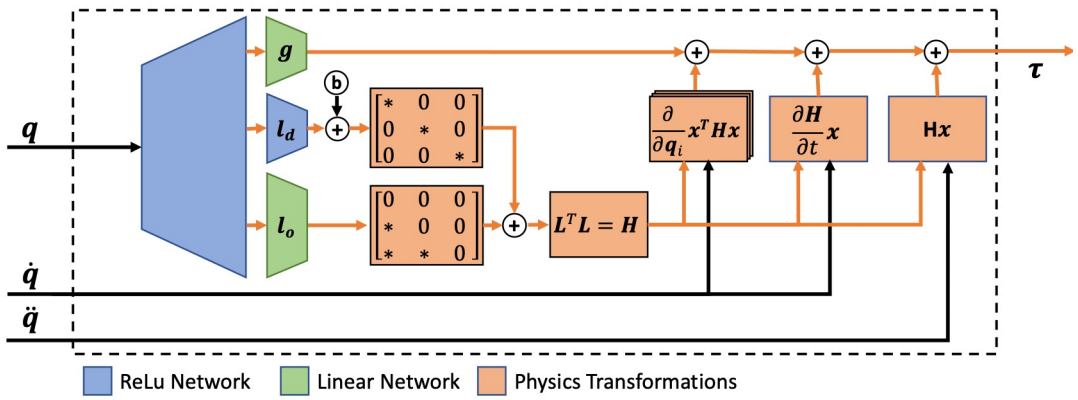
Physics-Encoded Architectures Part II: From Hybrid to Topology Approaches

Dr. Mattia Piccinini &
Prof. Gastone Pietro Rosati Papini

University of Trento &
Technical University of Munich,
Autonomous Vehicle Systems Lab



Lagrangian & Hamiltonian Approaches

$\mathcal{L} = \frac{1}{2}\dot{q}^T M(q)\dot{q} - \mathcal{V}(q)$

$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$

Learning dynamics from variational principles

NODE & Variational Integrator Networks

$\frac{dq}{dt} = f_{NN}(\cdot)$

variational integrator

Learning dynamics via structure-preserving integration

Hybrid Physics-Neural

Residual Learning

Physics Model +

Subsystems Learning

Physics

Physics

Sensor Processing

Physics

Physics-Encoded Topology Learning

cos

sin

tanh

Learning connectivity from physics priors

Physics-Encoded Neural Operators

ϕ_{physics} *Physics Lifting*

Neural Operator G_θ

Model-Structured

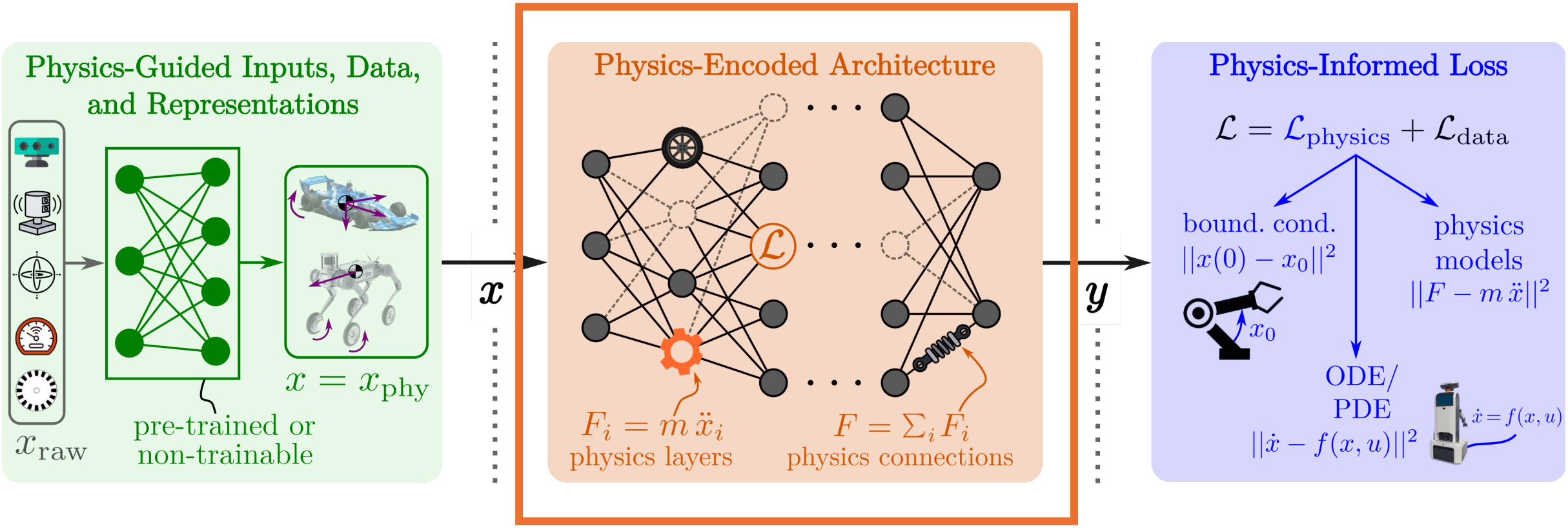
T_w

F_d

F_x

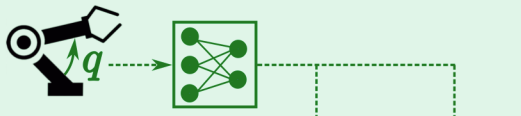
Physics-based layers, connections, constraints

Physics-Encoded Architectures: Positioning



Physics-Encoded Learning Architectures: Multiple Classes

Lagrangian & Hamiltonian Approaches

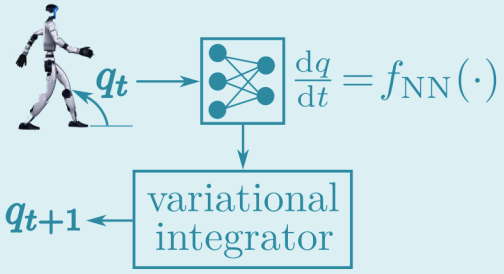


$$\mathcal{L} = \tfrac{1}{2}\,\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} - \mathcal{V}(\boldsymbol{q})$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

Learning dynamics from variational principles

NODE & Variational Integrator Networks



Learning dynamics via structure-preserving integration

Physics-Encoded Learning Architectures: Multiple Classes

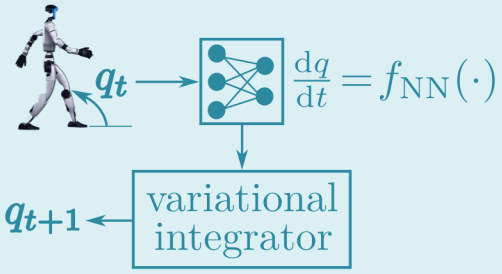
Lagrangian & Hamiltonian Approaches

$$\mathcal{L} = \tfrac{1}{2}\,\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} - \mathcal{V}(\boldsymbol{q})$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

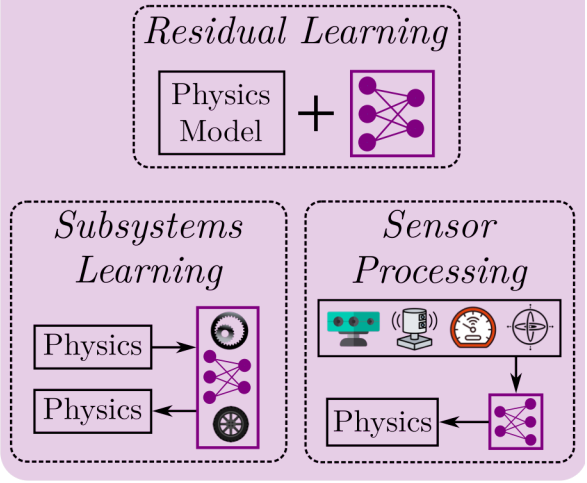
Learning dynamics from variational principles

NODE & Variational Integrator Networks



Learning dynamics via structure-preserving integration

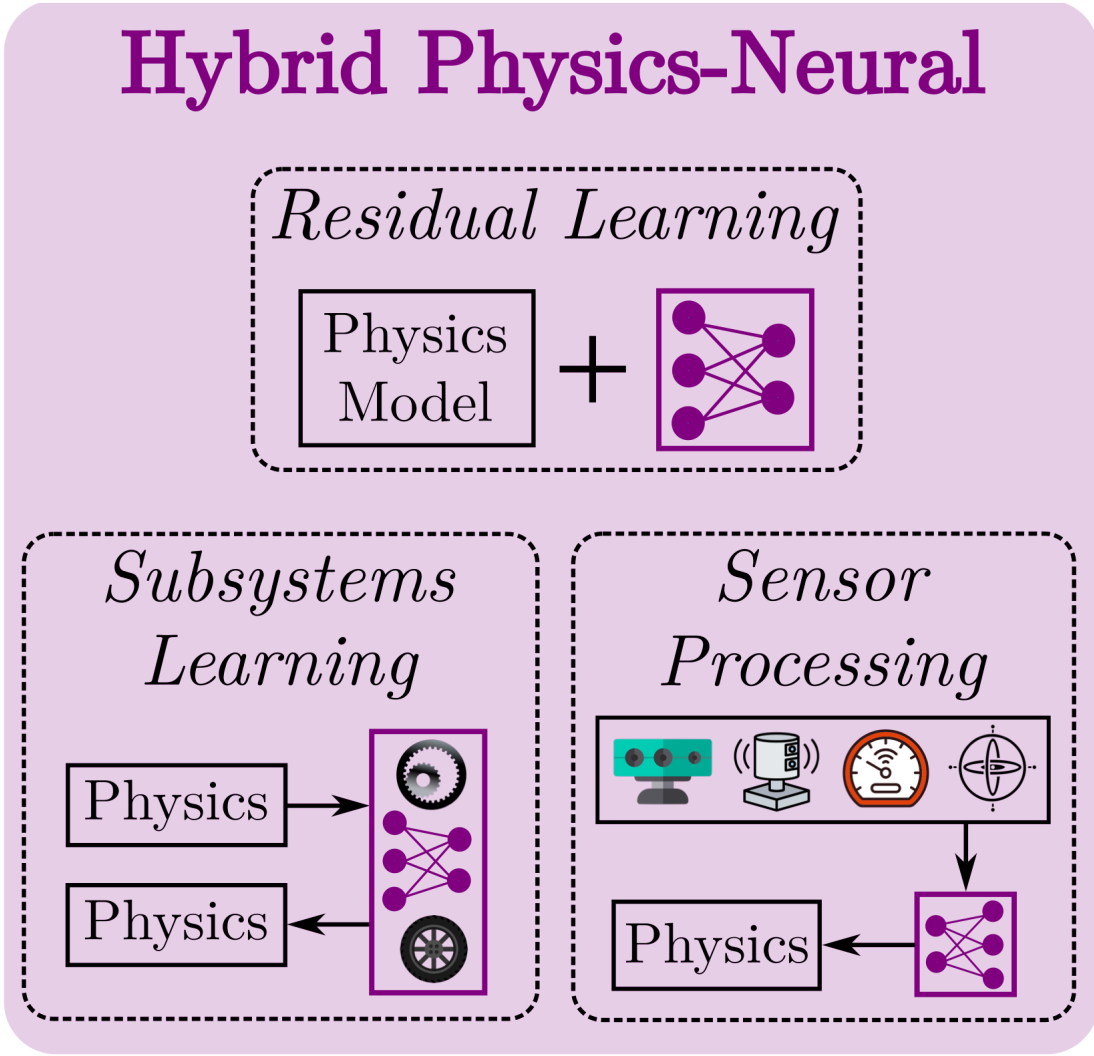
Hybrid Physics-Neural



Hybrid Physics-Neural Approaches

Modular combination of physics-based and neural network (or ML) models

- Use NNs as **separate modules**, interacting with physics-based ones
- No physics knowledge/constraints **inside** the NN modules → general-purpose design
- **Aims**:
  - **Correct** physical models
  - Learn **specific** parts of the system
  - Preprocess **sensor** data

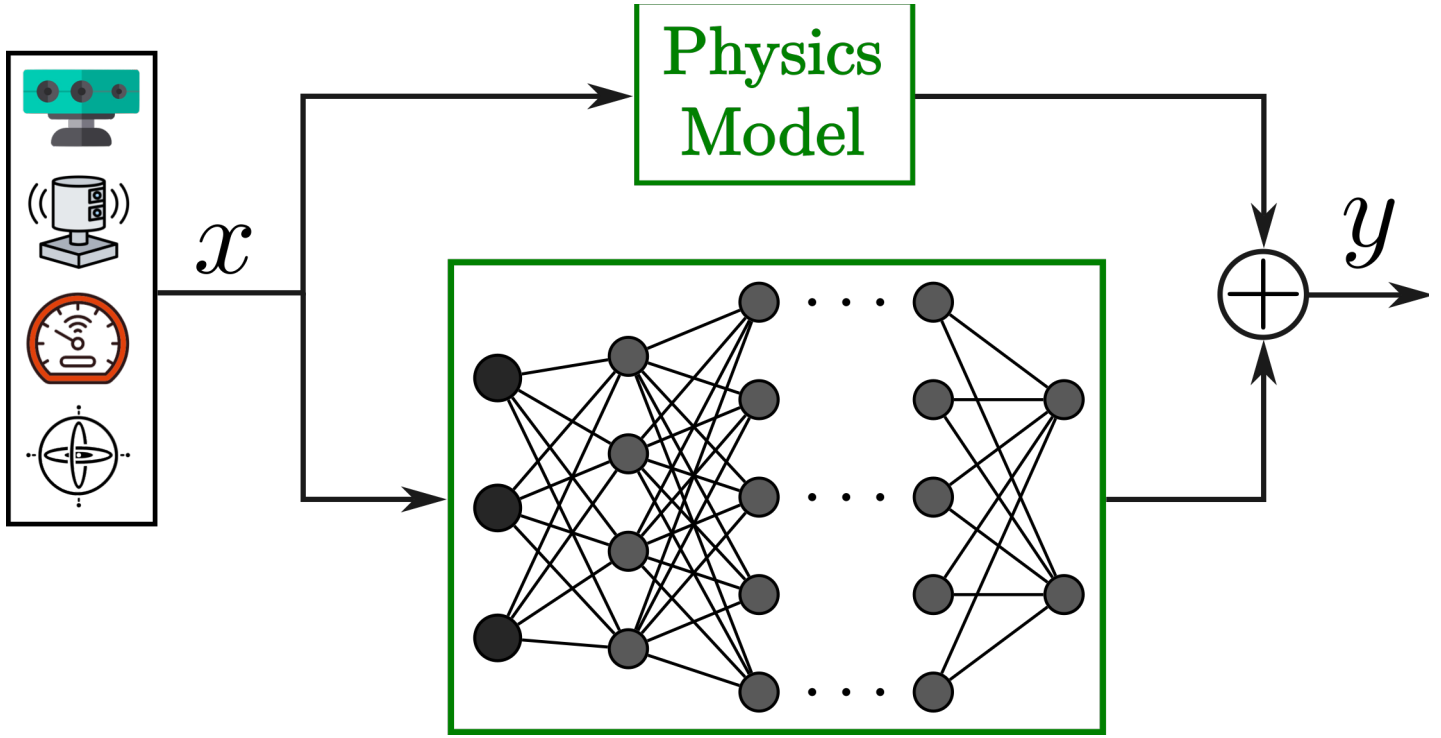



Residual
Learning

Residual Learning

Idea: Learn **discrepancies** between a physical model and the real data

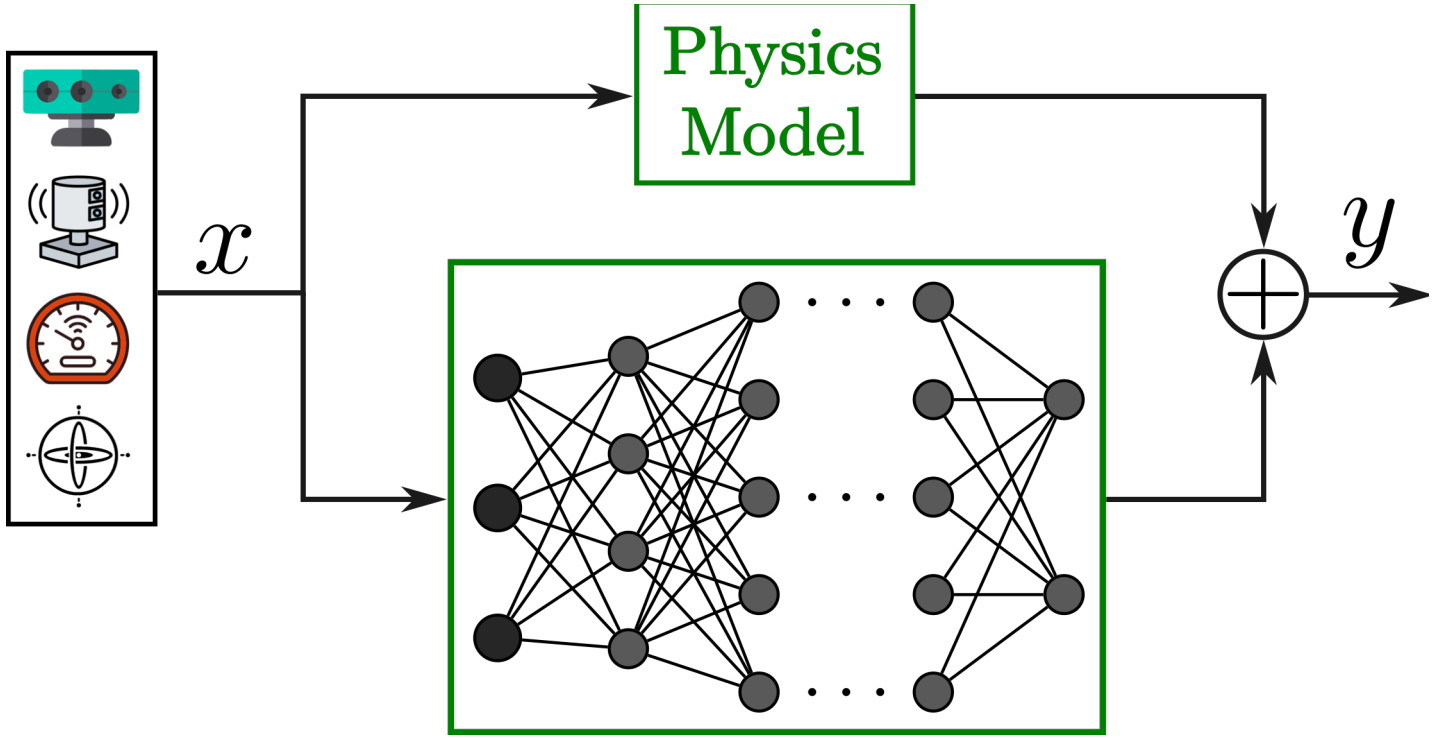
- Model mismatches
- Parameter uncertainties
- Model assumptions



Residual Learning

Idea: Learn **discrepancies** between a physical model and the real data

- Historically, **among the first** attempts to combine data and physics
- Used a lot in robotics and other physical systems



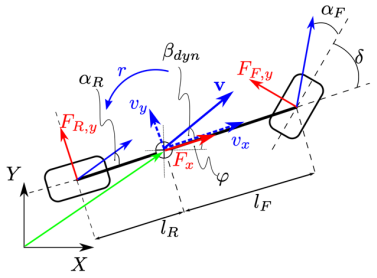
Residual Learning - Examples

Learn the residuals of a **physics-based** single-track vehicle model with **Gaussian Process Regression (GPR)**

Physics-based model

$$\dot{\mathbf{x}} = \begin{bmatrix} v_x \cos\varphi - v_y \sin\varphi \\ v_x \sin\varphi + v_y \cos\varphi \\ r \\ \frac{1}{m}(F_x - F_{F,y}\sin\delta + m v_y r) \\ \frac{1}{m}(F_{R,y} + F_{F,y}\cos\delta - m v_x r) \\ \frac{1}{I_z}(F_{F,y} l_F \cos\delta - F_{R,y} l_R + \tau_{\mathrm{TV}}) \\ \Delta\delta \\ \Delta T \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{B}_d(\mathbf{d}(\mathbf{z}_k) + \mathbf{w}_k)$$



GPR Residuals

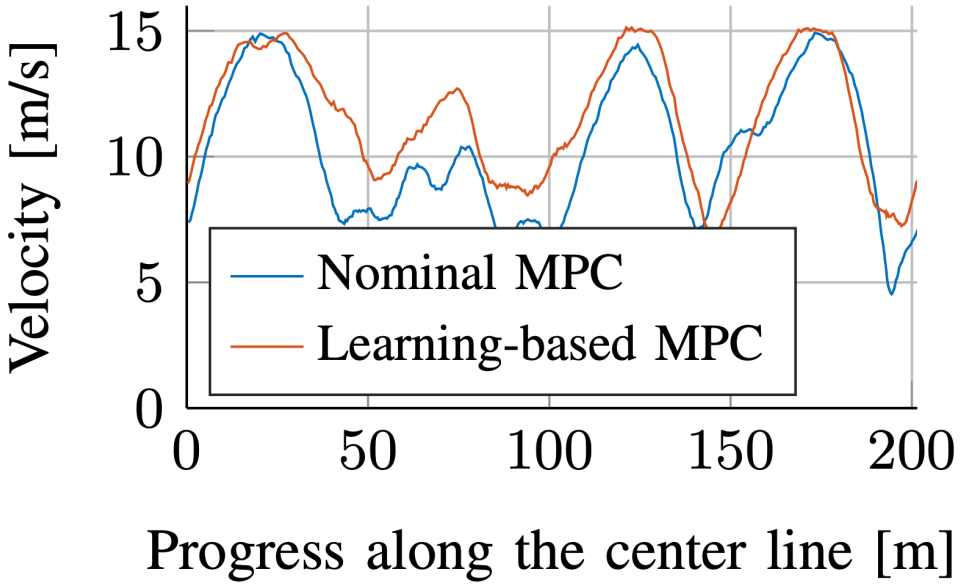
J. Kabzan, L. Hewing, A. Liniger, M. Zeilinger, “Learning-based Model Predictive Control for Autonomous Racing”, *IEEE RA-L*, 2019

Residual Learning - Examples

Vehicle model integration in an **MPC controller** for **autonomous racing**

- **Online learning** of the GPR residuals
- Lap time **improvement** by **10%** w.r.t. purely physics-based model

$$
\begin{aligned}
\min_{\{\mathbf{u}_k, v_k\}} \quad & \sum_{k=1}^{N} J(\boldsymbol{\mu}_k^{\mathbf{x}}, \theta_k, v_k) + U(\boldsymbol{\mu}_k^{\mathbf{x}}, \mathbf{u}_k) + L(\boldsymbol{\mu}_k^{\mathbf{x}}) \\
\text{s.t.} \quad & \boldsymbol{\mu}_0^{\mathbf{x}} = \mathbf{x}(t), \\
& \boldsymbol{\mu}_{k+1}^{\mathbf{x}} = \mathbf{f}(\boldsymbol{\mu}_k^{\mathbf{x}}, \mathbf{u}_k) + \mathbf{B}_d \tilde{\boldsymbol{\mu}}^d(\boldsymbol{\mu}_k^{\mathbf{z}}), \\
& \theta_{k+1} = \theta_k + v_k, \\
& (11) \text{ with } \overline{\boldsymbol{\Sigma}}_k^{XY}, \overline{\theta}_k, \\
& (12), (13),
\end{aligned}
$$



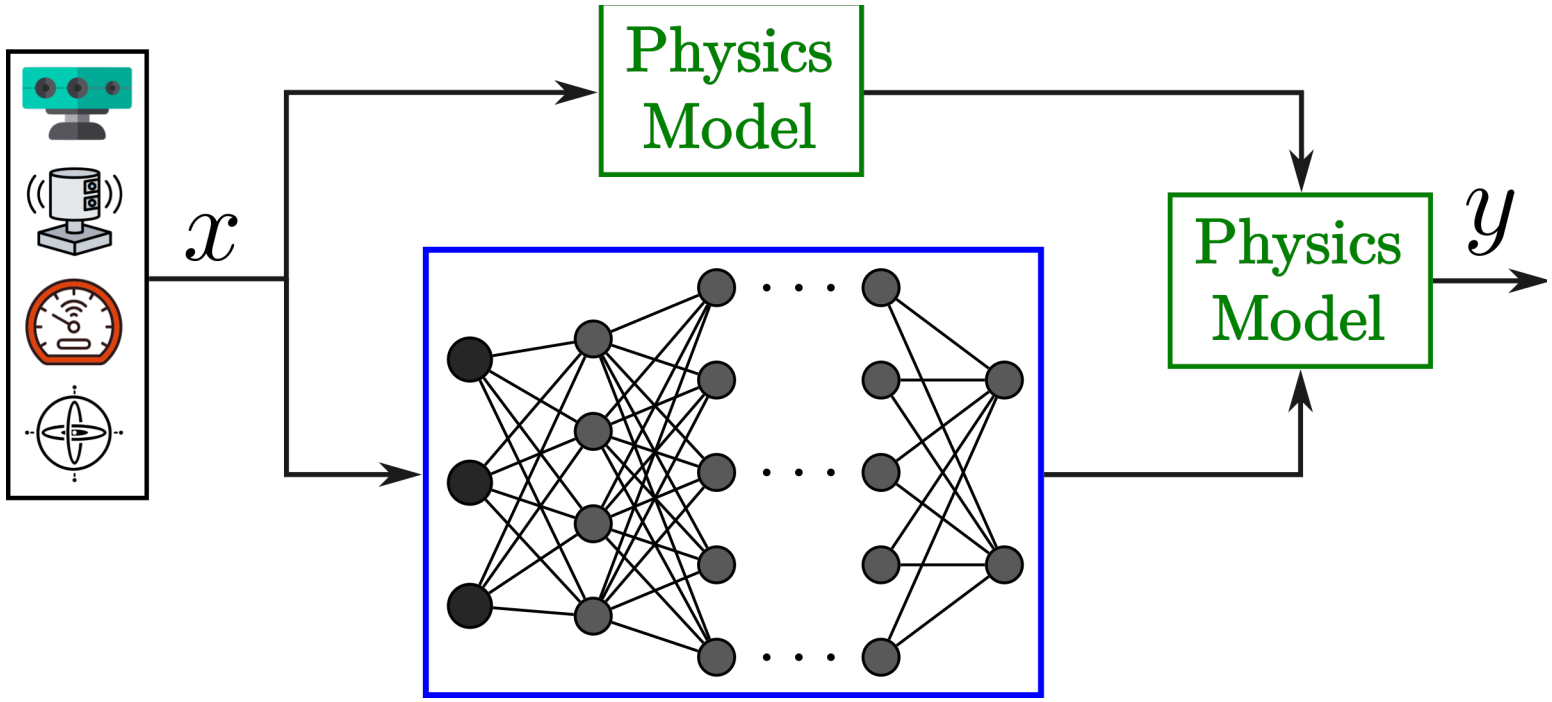
J. Kabzan, L. Hewing, A. Liniger, M. Zeilinger, "Learning-based Model Predictive Control for Autonomous Racing", *IEEE RA-L*, 2019


Sub-systems
Learning

Sub-sytems Learning

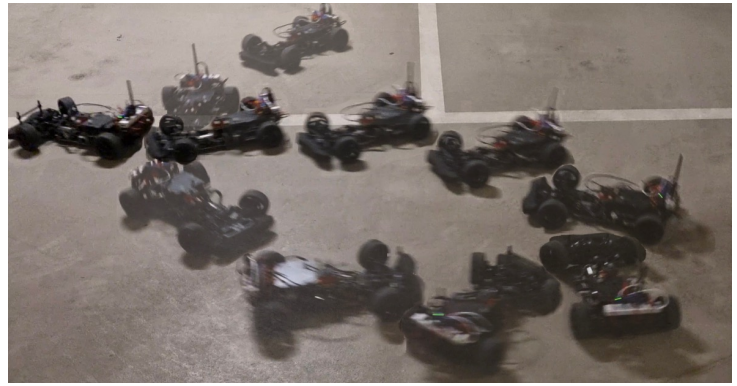
Idea: Learn **complex sub-systems** of physics-based models

- Their physical **parametrization** can be expensive, time-consuming, or require expertise
- Part of the model becomes **data-driven**, the rest remains **physics-based**
- **Examples** of sub-systems which can be learned with NNs: vehicle tires, robot gearboxes



Sub-sytems Learning - Examples

Learning vehicle dynamics for **velocity estimation**



Physics-based vehicle model

$$\dot{v}_x = \frac{1}{m}\left(F_{x_r} + F_{x_f}\cos(\delta) - F_{y_f}\sin(\delta) - F_{drag} + m v_y r\right),$$

$$\dot{v}_y = \frac{1}{m}\left(F_{x_f}\sin(\delta) + F_{y_r} + F_{y_f}\cos(\delta) - m v_x r\right),$$

$$\dot{r} = \frac{1}{I_z}\left(\left(F_{x_f}\sin(\delta) + F_{y_f}\cos(\delta)\right) l_f - F_{y_r} l_r\right),$$

$$\dot{\omega}_s = \frac{1}{I_e}\left(k_\phi I_q - R \cdot F_{x_f} - R \cdot F_{x_r} - \tau_t\right),$$

Issue: identifying **tire forces** requires ad-hoc experiments & machines + is time-consuming

Proposed solution: use **NNs to learn tire forces** $[F_{x_r}, F_{x_f}, F_{y_r}, F_{y_f}] = F_{NN}(x, u)$

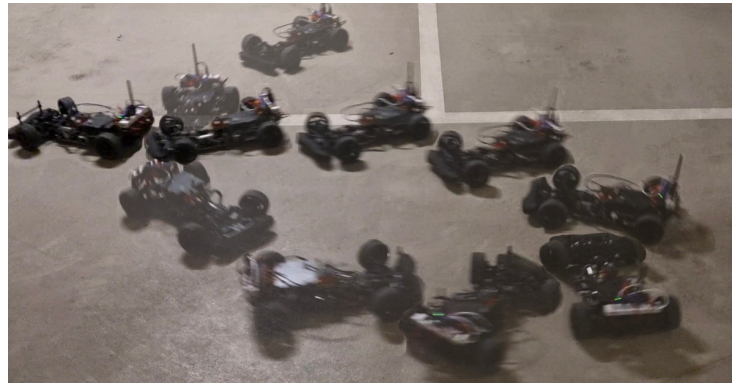
J. Wegrzynowski, G. Czechmanowski, P. Kicki, K. Walas, “Learning dynamics models for velocity estimation in autonomous racing”, *IEEE IROS*, 2024

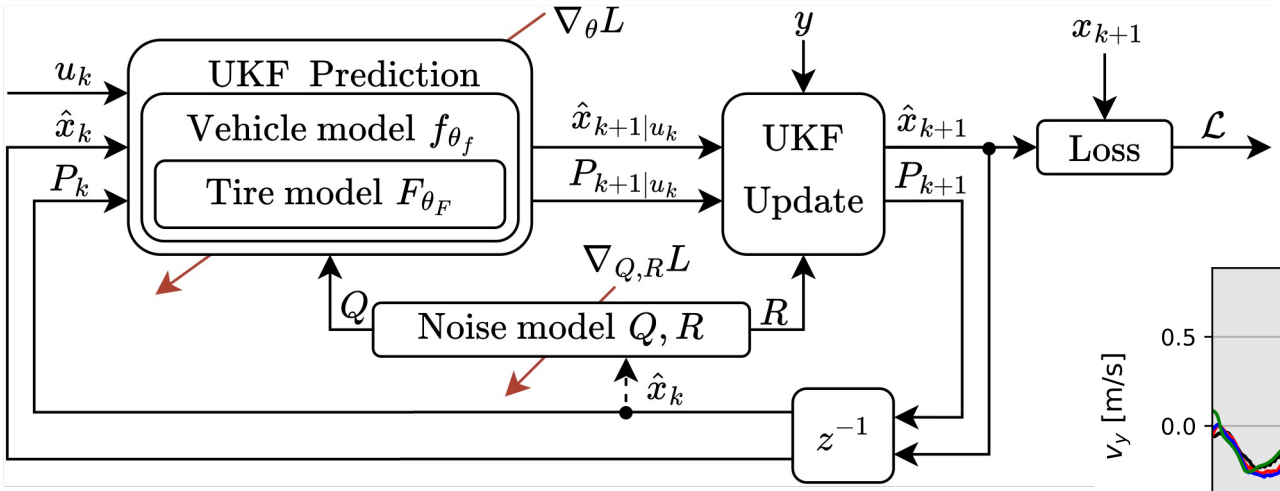
Sub-sytems Learning - Examples

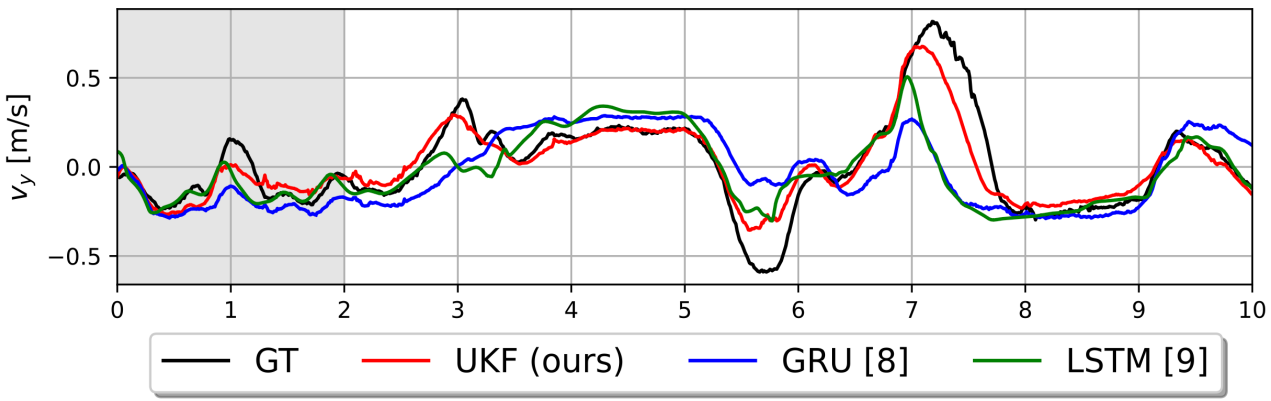
Vehicle + neural tire model integrated in an

Unscented Kalman Filter (UKF) for velocity estimation

- **Outperforming** general-purpose GRU and LSTM
- **Zero-shot generalization** to unseen road conditions







J. Wegrzynowski, G. Czechmanowski, P. Kicki, K. Walas, “Learning dynamics models for velocity estimation in autonomous racing”, *IEEE IROS*, 2024

Questions & Discussion

Physics-Encoded Learning Architectures: Multiple Classes

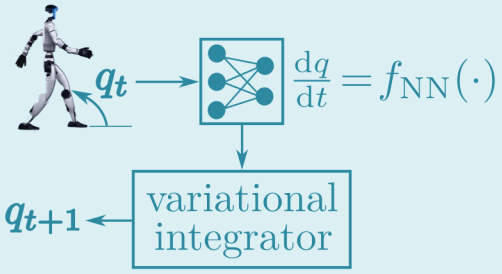
Lagrangian & Hamiltonian Approaches

$$\mathcal{L} = \tfrac{1}{2}\,\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} - \mathcal{V}(\boldsymbol{q})$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

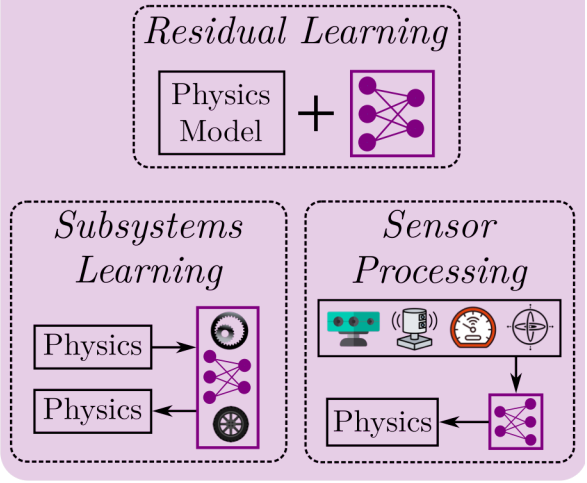
Learning dynamics from variational principles

NODE & Variational Integrator Networks



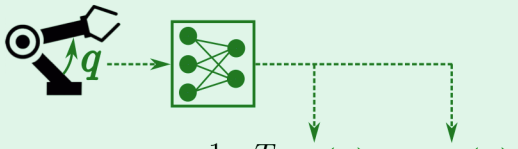
Learning dynamics via structure-preserving integration

Hybrid Physics-Neural



Physics-Encoded Learning Architectures: Multiple Classes

Lagrangian & Hamiltonian Approaches

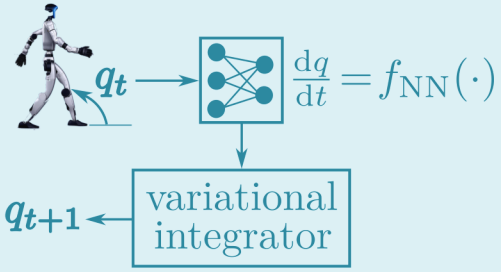


$$\mathcal{L} = \tfrac{1}{2}\dot{q}^T M(q)\dot{q} - \mathcal{V}(q)$$

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$$

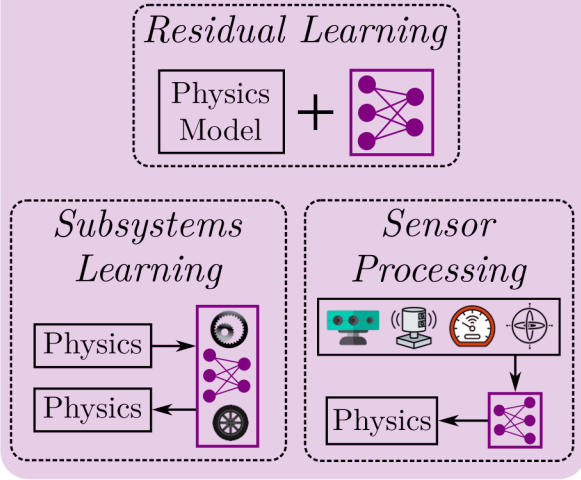
Learning dynamics from variational principles

NODE & Variational Integrator Networks

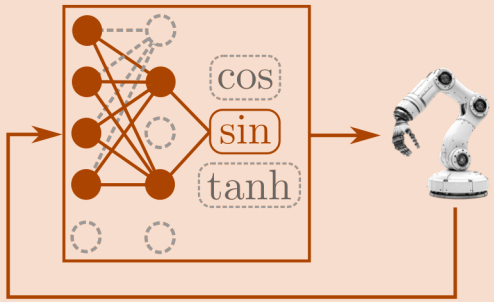


Learning dynamics via structure-preserving integration

Hybrid Physics-Neural

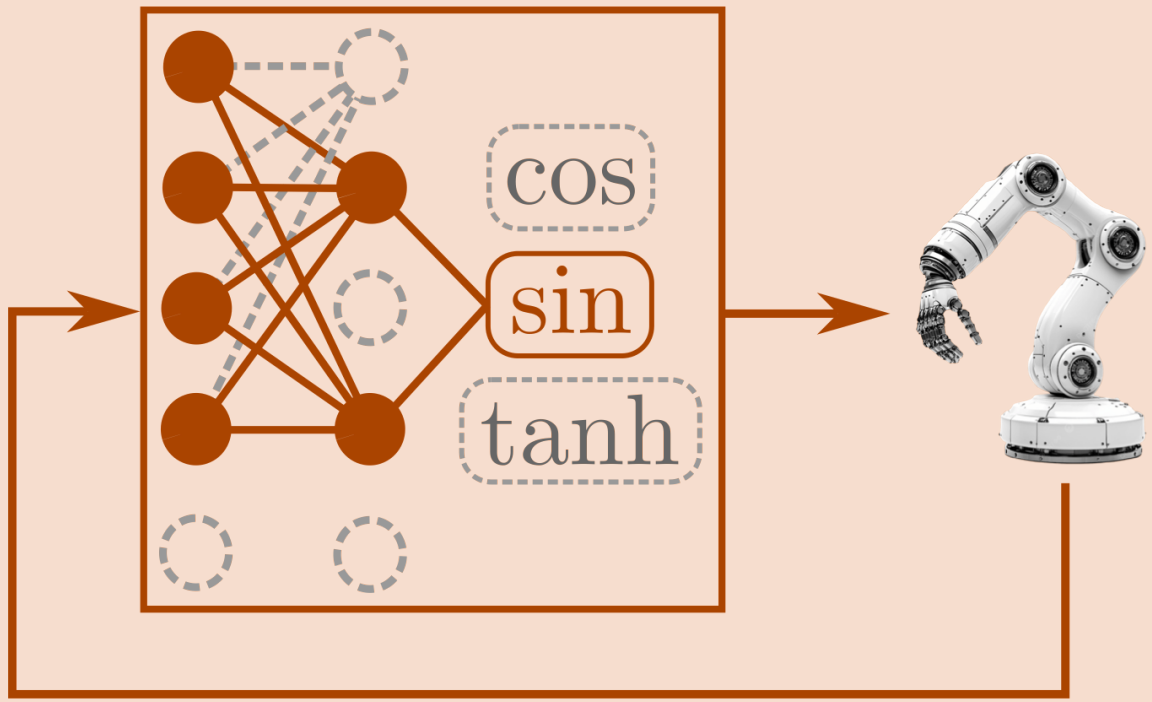


Physics-Encoded Topology Learning



Learning connectivity from physics priors

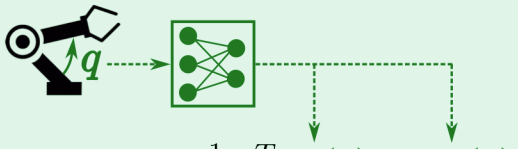
Physics-Encoded Topology Learning



Learning connectivity from physics priors

Physics-Encoded Learning Architectures: Multiple Classes

Lagrangian & Hamiltonian Approaches

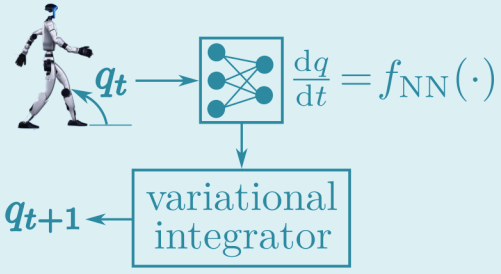


$$\mathcal{L} = \tfrac{1}{2}\dot{q}^T M(q)\dot{q} - \mathcal{V}(q)$$

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$$

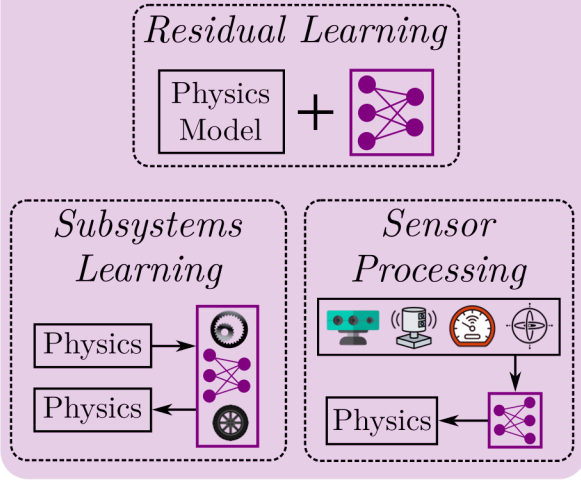
Learning dynamics from variational principles

NODE & Variational Integrator Networks

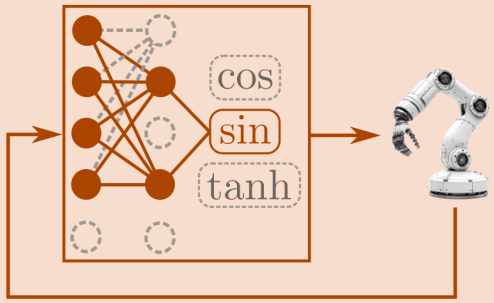


Learning dynamics via structure-preserving integration

Hybrid Physics-Neural



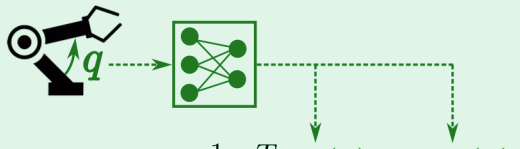
Physics-Encoded Topology Learning



Learning connectivity from physics priors

Physics-Encoded Learning Architectures: Multiple Classes

Lagrangian & Hamiltonian Approaches

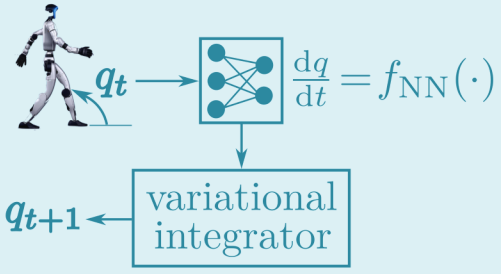


$$\mathcal{L} = \tfrac{1}{2}\dot{q}^T M(q)\dot{q} - \mathcal{V}(q)$$

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) = \tau$$

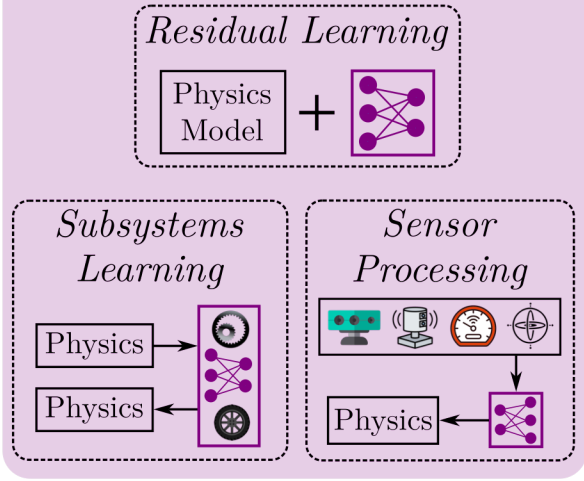
Learning dynamics from variational principles

NODE & Variational Integrator Networks

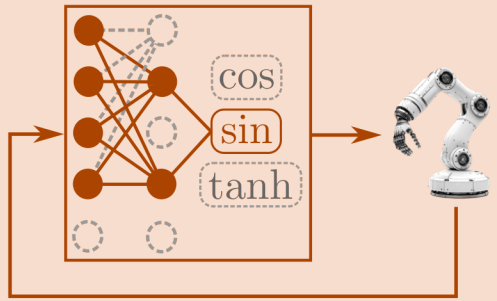


Learning dynamics via structure-preserving integration

Hybrid Physics-Neural

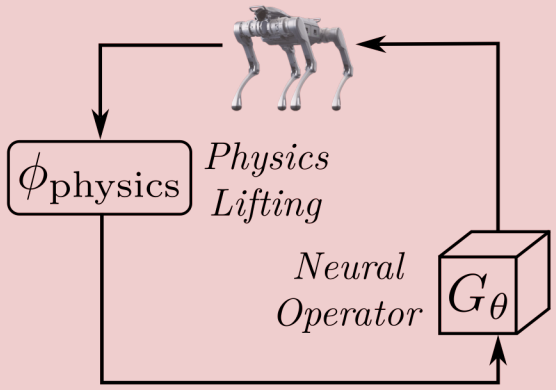


Physics-Encoded Topology Learning



Learning connectivity from physics priors

Physics-Encoded Neural Operators



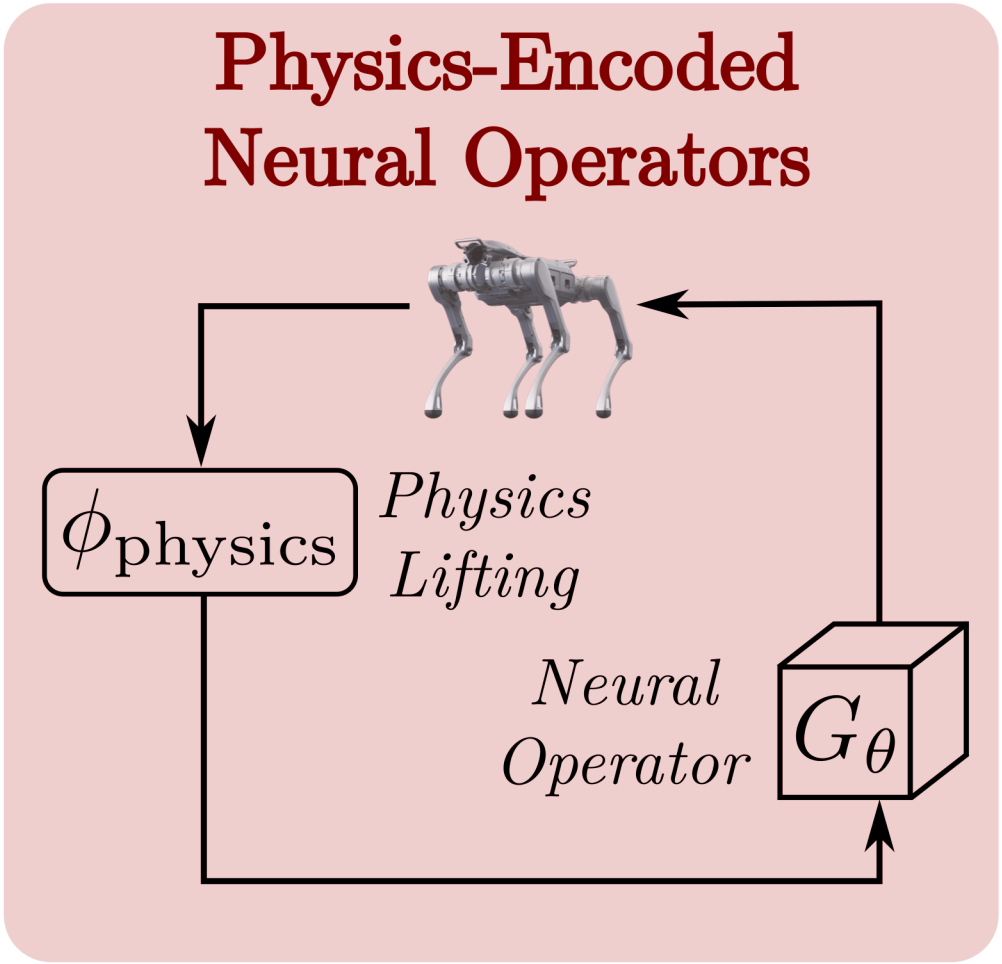
Physics-Encoded Neural Operators

Neural Operators (NOs) are different from **NNs**:

- **NNs** learn from fixed-dimensional **input/output** mappings
- **NOs** learn operators between infinite-dimensional **function spaces**

Examples of NOs:

- **Koopman networks** (most popular)
- Kolmogorov-Arnold networks
- DeepONets
- Graph neural operators

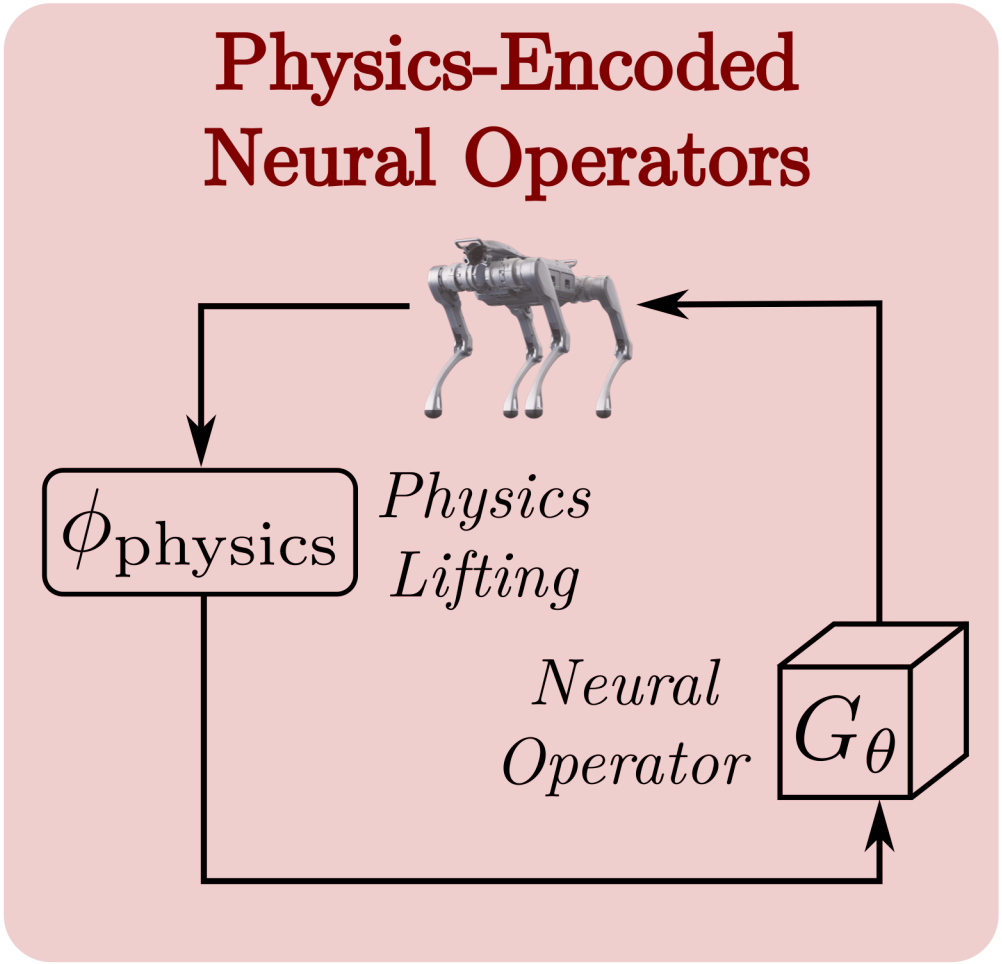


Physics-Encoded Neural Operators

How to **embed physics** in **Neural Operators?**

Example: **Koopman Networks**

- The system is **lifted** to a higher-dimensional space where it becomes approximately **linear**
- Using a set of **lifting functions** (aka dictionary) to map the system to the lifted linear space
- Incorporate **physics priors** into **lifting functions**
  - Different ways to do it (e.g., trigonometric insights)



Questions & Discussion

Funded with the contribution of the Ministry of University and Research pursuant to Decree No. 1236 of 01/08/2023 – FIS 2 Call; project FIS-2023-03684 'Neu4mes', CUP: E53C24003800001