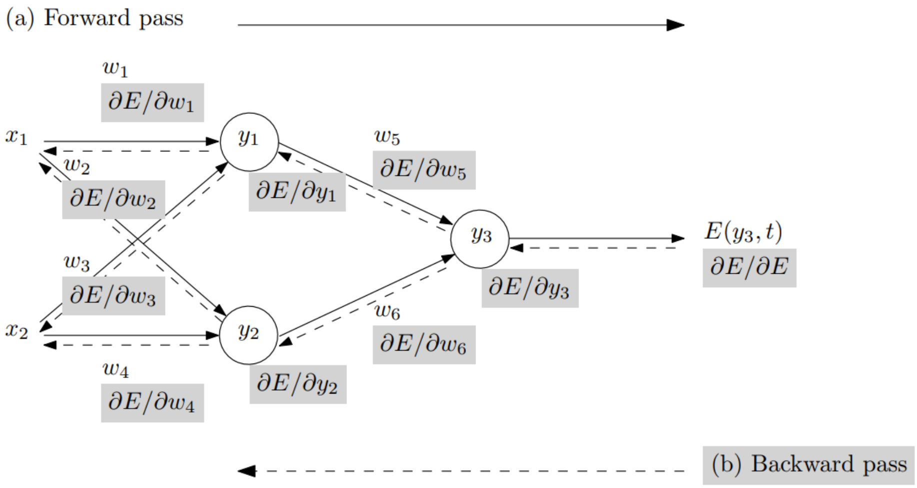
$$\mathcal{S}_\theta : \xi \mapsto u^\star(\cdot;\xi)$$

defines the parametric solution operator with trainable parameters θ.

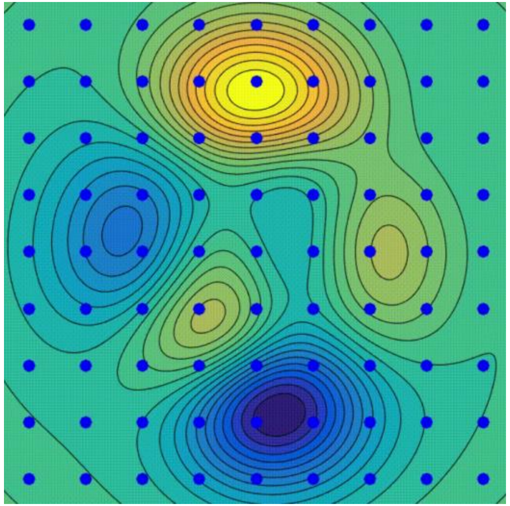
Automatic Differentiation (AD) in Machine Learning

AD enables efficient and accurate gradient computation, which is fundamental for training complex ML models using GPUs.

Backpropagation Algorithm



Gradient Descent Algorithm



Animation source: wikipedia

SW and HW Innovations

NVIDIA CUDA

Baydin, Atilim Gunes et al. Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research, 2015

Differentiable Programming Languages and Libraries

AD-supported Languages

- julia
- Swift
- Mojo

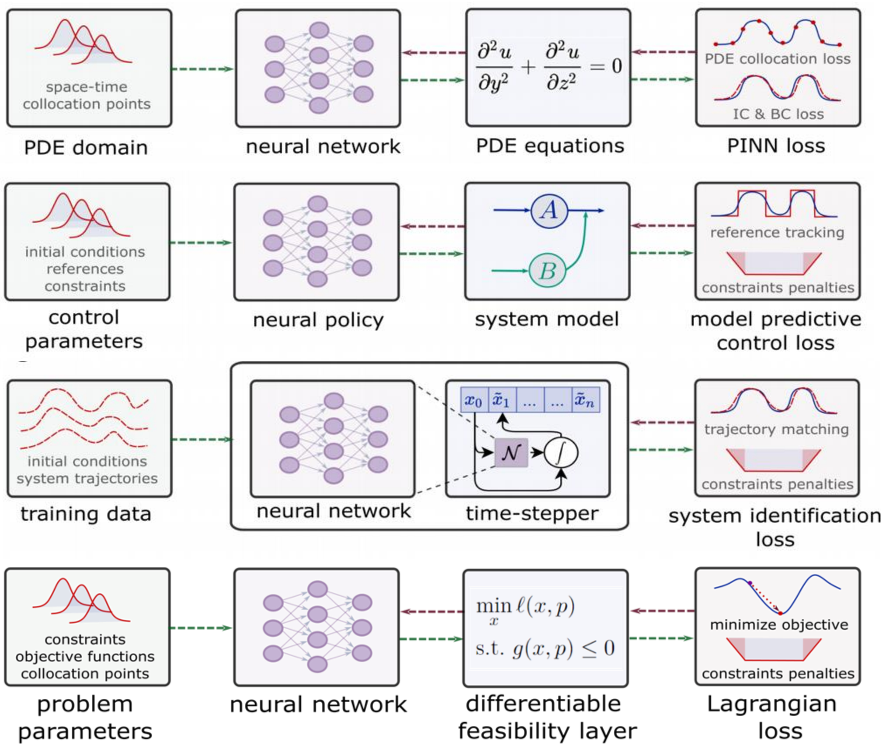
Domain-specific Languages and Libraries Utilizing AD

- CasADi
- Taichi Lang
- AMPL
- JuMP
- DifferentialEquations.jl

Machine Learning Libraries

- PyTorch
- TensorFlow
- flux

NeuroMANCER Scientific Machine Learning Library



Open-source library in PyTorch

- Physics-informed Neural Networks
- Learning to model (L2M)
- Learning to optimize (L2O)
- Learning to control (L2C)

github.com/pnnl/neuromancer

Scientific Machine Learning Abstraction

Paradigm	Unknown	Prior $\mathcal{N}$, $\mathcal{B}$	Objective $\mathcal{J}$	Solution map $\mathcal{S}_\theta(\xi)$
L2M	System model	Physics	Physics + data fit	Neural ODEs, PINNs
L2O	Decision map	Constraints	Optimality + feasibility	MLP, GNN, Differentiable Optimization Layers
L2C	Control policy	Dynamics, constraints	Closed-loop performance + safety	Differentiable MPC, Neural Networks

Where $\mathcal{P}_\theta = (\mathcal{N}, \mathcal{B}, \mathcal{J}, \mathcal{S}_\theta)$ is a differentiable program composed of physics $\mathcal{N}$, constraints $\mathcal{B}$, objectives $\mathcal{J}$, and solution operators $\mathcal{S}_\theta$.

Same abstraction. Different choices of $\mathcal{N}$, $\mathcal{B}$, $\mathcal{J}$, $\mathcal{S}_\theta$.