



Hybrid Modeling – Bridging Machine Learning with Domain Knowledge

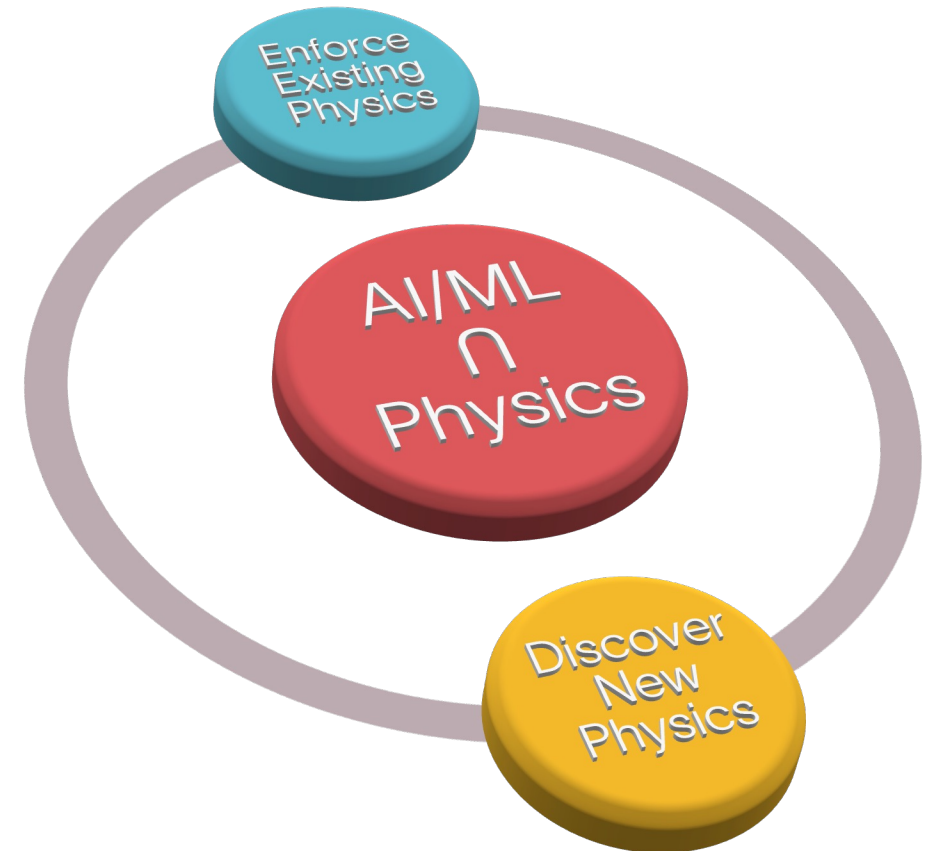
Indranil Roychoudhury
Principal AI Scientist, SLB

PHM Society Tutorial | November 2025

Physics-Informed Machine Learning: A Definition

“The paradigm of **physics-informed machine learning (PIML)** is to build a model that leverages **empirical data** and **available physical prior knowledge** to **improve performance** on a set of tasks that involve a **physical mechanism**”

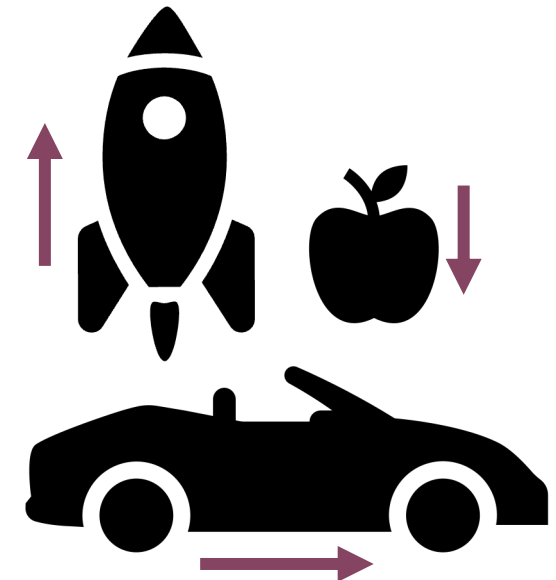
Ref: arxiv.org/2211.08064v1



Ref: <https://youtu.be/JoFW2uSd3Uo?si=-aIV5sVMfwyhw3Z>

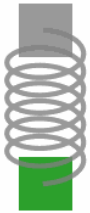
What is "Physics"?

- **Interpretability / Generalizability**
 - E.g., $Force = mass \times acceleration$
 - Extrapolation vs Interpolation
- **Parsimony / Simplicity**
 - "Everything should be made as simple as possible, but not simpler" - Einstein
- **Symmetrics / Invariances / Conservation**
 - Mass/Momentum conservation
 - Rotational/Translational symmetries
 - Invariance vs Equivariance



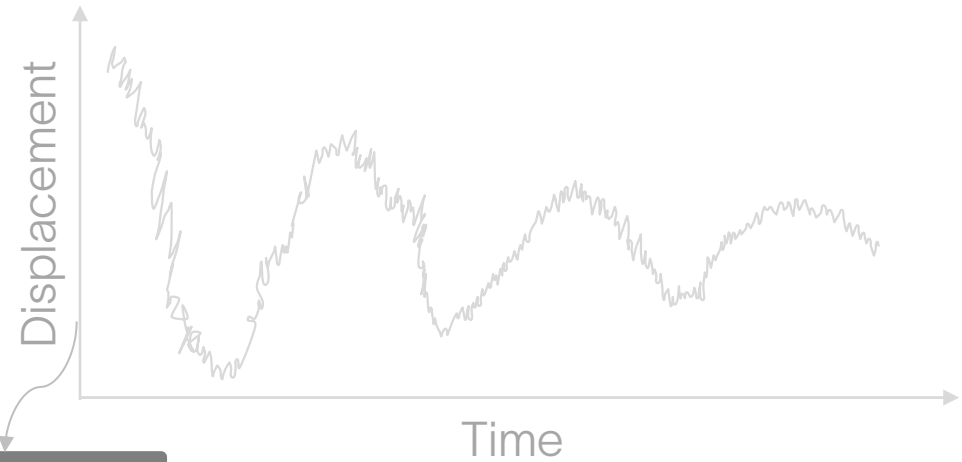
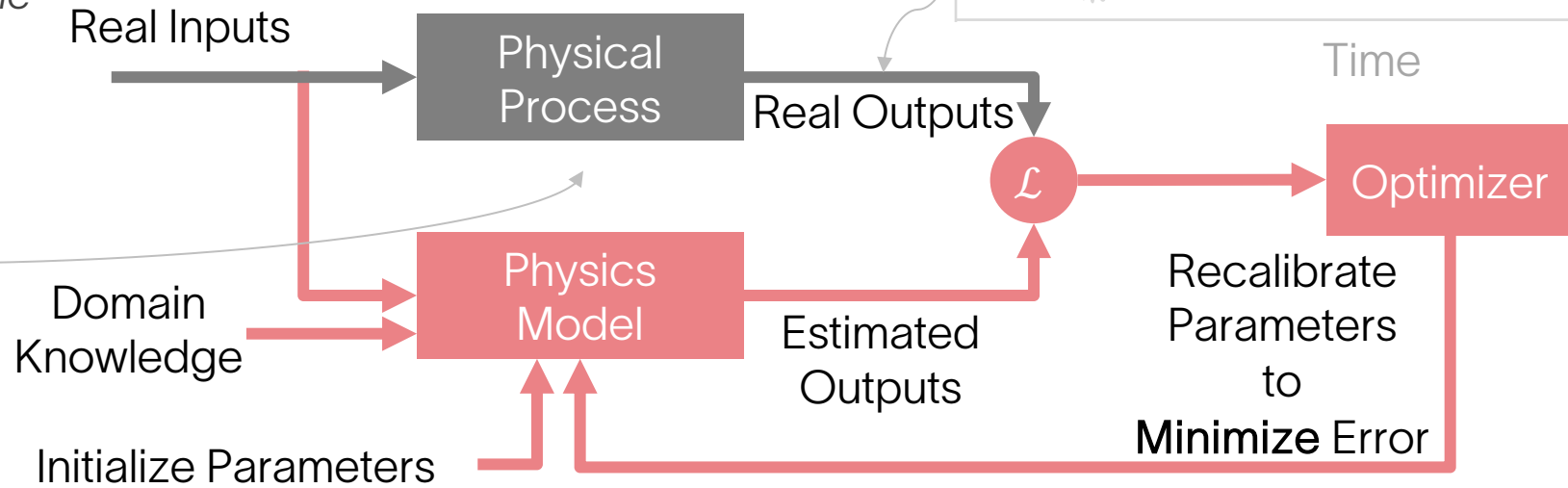
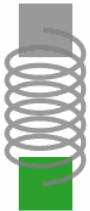
Modeling Dynamic Systems: An Example

*One-Dimensional
Damped Harmonic
Oscillator*



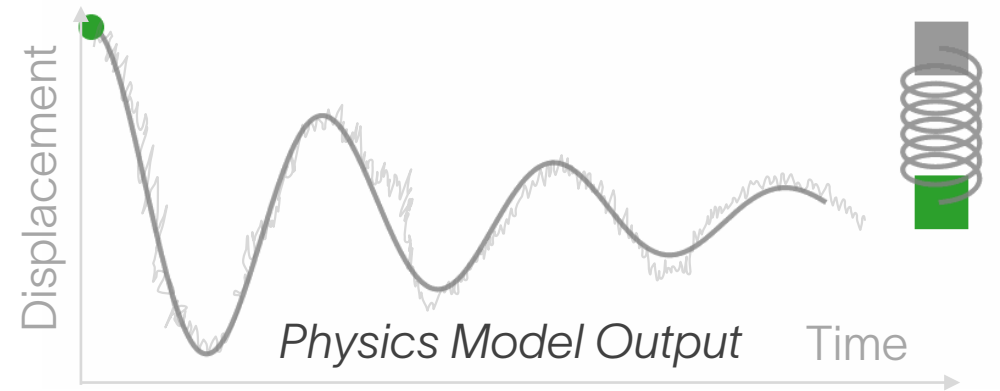
Building a Physics Model

One-Dimensional Damped Harmonic Oscillator



$$m \frac{d^2 u}{dx^2} + \mu \frac{du}{dx} + ku = 0$$

Physics Equation



What is Machine Learning?

- Machine Learning
 - The process of learning models from data via optimization
- Machine Learning (ML) is deeply embedded in modern life, influencing how we live, work, and interact.



Healthcare



Finance



Retail & E-commerce



Transportation



Communication



Manufacturing

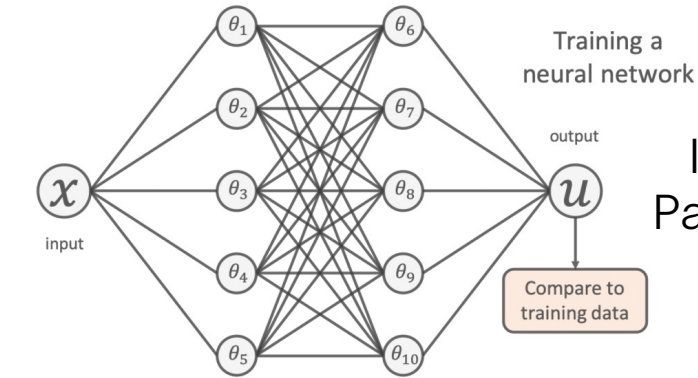


Science and Engineering

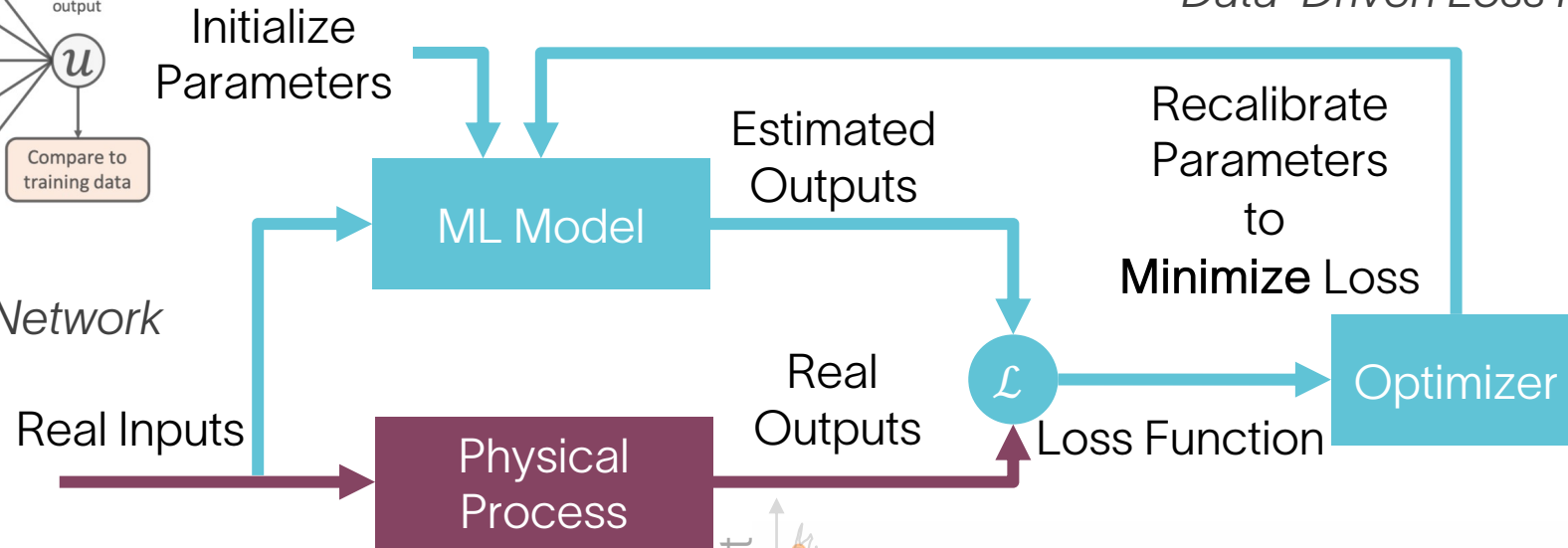
Building a Machine Learning Model

$$\min \frac{1}{N} \sum_i^N (u_{NN}(x_i; \theta) - u_{true}(x_i))^2$$

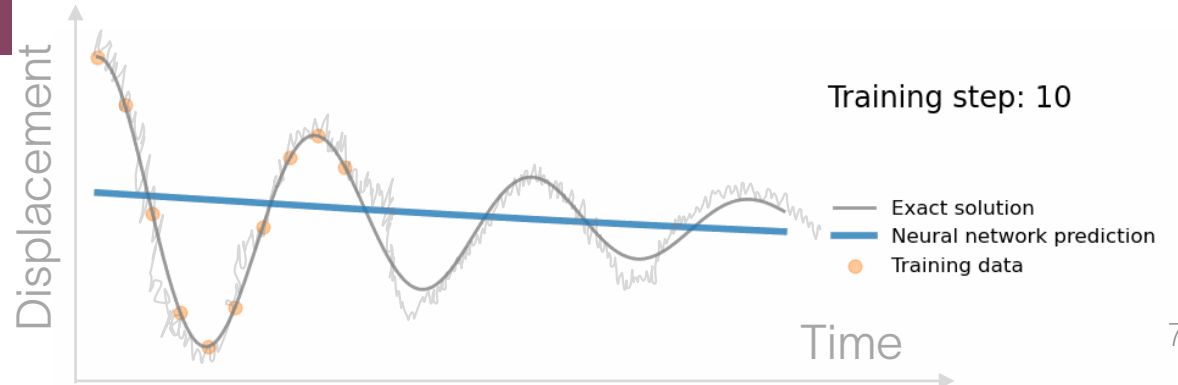
Data-Driven Loss Function



Schematic of a Neural Network



Fitting a Neural Network Model to Experimental Data



Advantages of Incorporating Physics into ML

- The goal of the embedding Physics to ML algorithm is to either get to a *better* model or learning *new* physics from data.
- Capture “expensive physics” through Surrogate modeling
 - Fast but accurate
 - Leverage automatic differentiation for optimization
- Capturing **multi-physics** interactions via Discrepancy modeling
- Advance the development of **virtual digital twins**, creating synergistic digital counterparts that closely mirror their real-world counterparts.

Advantages of Incorporating Physics into ML

- The goal of the embedding Physics to ML algorithm is to either get to a better model or learning *new physics* from data.
- Capture “expensive physics” through Surrogate modeling
 - Fast but accurate
 - Leverage automatic differentiation for optimization
- Capturing **multi-physics** interactions via Discrepancy modeling
- Advance the development of **virtual digital twins**, creating synergistic digital counterparts that closely mirror their real-world counterparts.

Making the ML model both physically and scientifically consistent.

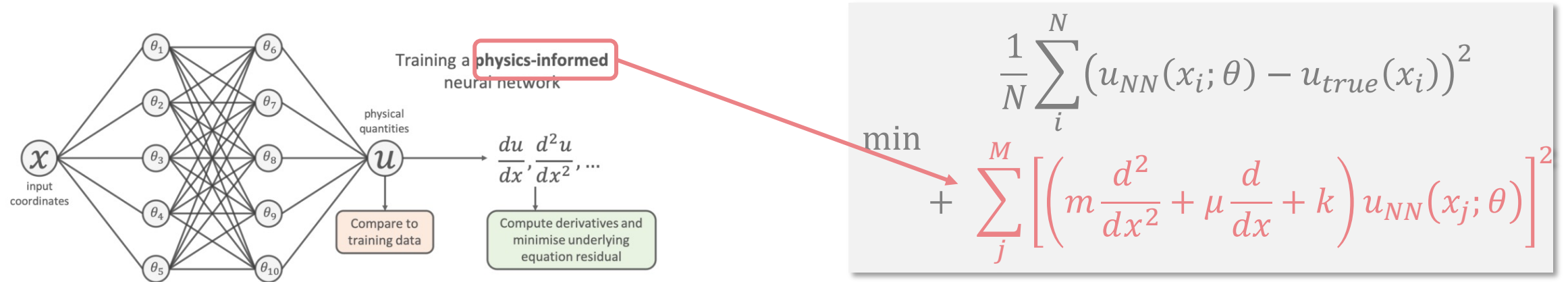
Model training becomes highly data-efficient, i.e. trainable with fewer data.

Acceleration of the model training process, such that the models converge faster to an optimal solution.

Generalization capabilities of the model, such that models can make better prediction for scenarios unseen during the training phase

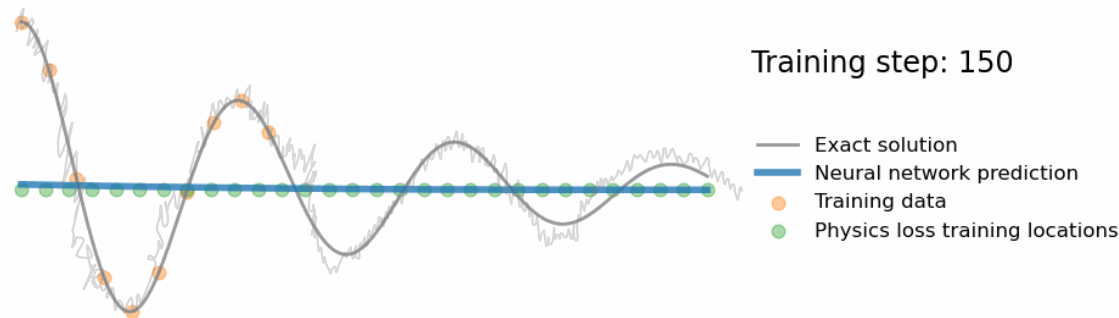
Improvement of transparency and interpretability of models.

Damped Harmonic Oscillator: Physics + ML Model



Schematic of a Physics-Informed Neural Network

Physics-Informed Loss Function



Fitting a Physics-Informed Neural Network Model to Experimental Data

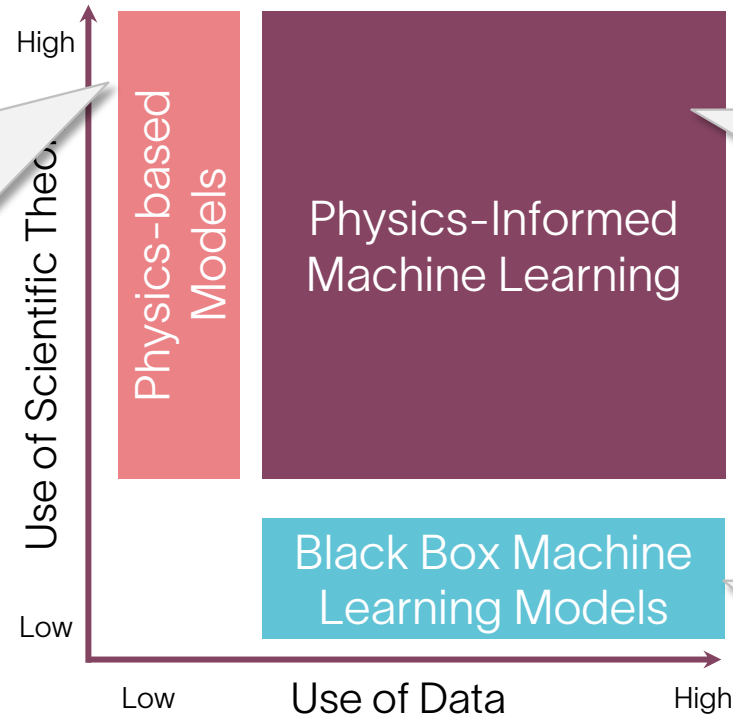
Ref: <https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/>

Benefits of `Physics AND ML' over `Physics OR ML'

- Approximations of reality
- Large number of parameters for more accuracy
- Data is typically limited
- Large simulators are computationally expensive

- Physics-Informed Machine Learning
- Physics-Guided Machine Learning
- Physics-Aware Machine Learning
- **Hybrid Modeling**
- Scientific Machine Learning
- ...

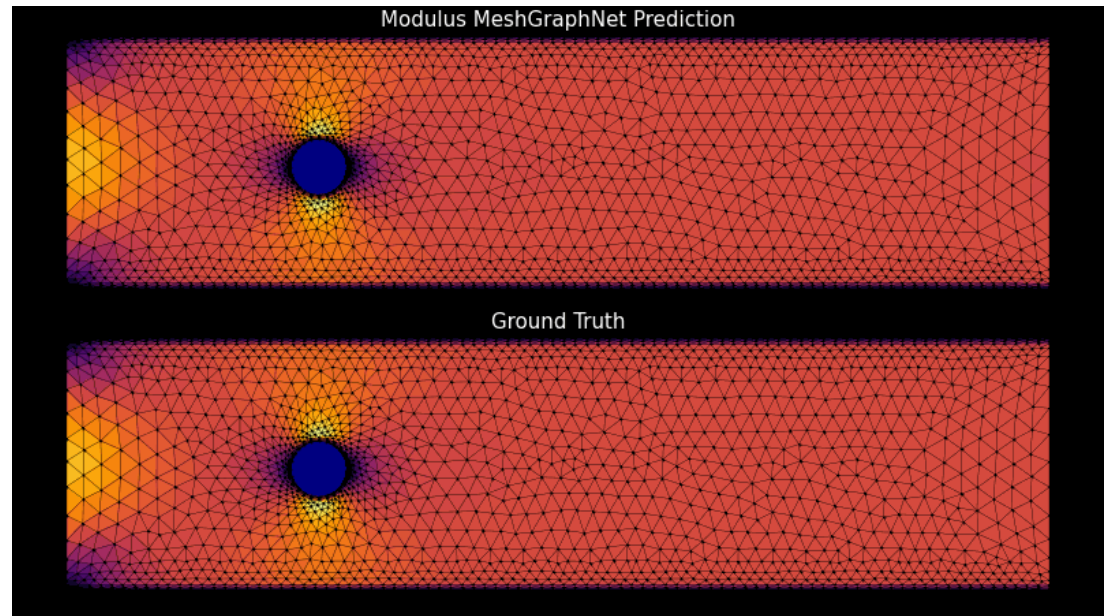
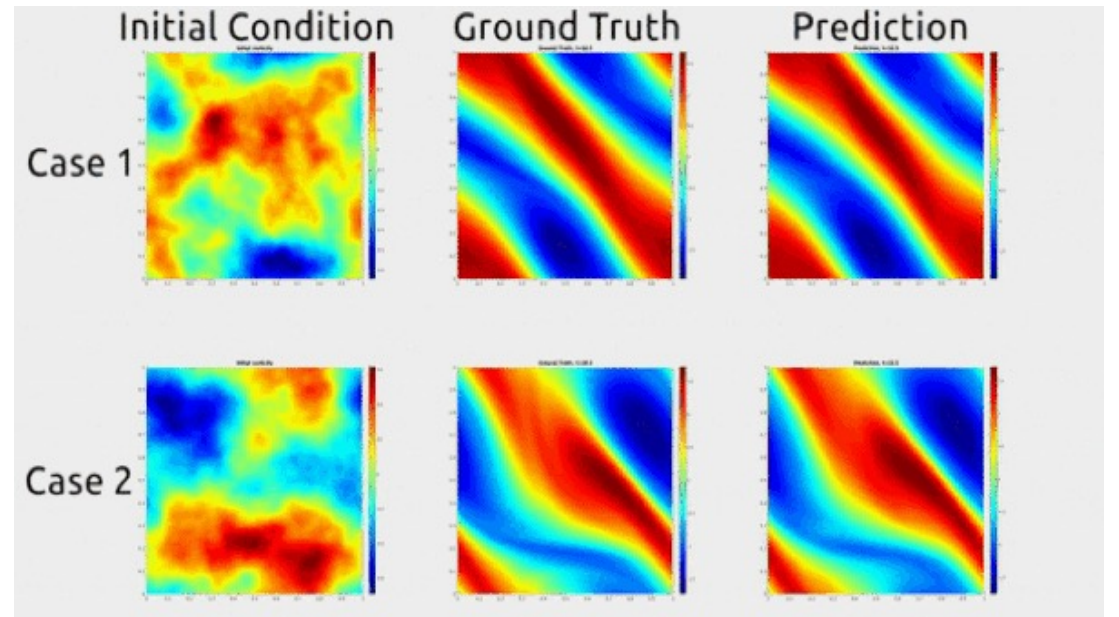
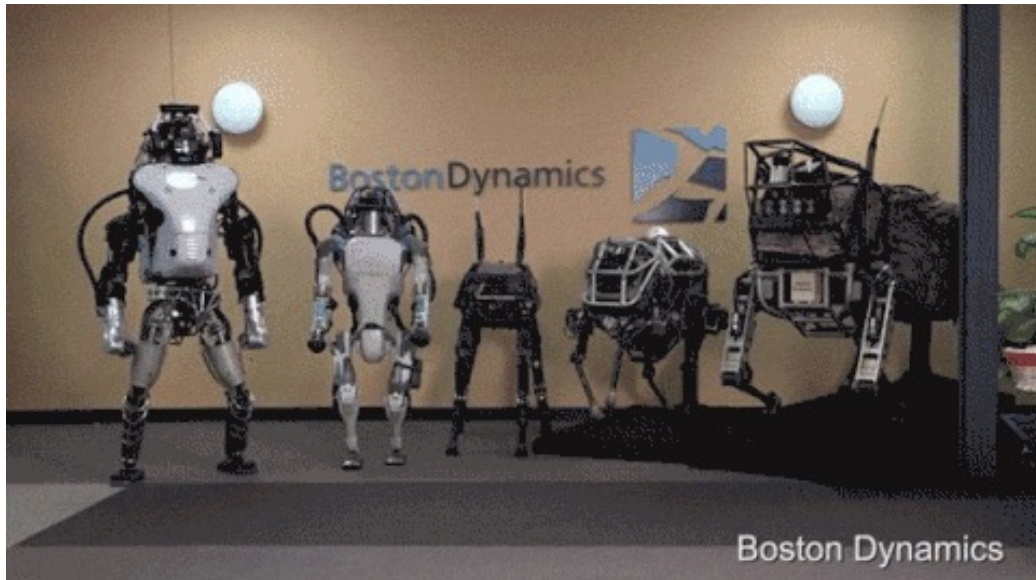
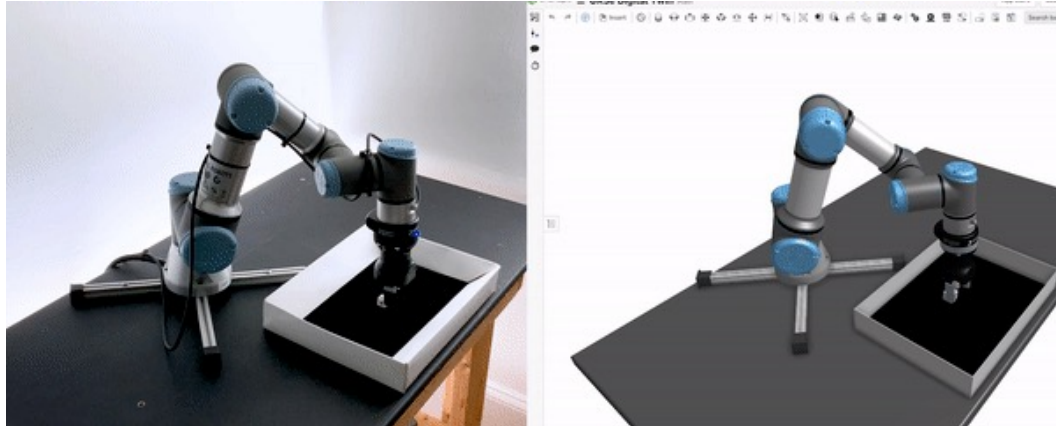
Synonyms for PIML



- Better capture system dynamics
- Scientifically consistent
- Generalizable
- Interpretable results
- Less Data Requirement

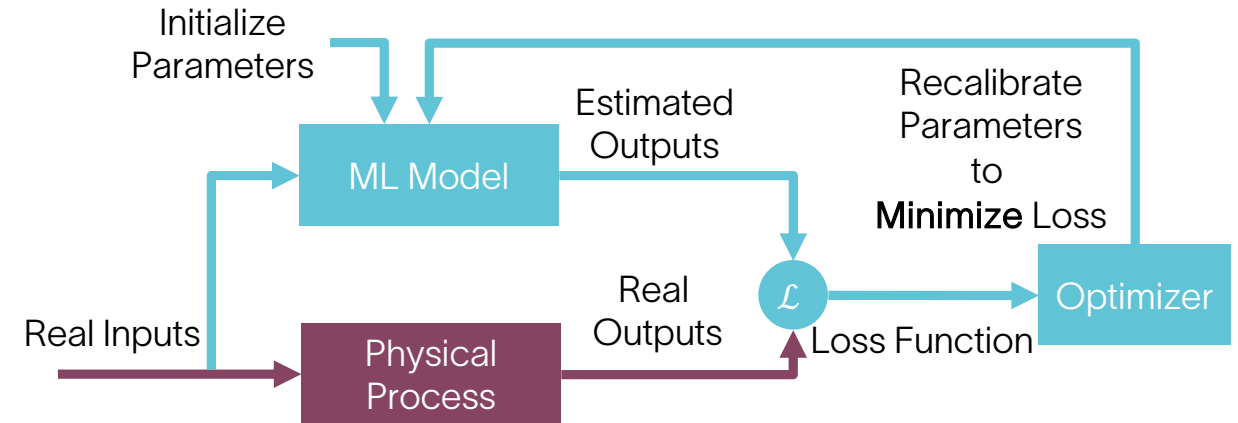
- Very fast
- Only as good as the data
- Large data requirement
- Scientifically inconsistent results are possible

Some Applications of PIML

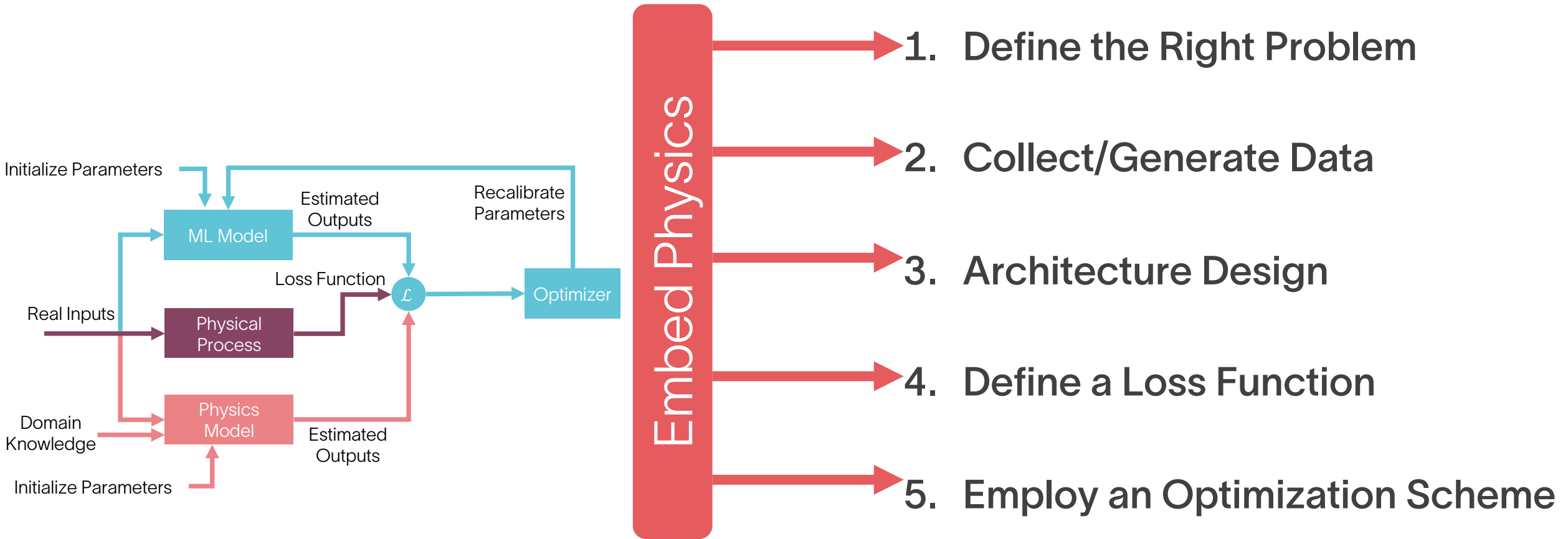


Five Stages of Machine Learning

1. Define the Right Problem
2. Collect/Generate Data
3. Architecture Design
4. Define a Loss Function
5. Employ an Optimization Scheme

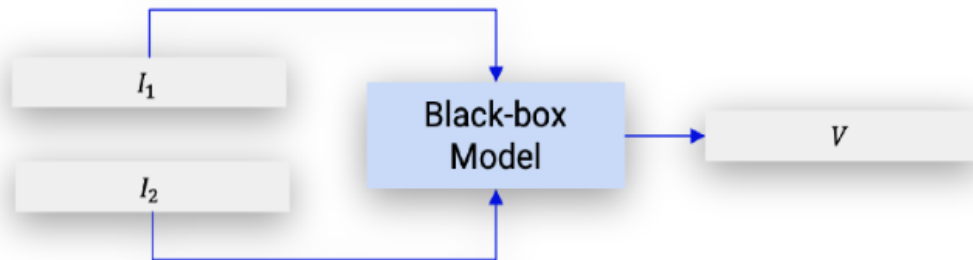


Five Stages of **Physics-Informed** Machine Learning



Amount of Physics Knowledge Also Influences PIML Design

(1) Black-Box Models



(3) Functional Models

→ e.g., $V = I_1 \times R_1 + I_2 \times R_2$, where R_1 and R_2 are **not** instantiated

(2) Causal Directed Acyclic Graphs



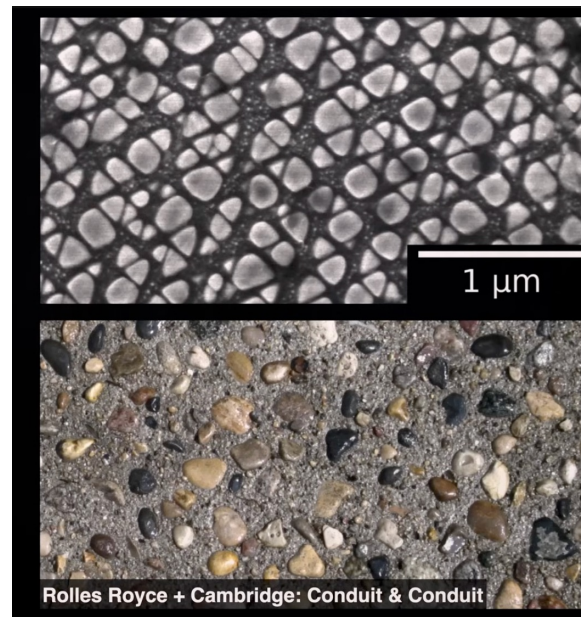
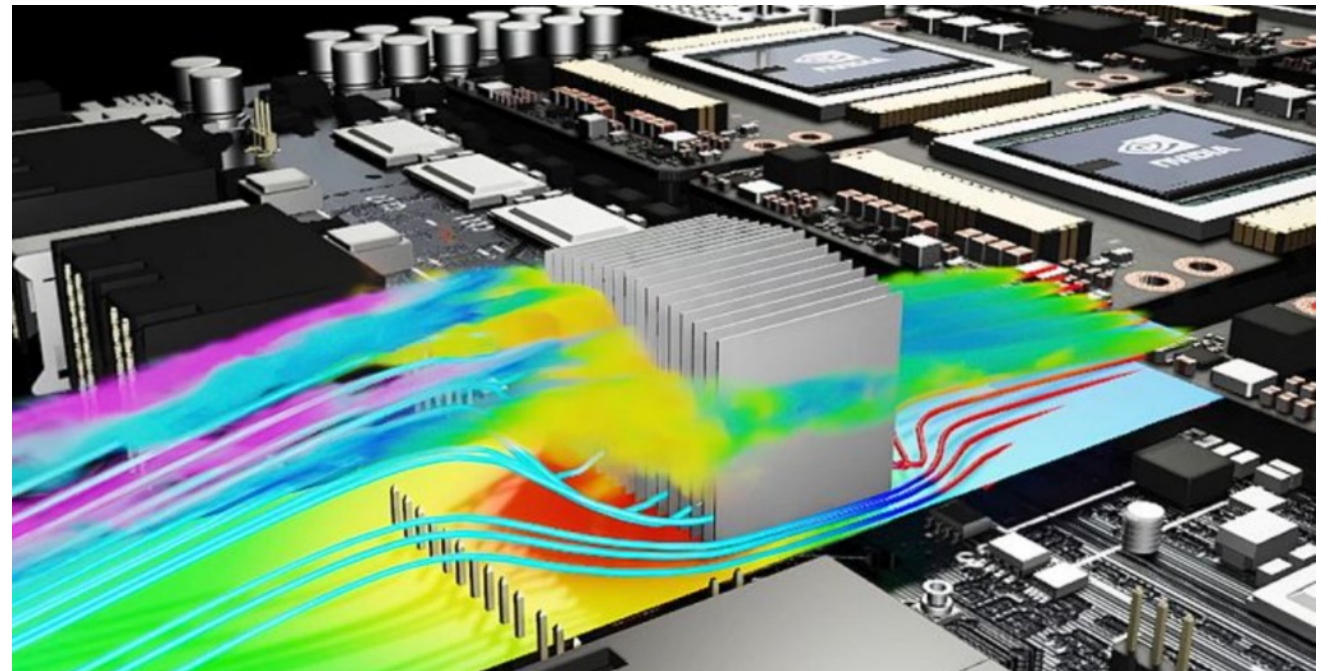
(4) Realized Models



- e.g., $V = I_1 \times R_1 + I_2 \times R_2$, where $R_1 = 5m\Omega$, and $R_2 = 15m\Omega$, i.e., parameters **are** instantiated
- Based on accuracy of parameters, a realized model can be **high-fidelity** or **low-fidelity**

Define the Right Problem

Decide on the Problem

- Modeling fluid flows and turbulence
- Materials Discovery
- Computational Bio-chemistry
- Drug Discovery/Protein Design
- Shape Optimization
- Chaotic systems - might want probability



<p>astronomy</p>  <p>The science of everything outside of Earth's atmosphere.</p>	<p>astrology</p>  <p>The practice of using the positions of celestial objects to interpret human behavior.</p>
--	---

Curate the Right Data

Data Curation for PIML

- Embedding physics **reduces data requirements**, ideal for costly experiments.
- **Data Augmentation**
 - Use symmetries (e.g., rotation invariance) to expand datasets efficiently.
- **Coordinate System Selection**
 - Choosing optimal coordinates simplifies learning tasks.
- **Simulation vs. Experiment Trade-offs**
 - Simulations: High spatial resolution, expensive for long runs.
 - Experiments: Real-world fidelity, long time series, lower spatial resolution.

Static Data

- Simulated or experiment

Interactive Data

- Simulated only

Living & Physical

- Cyberphysical experiment

Heliocentrism



Geocentrism



<https://sciencegifs.csullender.com/2016/07/25/Heliocentrism-vs-Geocentrism.html>

Data Curation for PIML

- **Generalization for Design Optimization:**
Physics-based constraints help models extrapolate to unseen scenarios.
- **Sparse & Expensive Data**
Requires clever architectures to extract maximum value.
- **Bias & Small Signals**
Watch for training bias (e.g., only successful cases) and subtle physical effects.
- **Hidden Variables**
ML can infer unmeasured variables using partial observations.
- **Digital Twins & Active Learning**
Combine multi-fidelity data sources; models query for high-uncertainty regions

Effective data curation is iterative and intuition-driven, blending physics, ML, and human expertise

Architecture Design

Constraining Architectures with Causal DAGs

- A standard fully-connected layer:

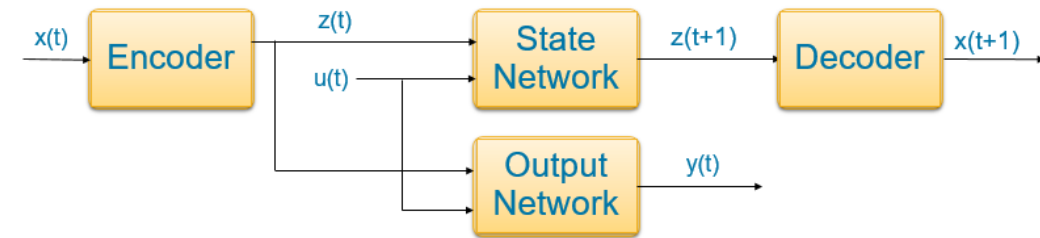
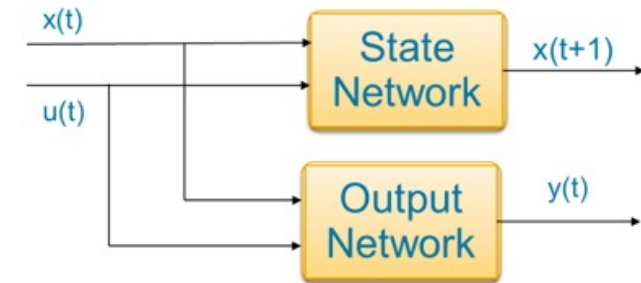
$$\text{output} = W * \text{input} + b$$

- With a causal mask M derived from the DAG:

$$\text{output} = (W \odot M) * \text{input} + b,$$

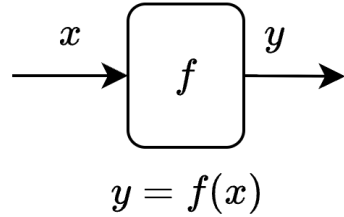
where \odot is element-wise multiplication.

- The mask M is not learned; it is a fixed structural prior imposed on the network.
- This prevents the model from learning spurious correlations and improves its ability to generalize to interventions.

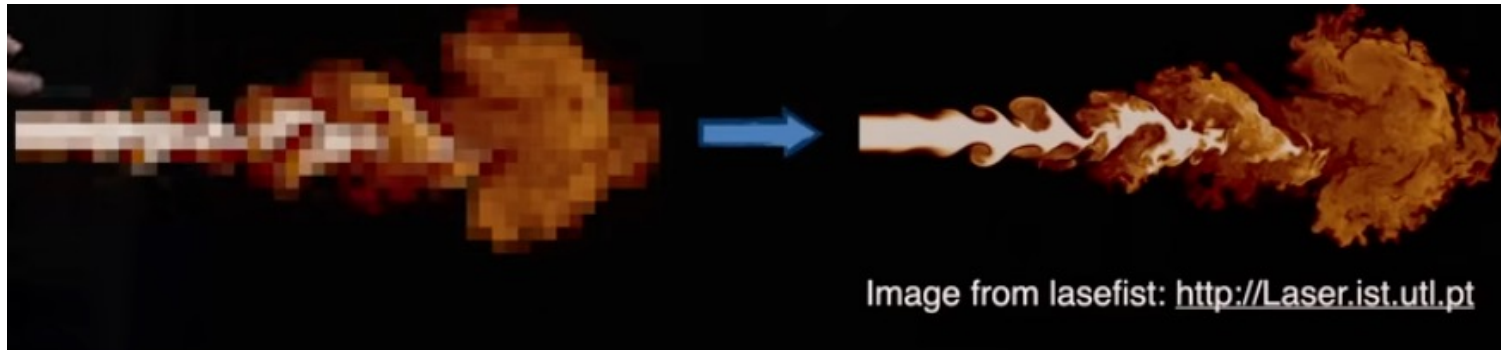
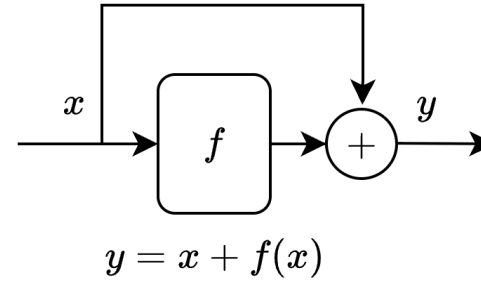


<https://www.mathworks.com/help/ident/ug/what-are-neural-state-space-models.html>

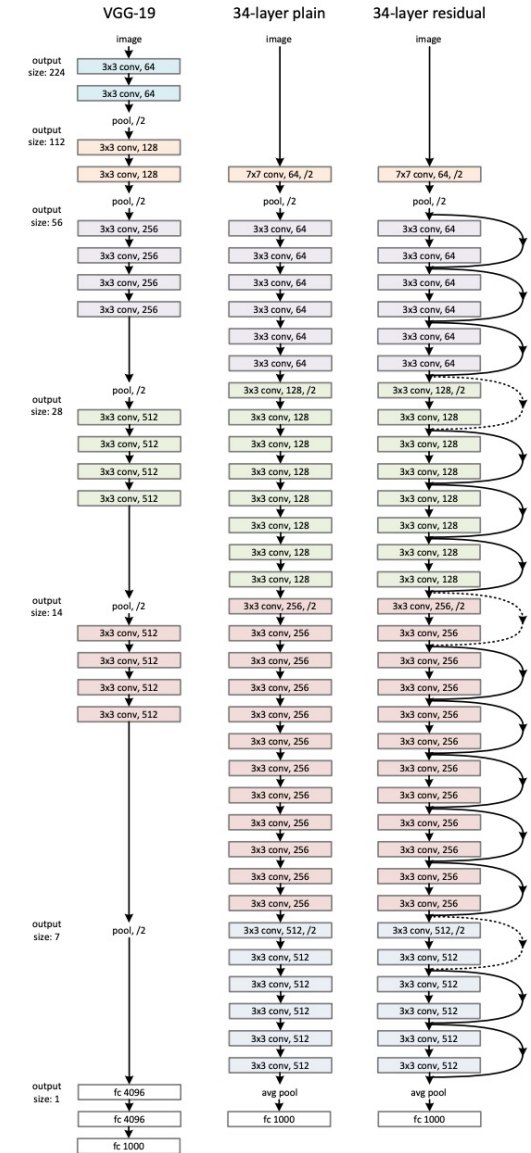
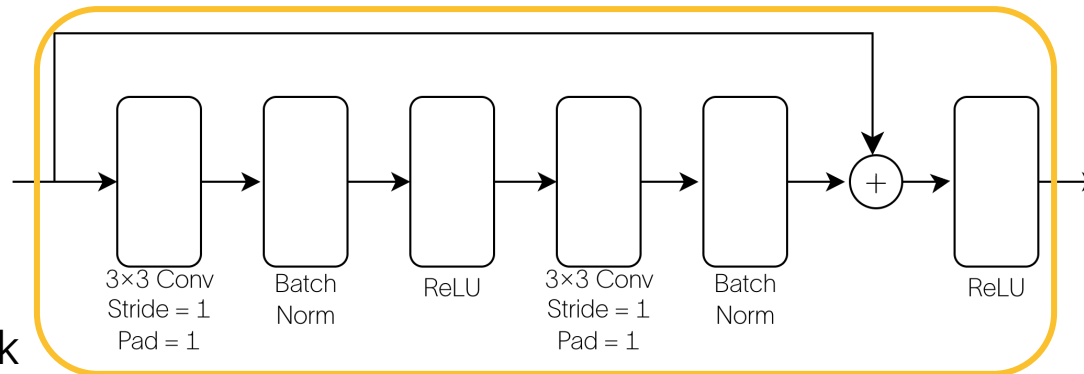
Residual Network (ResNet)



VS



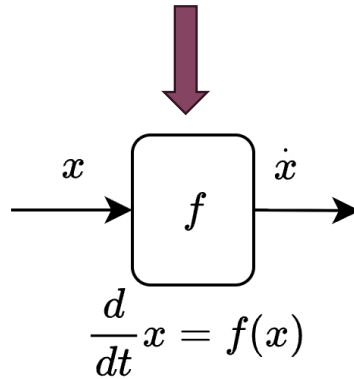
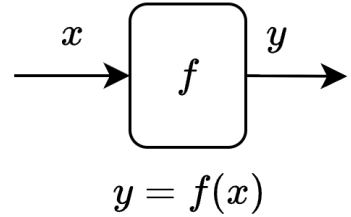
ResNet Block



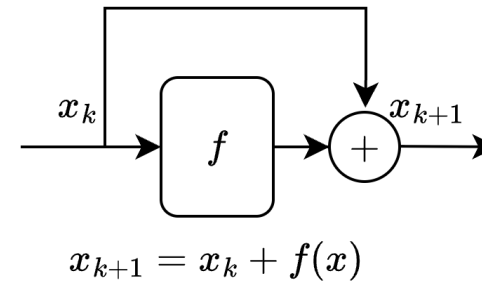
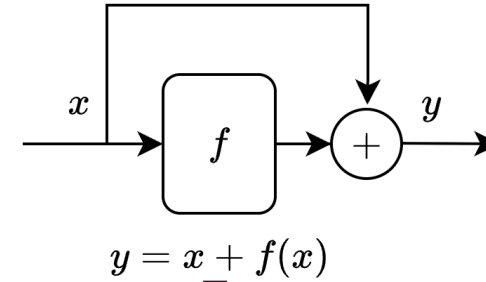
Ref: <https://arxiv.org/pdf/1512.03385>

Residual Network (ResNet) for ODEs

ODES

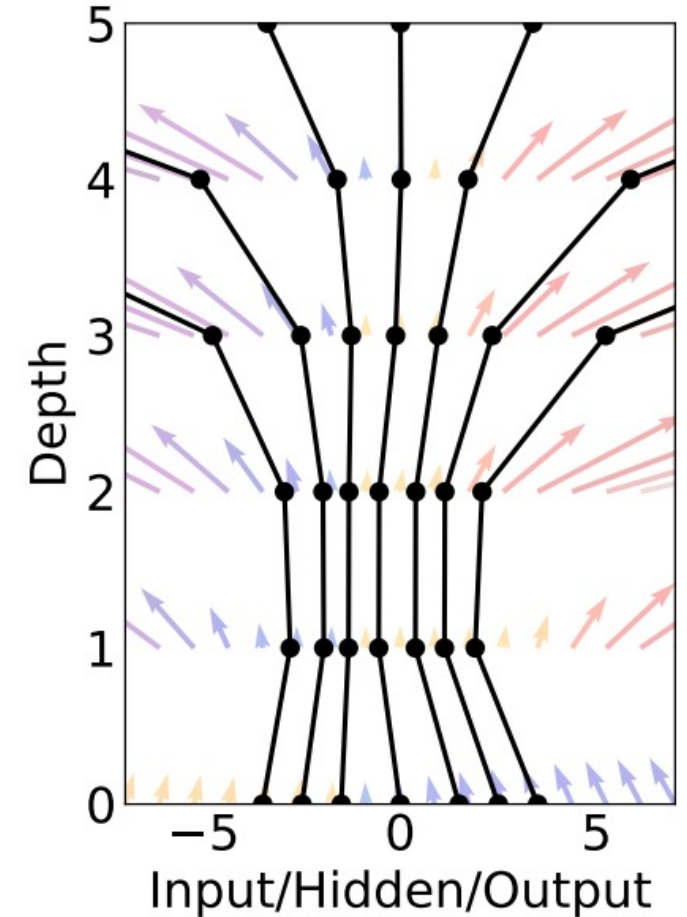


vs

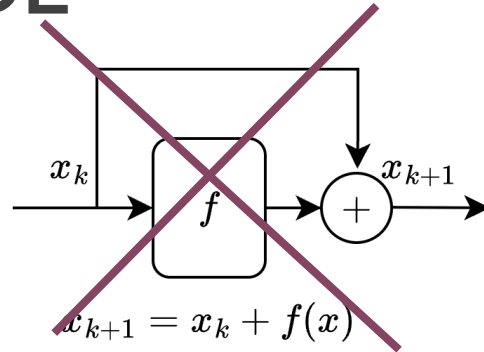
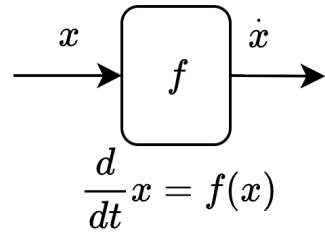


Euler Integrator!

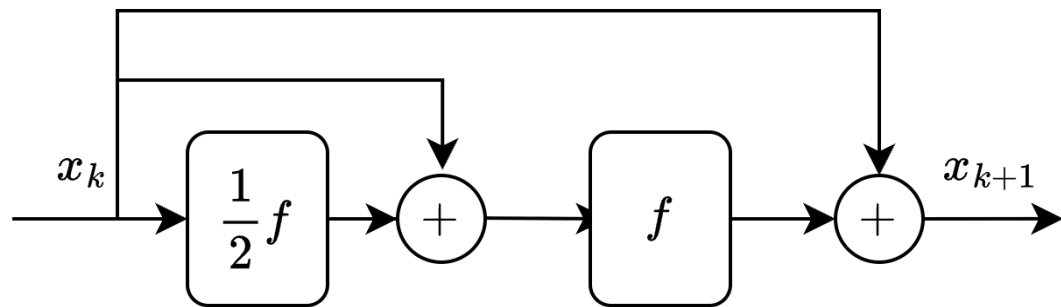
Residual Network



Neural ODE

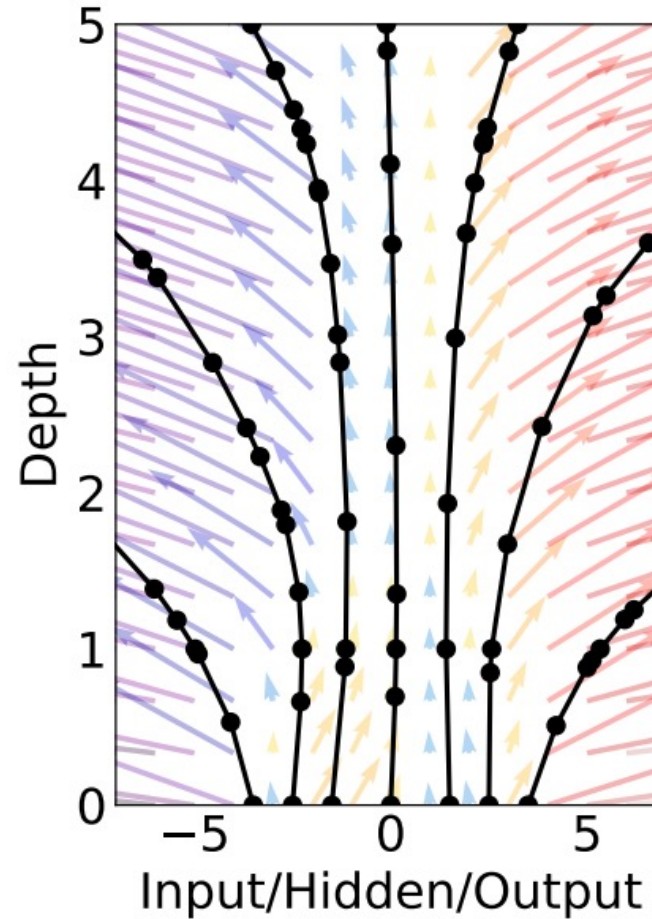


$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f(x(\tau))d\tau$$

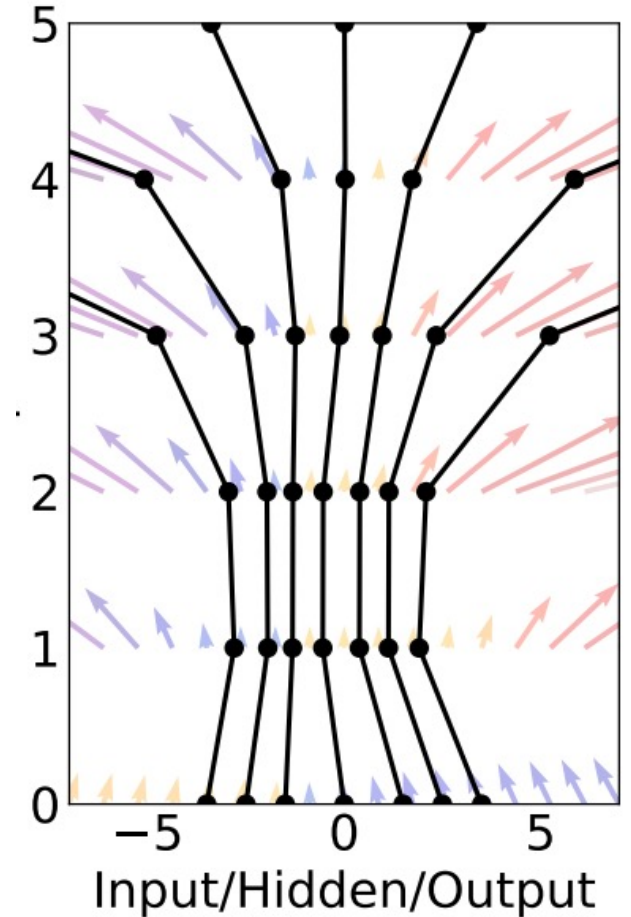


$$x_{k+1} = x_k + f\left(x_k + \frac{1}{2}f(x_k)\right)$$

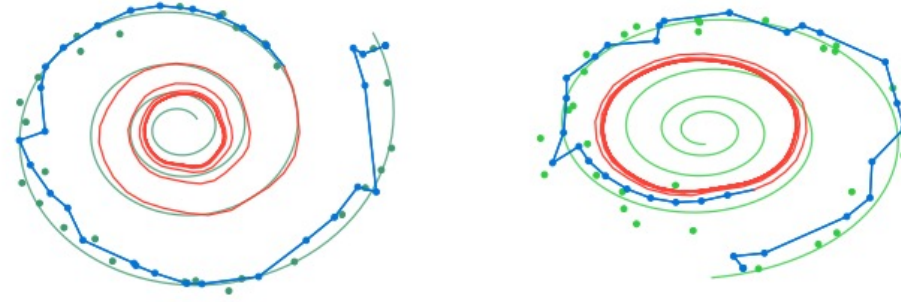
ODE Network



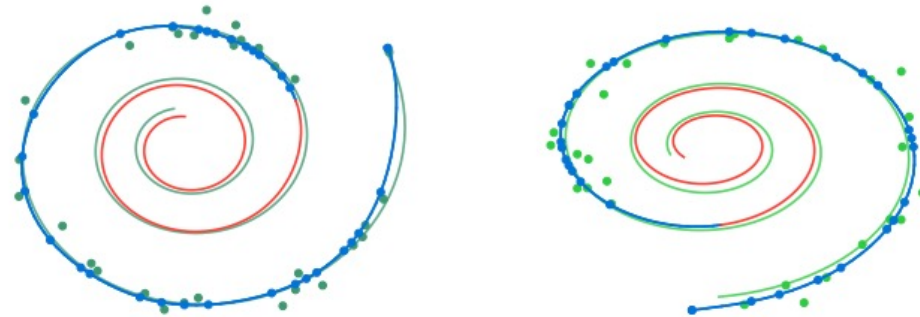
Residual Network



Neural ODE

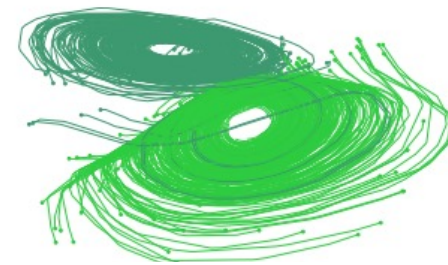


(a) Recurrent Neural Network



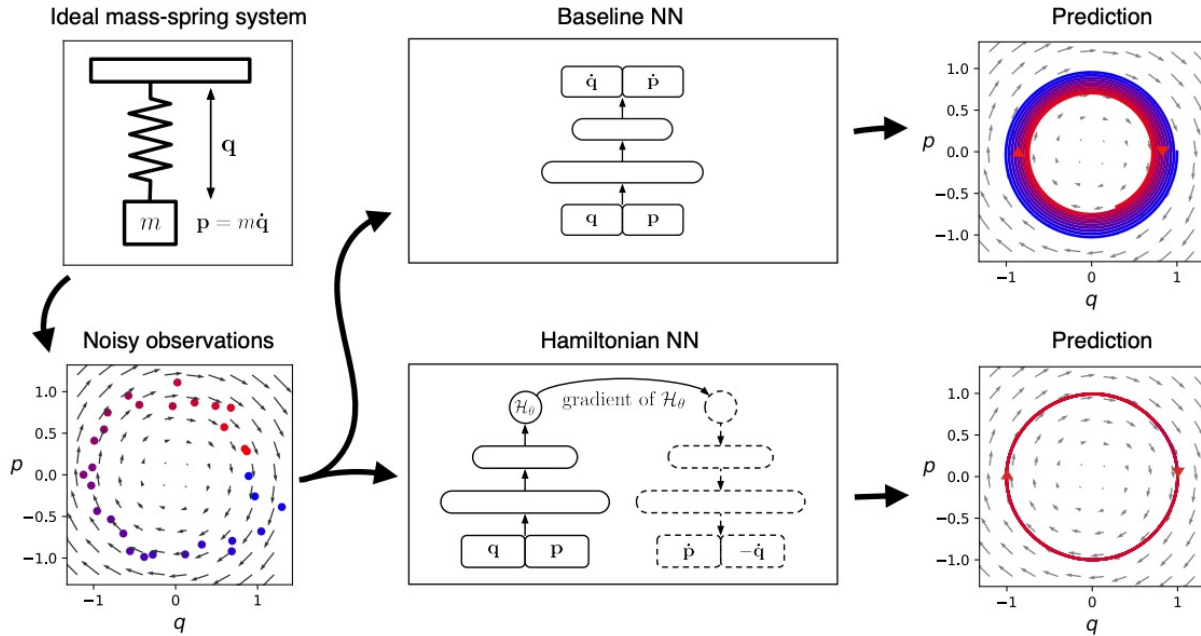
(b) Latent Neural Ordinary Differential Equation

- Ground Truth
- Observation
- Prediction
- Extrapolation



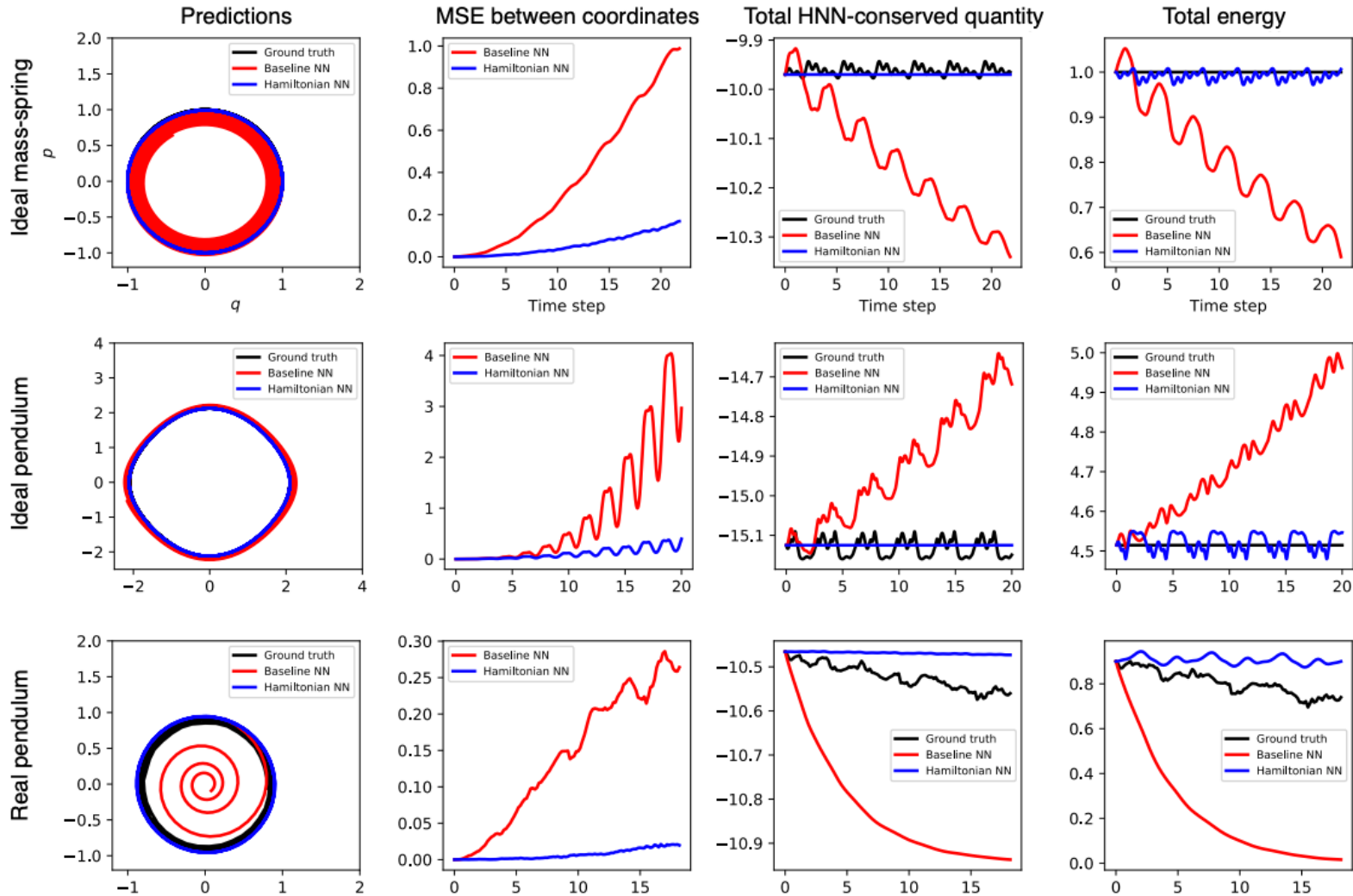
(c) Latent Trajectories

Hamiltonian Neural Networks



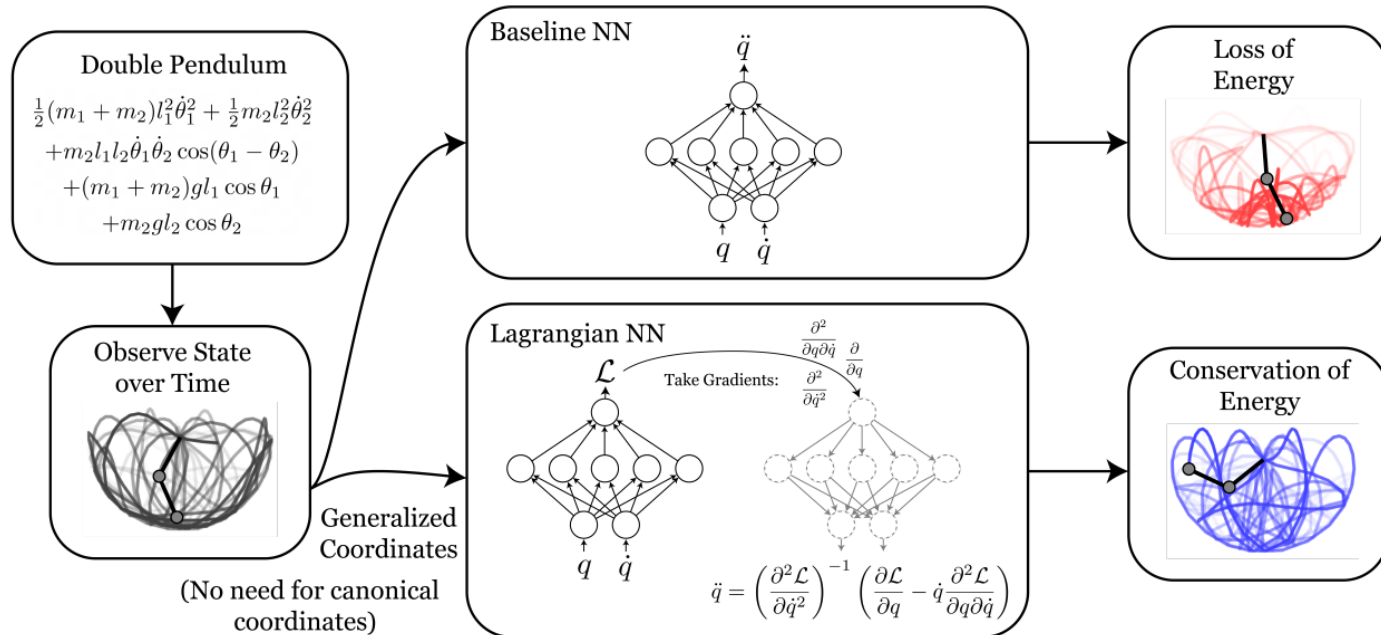
- Learn \mathcal{H}
- $\frac{dq}{dt} = \frac{\partial \mathcal{H}}{\partial p} \frac{dp}{dt} = -\frac{\partial \mathcal{H}}{\partial q}$
- $\mathcal{L} = \left\| \frac{\partial \mathcal{H}}{\partial p} - \frac{dq}{dt} \right\|_2 + \left\| \frac{\partial \mathcal{H}}{\partial q} + \frac{dp}{dt} \right\|_2$
- Essentially a Neural ODE with additional structure
- Uses structure and loss function to induce physics
- Enforces energy conservation

Hamiltonian Neural Networks



<https://arxiv.org/pdf/1906.01563>

Lagrangian Neural Networks



$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_j} = \frac{\partial \mathcal{L}}{\partial q_j}$$

$$\frac{d}{dt} \nabla_{\dot{q}} \mathcal{L} = \nabla_q \mathcal{L}$$

$$(\nabla_{\dot{q}} \nabla_{\dot{q}}^T \mathcal{L}) \ddot{q} + (\nabla_q \nabla_{\dot{q}}^T \mathcal{L}) \dot{q} = \nabla_q \mathcal{L}$$

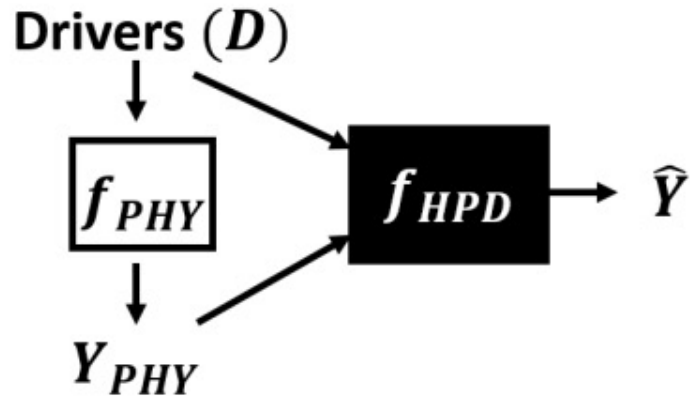
$$\ddot{q} = (\nabla_{\dot{q}} \nabla_{\dot{q}}^T \mathcal{L})^{-1} [\nabla_q \mathcal{L} - (\nabla_q \nabla_{\dot{q}}^T \mathcal{L}) \dot{q}]$$

- \ddot{q} is learned using AD like JAX
- Essentially a Neural ODE with additional structure
- Uses structure and loss function to induce physics
- Enforces a variational integrator to solve the Euler-Lagrangian Equations

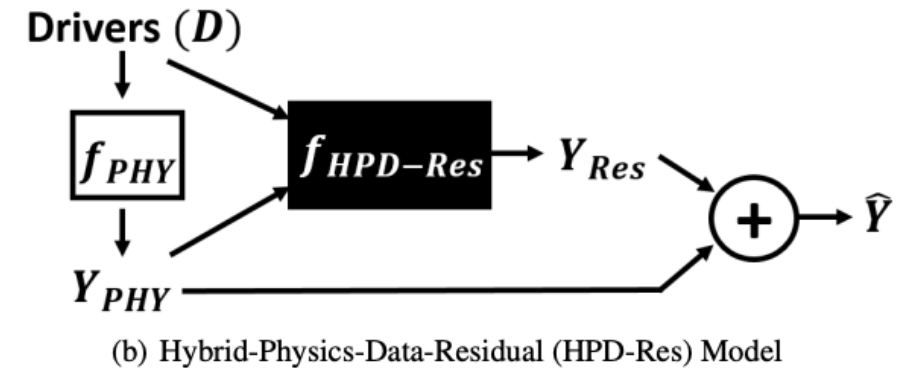
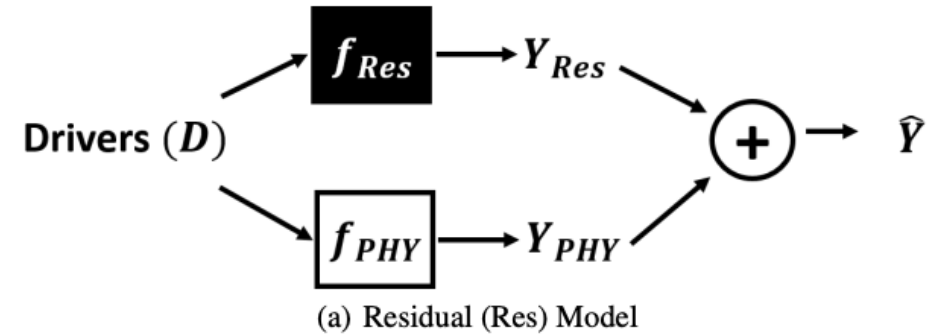
	Neural net	Neural ODE	HNN	LNN (ours)
Can model dynamical systems	✓	✓	✓	✓
Learns differential equations		✓	✓	✓
Learns exact conservation laws			✓	✓
Learns from arbitrary coords.	✓	✓		✓
Learns arbitrary Lagrangians				✓

<https://arxiv.org/pdf/2003.04630>

Physics-Guided Neural Networks

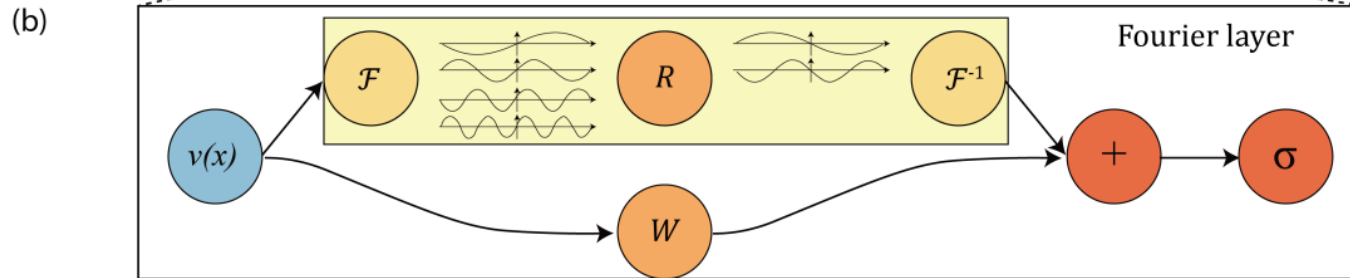
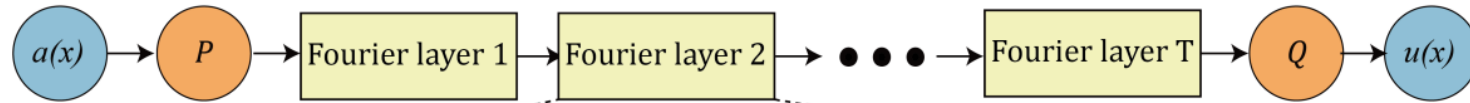


$$\arg \min_f \underbrace{Loss(\hat{Y}, Y)}_{\text{Empirical Error}} + \underbrace{\lambda R(f)}_{\text{Structural Error}} + \underbrace{\lambda_{PHY} Loss.PHY(\hat{Y})}_{\text{Physical Inconsistency}},$$

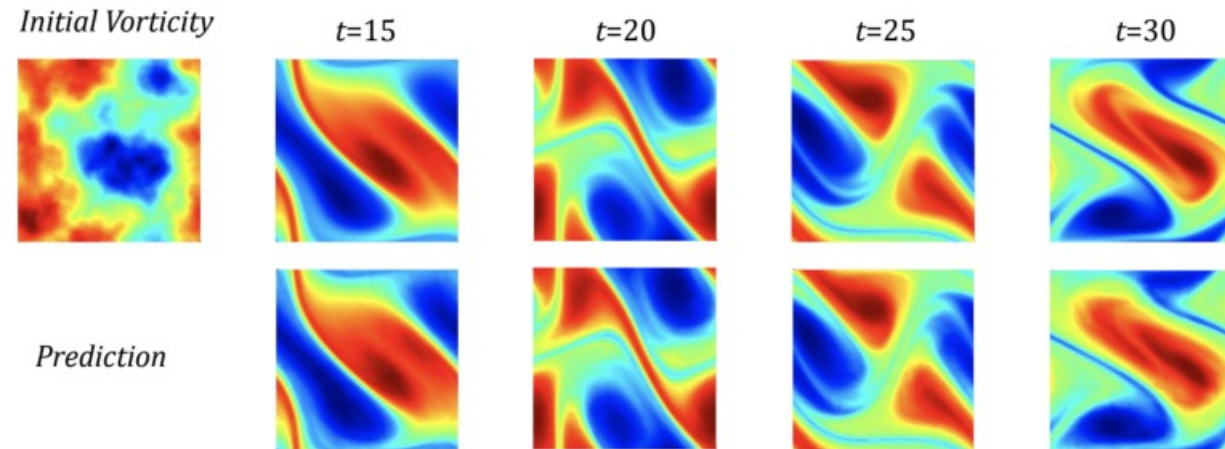


Other Variants of PGNN

Fourier Neural Operator

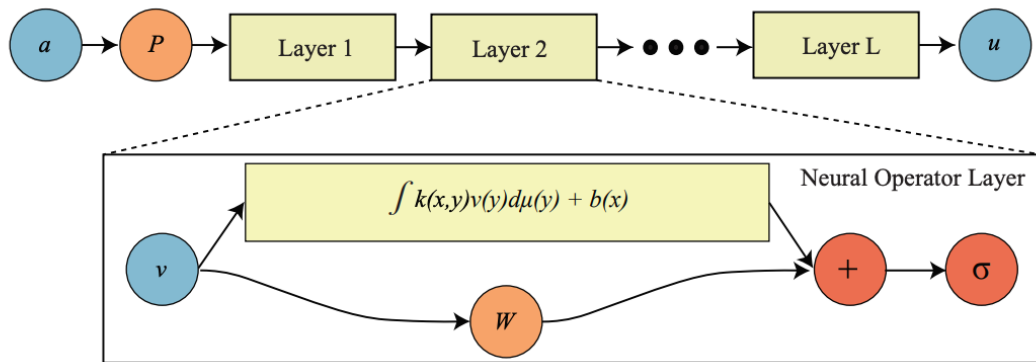


Operator learning can be taken as an image-to-image problem. The Fourier layer can be viewed as a substitute for the convolution layer.

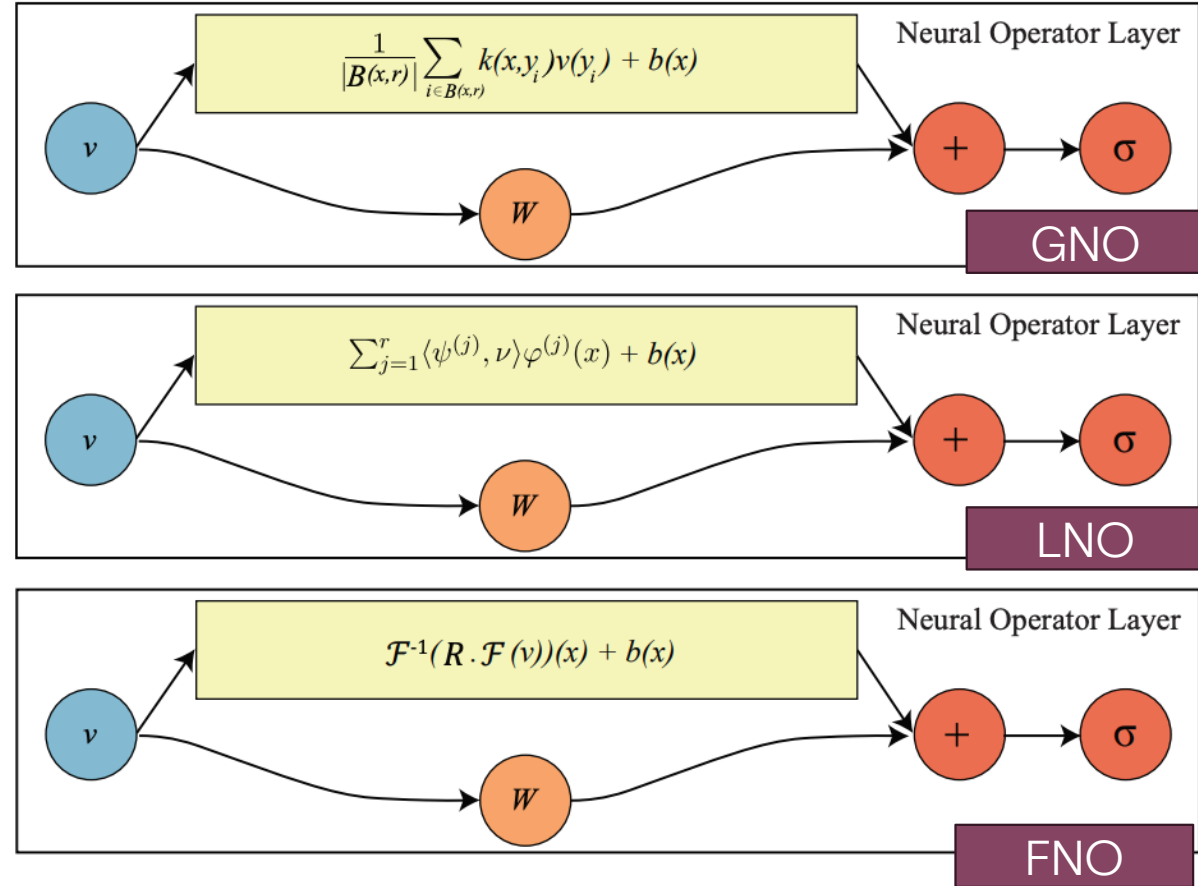


Zero-shot super-resolution: Navier-Stokes Equation with viscosity $\nu = 1e-4$; Ground truth on top and prediction on bottom; trained on $64 \times 64 \times 20$ dataset; evaluated on $256 \times 256 \times 80$ (see Section 5.4).

Operator Methods

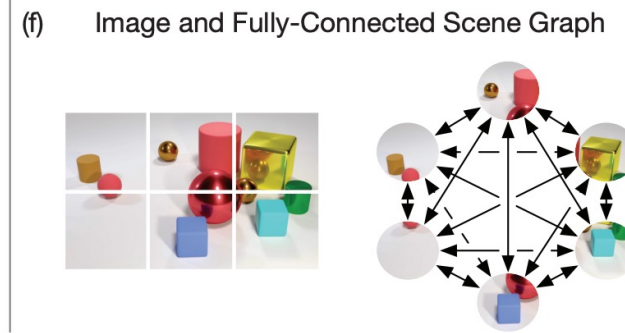
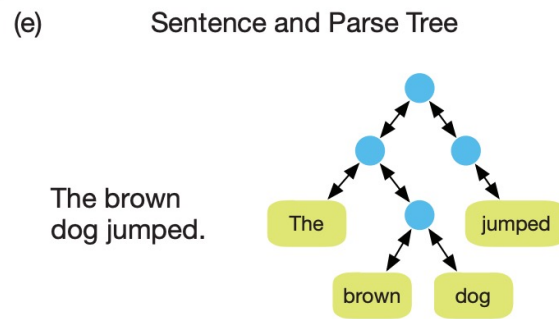
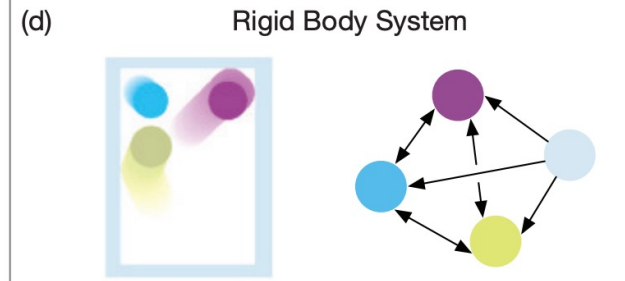
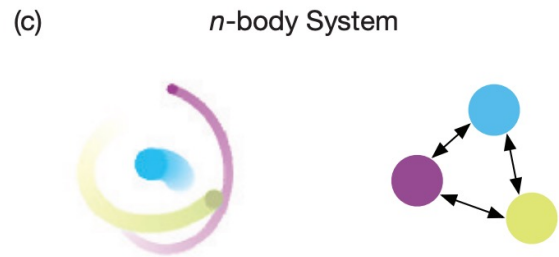
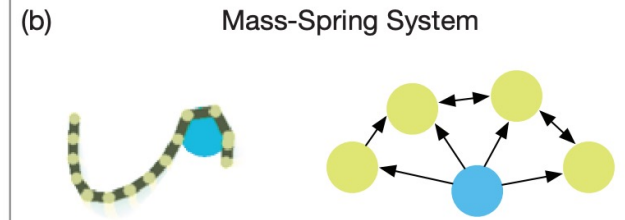
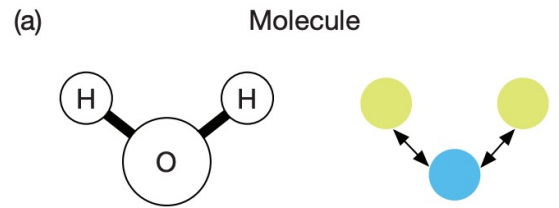


General Neural Operator Architectures

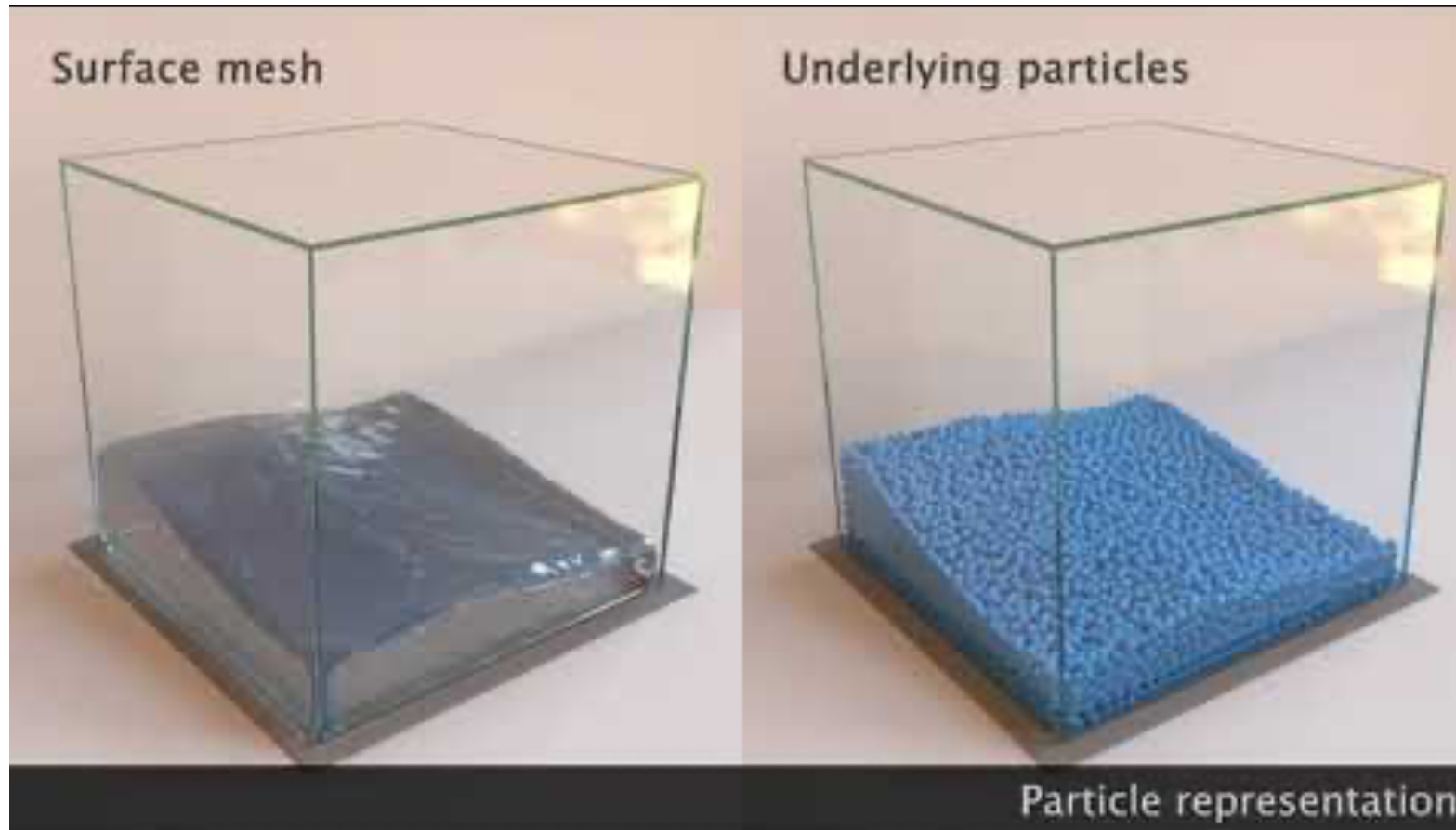
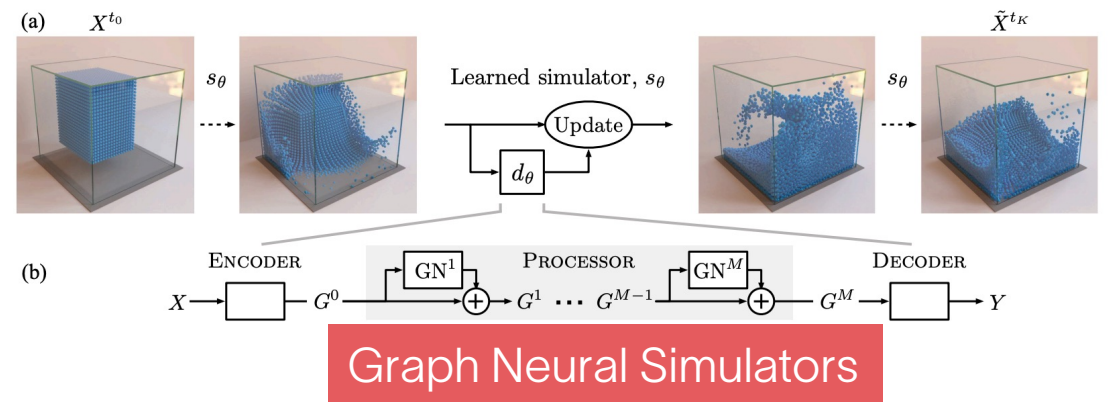


Instantiations of Neural Operator Architectures

Graph Neural Networks



Graph Neural Simulators





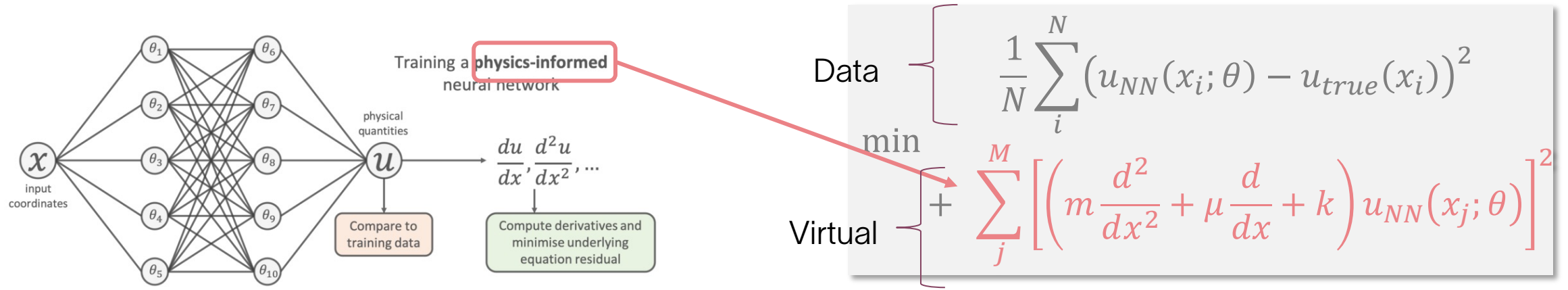
Define a Loss Function

Loss functions **encourage** physics

Steven L Brunton

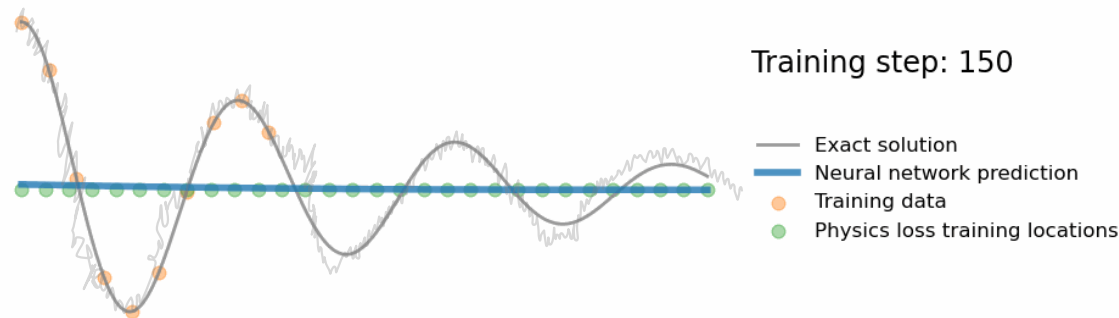
36

Physics Informed Neural Networks



Schematic of a Physics-Informed Neural Network

Physics-Informed Loss Function



Fitting a Physics-Informed Neural Network Model to Experimental Data

Ref: <https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/>



Employ an Optimization Scheme

Loss functions **encourage** physics

Optimization algorithms **enforce** physics

Steven L Brunton

Energy Conserving Constraints

Loss Function

$$\min_{\xi, z} \|\hat{\Theta}(\mathbf{X})\mathbf{E} - \dot{\mathbf{X}}\|_2^2 + z^T(\mathbf{C}\xi - d)$$

Model error

Constraints

Energy Conservation Constraints

$$\xi_8^{(a_1)} + \xi_6^{(a_2)} + \xi_5^{(a_3)} = 0,$$

$$\xi_4^{(a_1)} = \xi_7^{(a_2)} = \xi_9^{(a_3)} = 0,$$

$$\xi_5^{(a_1)} = -\xi_4^{(a_2)},$$

$$\xi_7^{(a_1)} = -\xi_5^{(a_2)},$$

$$\xi_6^{(a_1)} = -\xi_4^{(a_3)},$$

$$\xi_9^{(a_1)} = -\xi_6^{(a_3)},$$

$$\xi_8^{(a_2)} = -\xi_7^{(a_3)},$$

$$\xi_9^{(a_2)} = -\xi_8^{(a_3)},$$

Optimization

$$\begin{bmatrix} 2\hat{\Theta}(\mathbf{X})^T \hat{\Theta}(\mathbf{X}) & \mathbf{C}^T \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \xi \\ z \end{bmatrix} = \begin{bmatrix} 2\hat{\Theta}(\mathbf{X})^T \dot{\mathbf{X}}(:) \\ d \end{bmatrix}$$

Karush–Kuhn–Tucker (KKT) Constrained Least Squares Optimization

<https://arxiv.org/pdf/1611.03271>

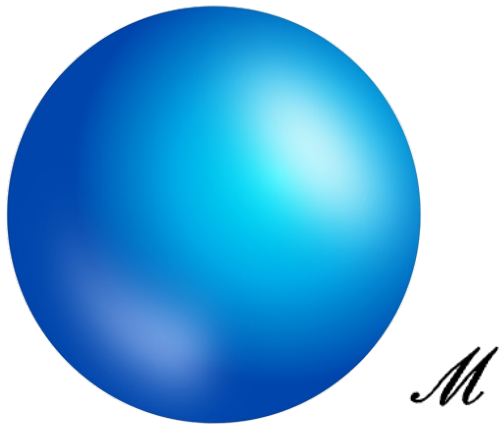
Optimizing Subject to Submanifold/Subspace Constraints

Submanifold Constraint

$$\|y - \hat{y}\|_2^2 + \|\hat{y} - \mathbb{P}_{\mathcal{M}}(\hat{y})\|_2$$

Loss Function

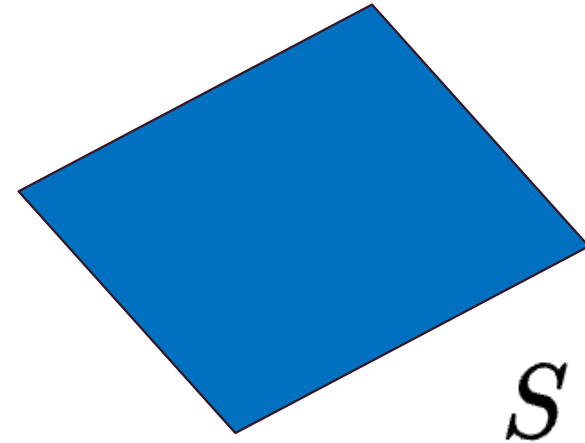
$$\operatorname{argmin}_{\hat{y} \in \mathcal{M}} \|y - \hat{y}\|_2^2$$



Subspace Constraint

$$\|y - \hat{y}\|_2^2 + \|\hat{y} - \mathbb{P}_S(\hat{y})\|_2$$

$$\operatorname{argmin}_{\hat{y} \in S} \|y - \hat{y}\|_2^2$$



PySR: Symbolic Regression

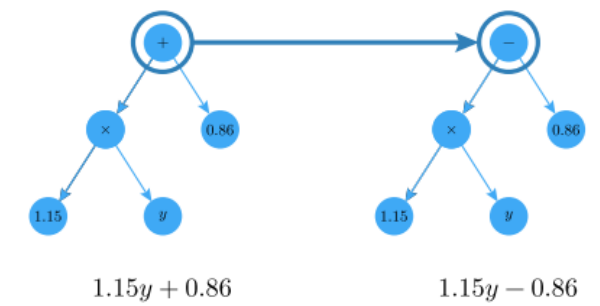


Figure 1: A mutation operation applied to an expression tree.

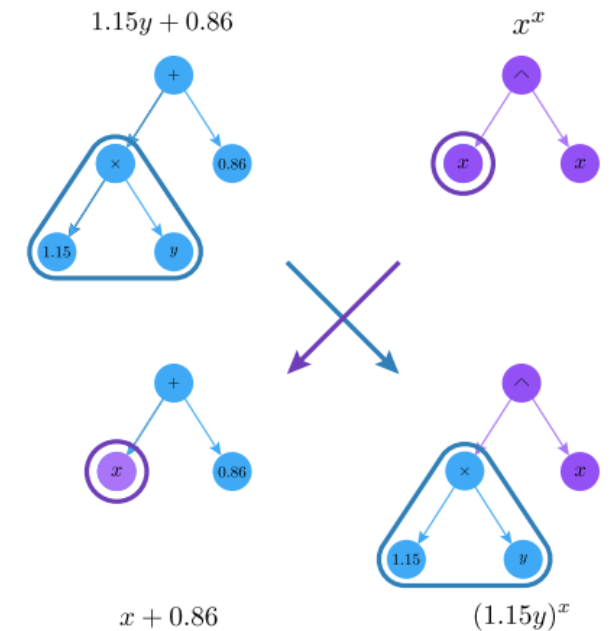


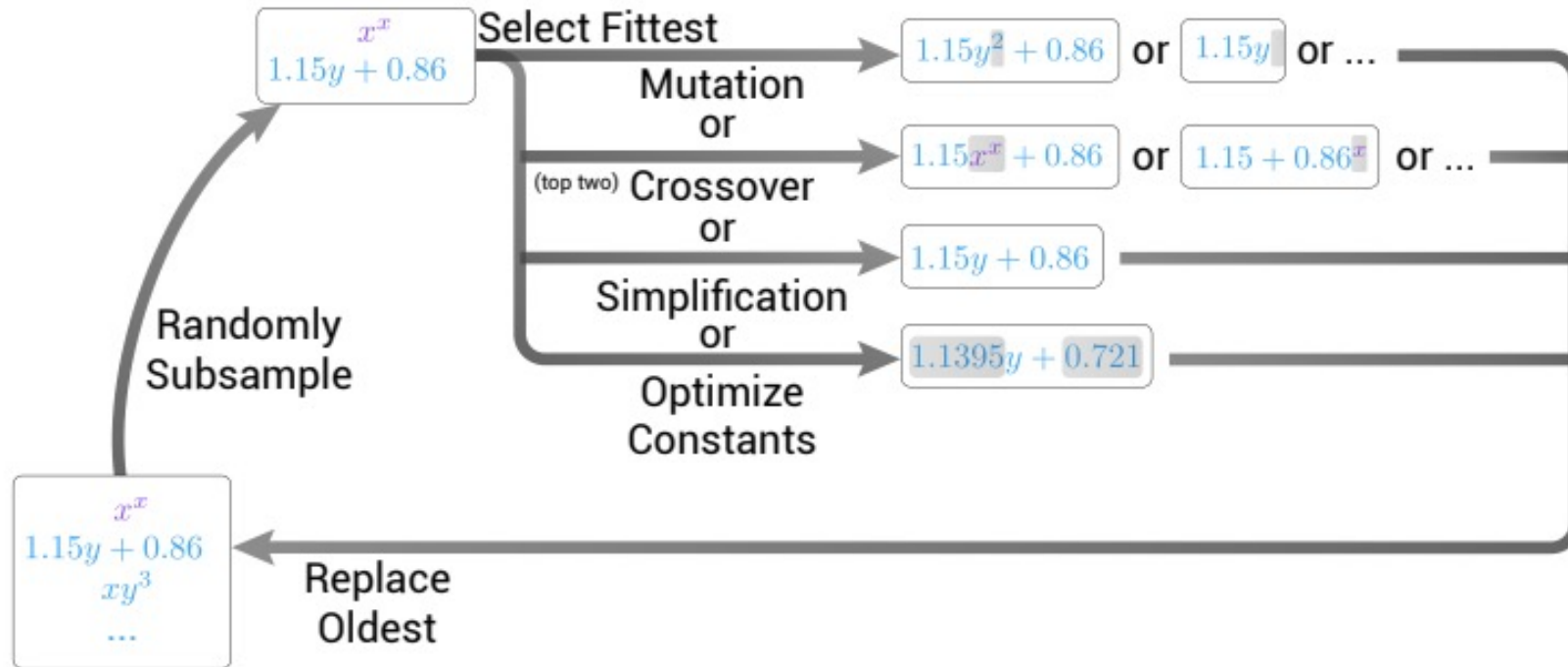
Figure 2: A crossover operation between two expression trees.

<https://arxiv.org/abs/2305.01582>

PySR and SymbolicRegression.jl

PySR: Symbolic Regression

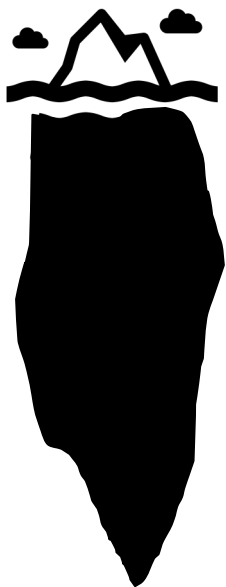
- Evolutionary Optimization



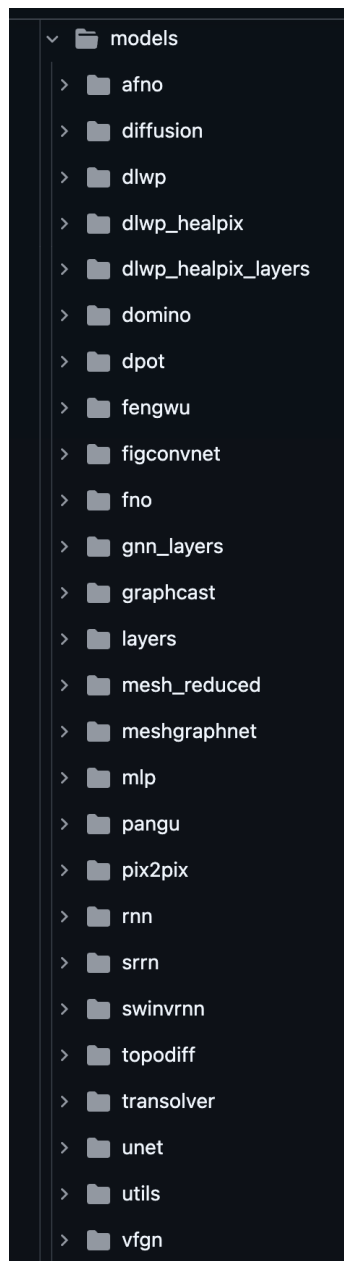
In Summary

Summary

- We have barely scratched the surface of the topic
- There are many types of PIML approaches, e.g., diffusion models
- There are many variants of each architecture

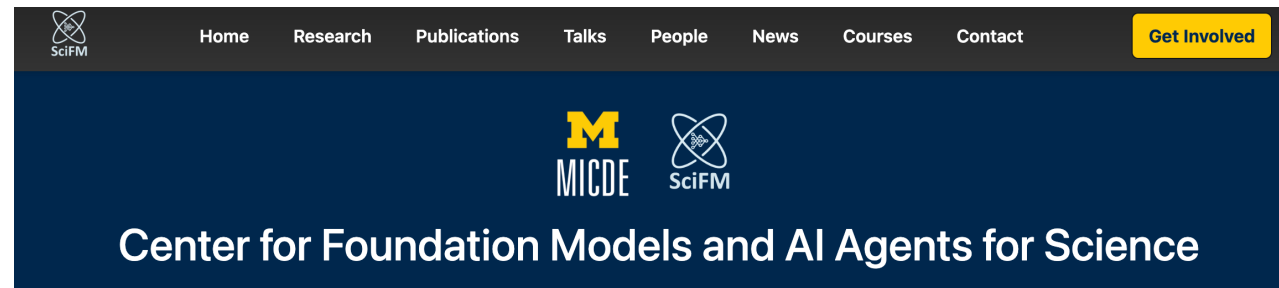


Occam's Razor is still valid – use PIML only when necessary



<https://github.com/NVIDIA/physicsnemo/tree/main/physicsnemo/models>

Scientific Foundation Models



Scientific foundation models extend the breakthroughs of large language models and computer vision into science. Trained on massive unlabeled scientific data — experiments, simulations, papers, instrument logs — they learn generalizable, transferable knowledge across chemistry, materials, biology, and beyond without task-specific supervision. Coupled with AI agents that can plan experiments, run tools, and interpret results, they accelerate discovery from hypothesis generation to validation.

SciFM at Michigan

This effort started as an MICDE Initiative on Scientific Foundation Models in May 2023, and morphed into the Center for Foundation Models & AI Agents for Science in September 2024. We are supported by Los Alamos National Laboratory, NSF NAIRR, NVIDIA, OpenAI and Microsoft.

Fun Fact

According to GPT-5, the words 'Scientific Foundation Models' first appeared on the internet in July 2023. If this is accurate, we were among the first to introduce this term.

Who Are We?

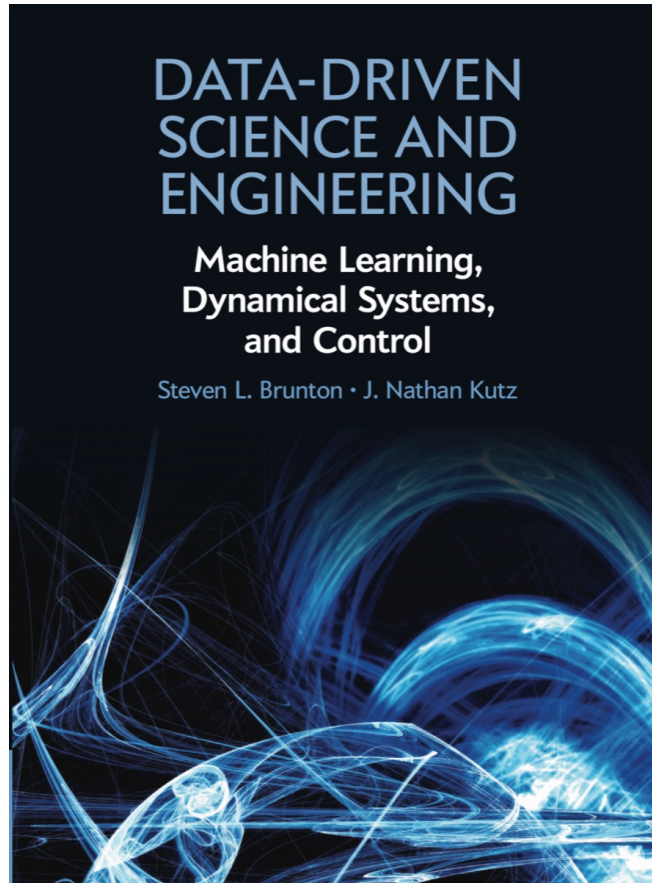
We are a multidisciplinary team combining expertise in physics, chemistry, materials science, and computer science. Our group brings together PIs, postdocs, and students from Michigan, along with our collaborators across the globe, to build foundation models that are scalable and practically useful. We collaborate closely with domain experts to ensure models respect scientific constraints, are interpretable, and can be integrated directly into experimental and simulation workflows.

Vision

At SciFM, we envision a future where scientific discovery is accelerated by foundation models and AI — where computations don't just support research, but actively generate new knowledge and deliver direct solutions to today's most pressing challenges.

<https://scifm.ai/>

Resources



Steven L. Brunton and J. Nathan Kutz
<https://datatoolbox.com/> + YouTube

neuraloperator is a comprehensive library for learning neural operators in PyTorch. It is the official implementation of Fourier Neural Operators and Tensorized Neural Operators.

Unlike regular neural networks, neural operators enable learning mapping between function spaces, and this library provides all of the tools to do so on your own data.

NeuralOperators are also resolution invariant, so your trained operator can be applied on data of any resolution.

[NeuralOperators in PyTorch](#)

242 results (178 ms)

- luluvv/deeplxde: A library for scientific machine learning and physics-informed learning
- jdotoscano94/Learning-Scientific_Machine_Learning_Residual_Based_Attention_PINNs_PIKANs_DeepONets: Physics Informed Machine Learning Tutorials (Pytorch and Jax)
- SciML/NeuralPDE.jl: Physics-Informed Neural Networks (PINN) Solvers of (Partial) Differential Equations for Scientific Machine Learning (SciML) accelerated s...
- xiaohyturbulence-modeling-PIML: Data-driven Reynolds stress modeling with physics-informed machine learning

[Github Code/Libraries](#)

Physics-Informed Machine Learning

Extend deep learning workflows in areas of physics-informed machine learning (PIML) and physics-informed neural networks (PINNs).

Use Deep Learning Toolbox™ for physics-informed machine learning (PIML) and physics-informed neural networks (PINNs).

Physics-informed machine learning (PIML) and physics-informed neural networks refer to machine learning and deep learning concepts where you can integrate laws and principles of physical systems into your machine learning models. Integrating these concepts can improve accuracy and robustness in these models and can help ensure that the model predictions also follow such laws and principles. For example, you can train a neural network that models heat transfer using a loss function that incorporates laws of thermodynamics.

[MathWorks Deep Learning Toolbox for PIML and PINNs](#)

NVIDIA PhysicsNeMo

NVIDIA PhysicsNeMo is an open-source Python framework for building, training, and fine-tuning physics AI models at scale.

NVIDIA PhysicsNeMo provides utilities that enable developers to build AI surrogate models that combine physics-driven causality with simulation and observed data, enabling real-time predictions. From neural operators and GNNs to generative AI models, developers can develop proprietary AI models to enhance engineering simulations and generate higher-fidelity data for scalable, responsive designs. NVIDIA PhysicsNeMo supports the creation and validation of large-scale digital twin models across various physics domains, from computational fluid dynamics and structural mechanics to electromagnetics.

Use NVIDIA PhysicsNeMo to bolster your engineering simulations with AI. You can build and validate models for enterprise-scale digital twin applications across multiple physics domains, from computational fluid dynamics and structural to electromagnetics.

[NVIDIA PhysicsNeMo](#)



Thank You!

Questions?