SAPIENZA UNIVERSITÀ DI ROMA

Space Systems Engineering Laboratory

Class of Optimal Space Guidance Problems solved via Indirect Methods and Physics-Informed Neural Networks

[1]*Enrico Schiassi*,

[2]Andrea D'Ambrosio, [1]Andrea Scorsoglio, [1]Roberto Furfaro, and [2]Fabio Curti,

[1]University of Arizona, USA

[2]Sapienza University of Rome

31st AAS/AIAA Space Flight Mechanics Meeting,
Virtual, February 1-4, 2021

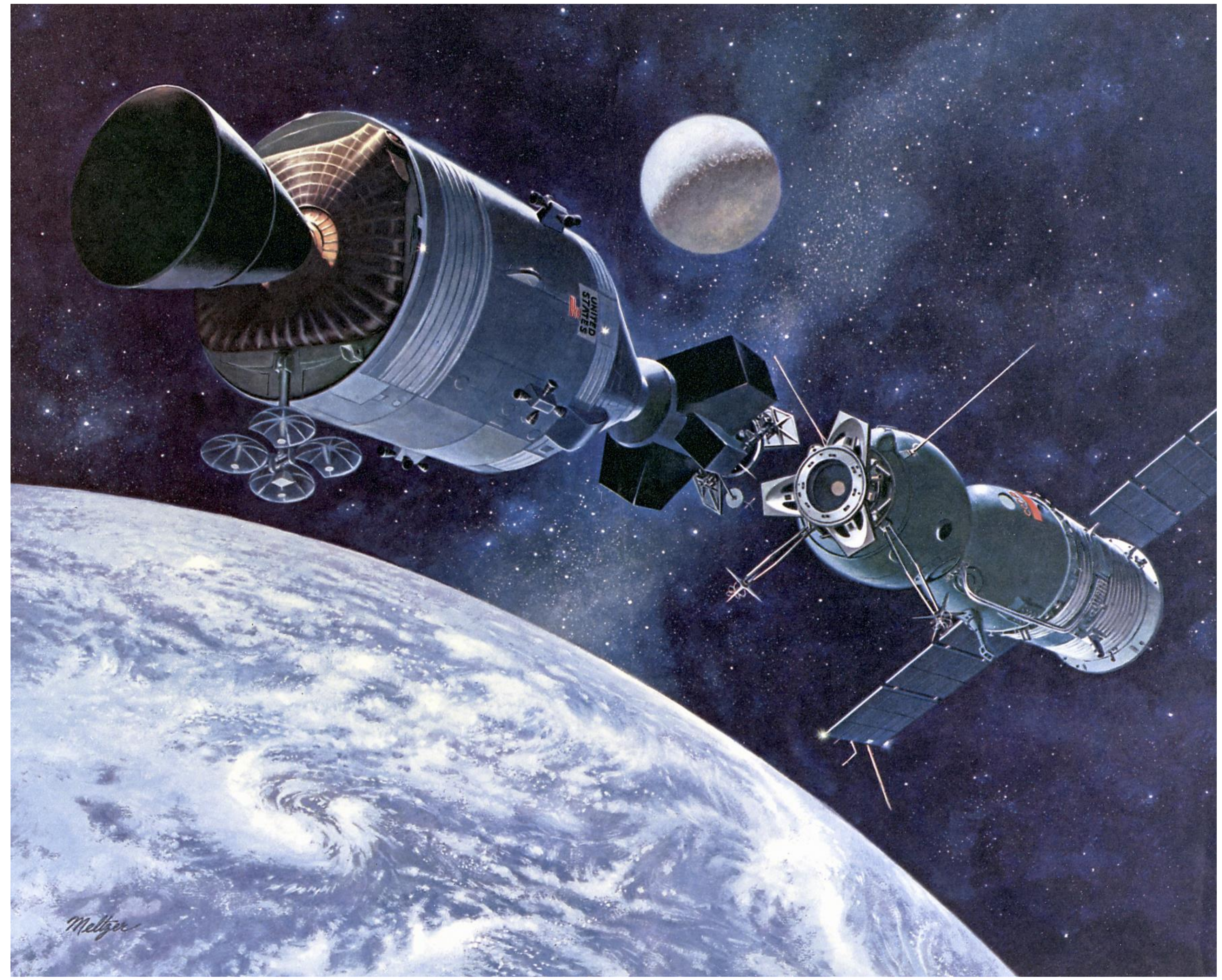
Contents

Introduction: Overview and Motivations

Space Systems
Engineering Laboratory

- Optimal Control Problems (OCPs) represent are among the most interesting optimization problems, in many fields, especially in aerospace engineering.

- Energy Optimal problems such as Landing, Intercept, Rendezvous, etc.; are of extreme interest for space applications
  - It is important to have robust algorithms, eventually suitable for real-time applications.



Introduction: Goals

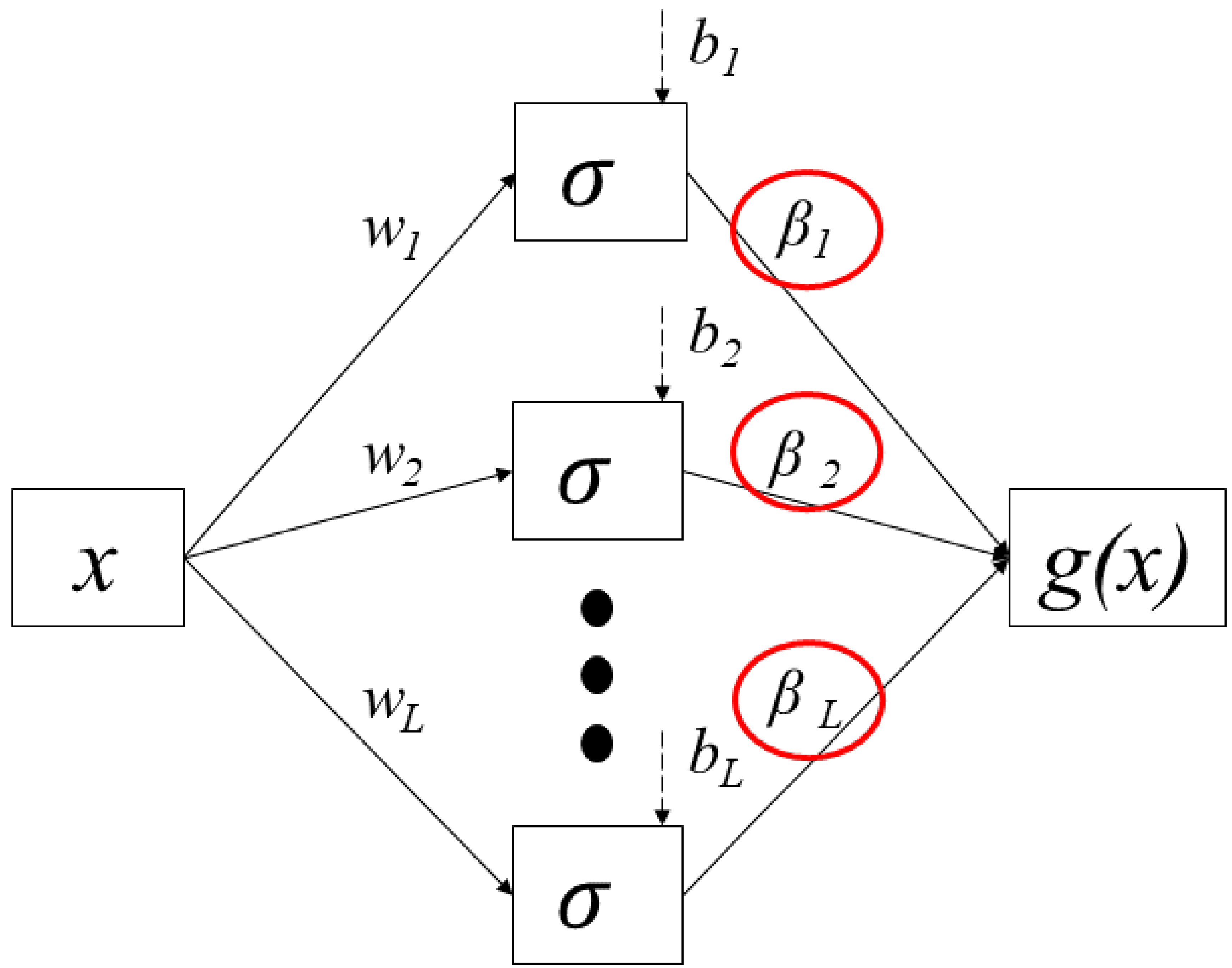
Space Systems Engineering Laboratory

- To employ ***Physics-Informed Neural Networks (PINN)*** to solve a class Optimal Control Problems (OCPs) via indirect method
  - Generic and space applications
- We consider the class of OCPs with integral quadratic cost
- The focus of this talk is to show the effectiveness of PINN based algorithms in solving the class of OCPs considered
- Three different types of PINN frameworks are tested
  - Standard PINN [Raissi et al.]
  - Physics-Informed Extreme Learning Machine (PIELM) [Dwivedi and Srinivasan]
  - Extreme Theory of Functional Connections (X-TFC) [Schiassi et al.]

ELM algorithm

Space Systems
Engineering Laboratory

- ELM is a training algorithm for shallow NN that randomly selects input weights and bias, and computes the output weights via least-square
  - Input weights and bias are not tuned during the training
- The convergence of the ELM algorithm is proved by Huang et al. [2006]
  - The convergence is guaranteed for any input weights and bias randomly chosen to any continuous probability distribution

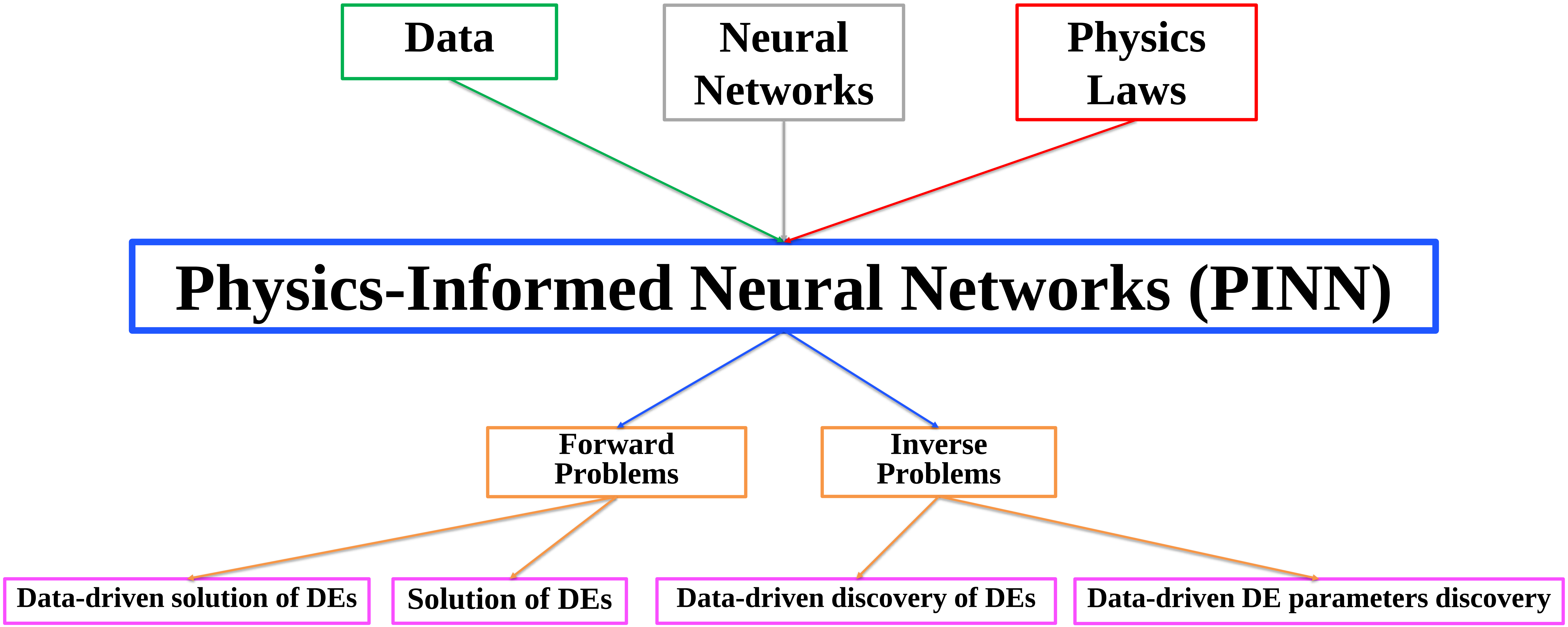


Optimal Control Problems

- Optimal Control Problems (OCPs) are generally hard and computationally expensive.
  - Closed-Loop solutions
  - Open-Loop Solutions
- In general, Open-Loop solutions can be found in two ways
  - **Direct Method**: Transform a continuous problem in a finite NLP problem and find the minimum
  - **Indirect Method**: Apply Pontryagin Minimum Principle (PMP) to derive the necessary conditions for optimality
    - The solution of the OCP reduces to the solution of a Two Point Boundary Value Problem (TPBVP) that is a system of ODEs

Physics-Informed Neural Networks

Space Systems
Engineering Laboratory



Standard PINN

Space Systems
Engineering Laboratory

$$f\left(\mathbf{x}; \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) = 0, \quad \mathbf{x} \in \Omega, \qquad \mathcal{B}(u, \mathbf{x}) = 0 \quad \text{on} \quad \partial\Omega,$$

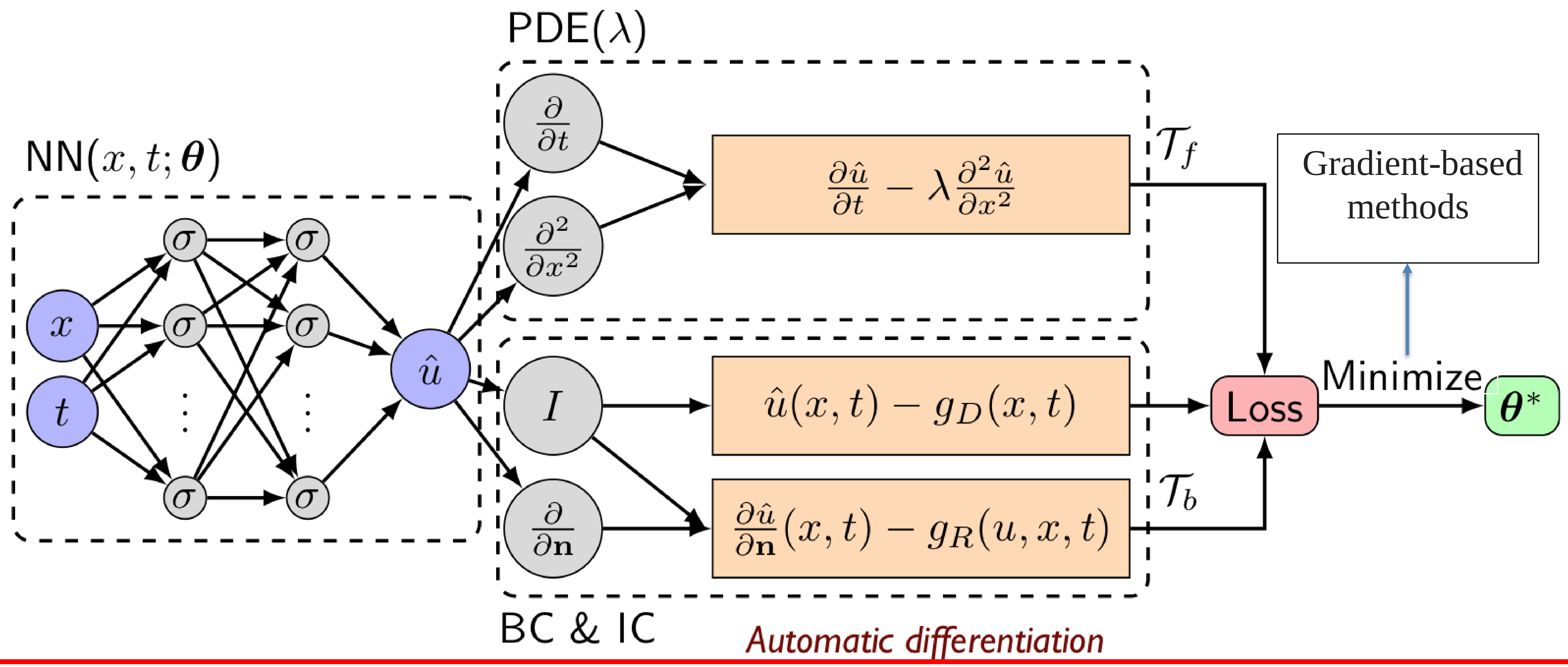


Image taken from: Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E., 2019. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*.

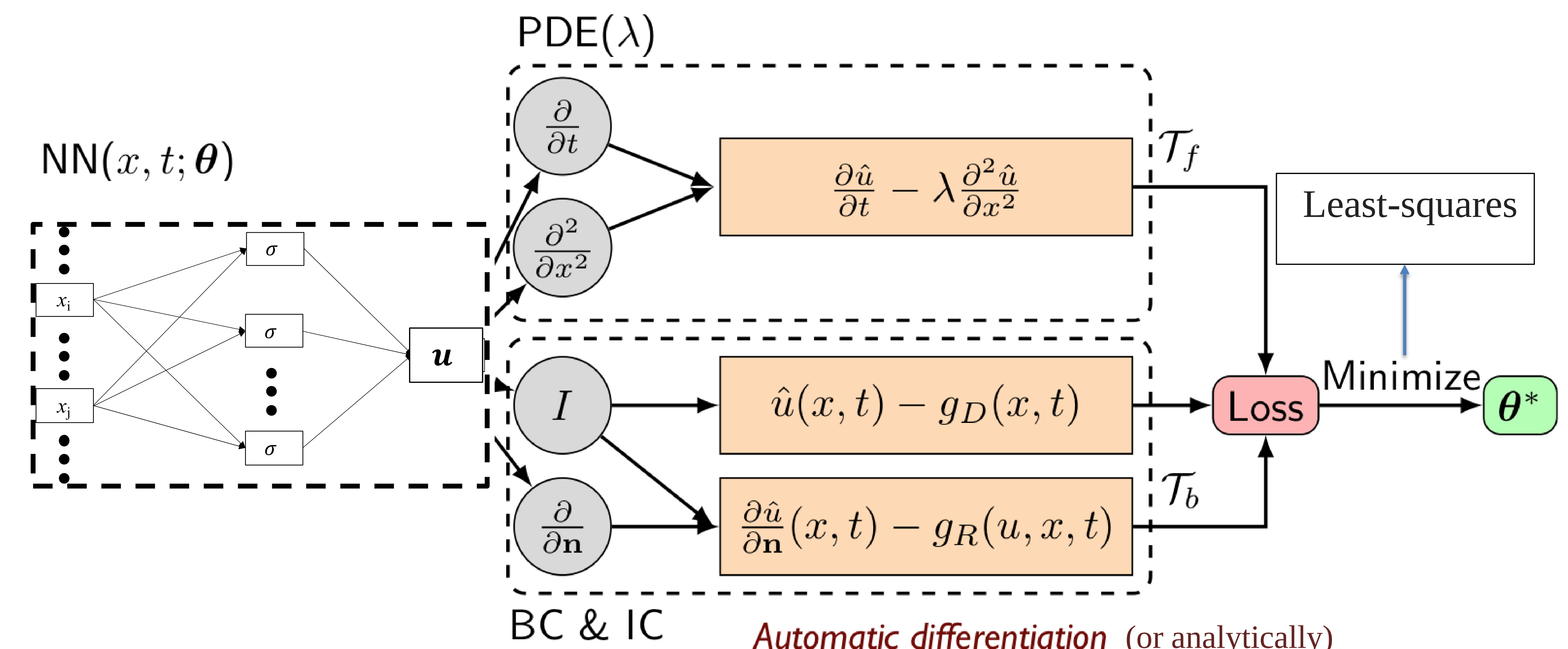
Main Limitations

1. Gradient based methods used to minimize the loss
   - Computationally expensive
2. Initial Conditions and/or Boundary Conditions are not analytically satisfied
   - Increases the computational cost
   - Can cause gradient pathologies (especially when DNN are used)

PIELM

Space Systems
Engineering Laboratory

$$f\left(\mathbf{x}; \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \ldots; \boldsymbol{\lambda}\right) = 0, \quad \mathbf{x} \in \Omega, \qquad \mathcal{B}(u, \mathbf{x}) = 0 \quad \text{on} \quad \partial\Omega,$$

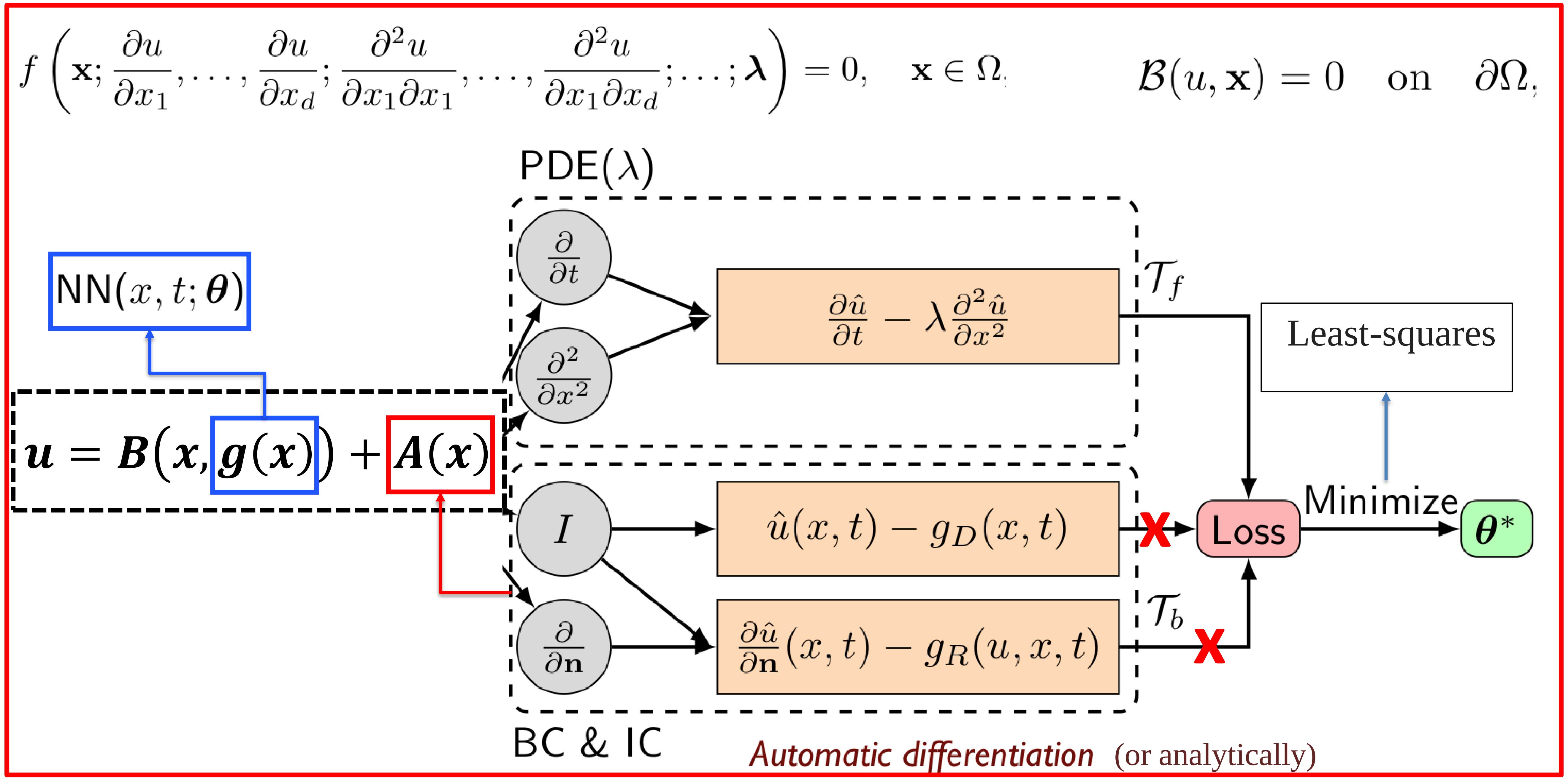


Main Limitations

1. ~~Gradient based methods used to minimize the loss~~
   - ~~Computationally expensive~~
2. Initial Conditions and/or Boundary Conditions are not analytically satisfied
   - Increases the computational cost
   - Can cause gradient pathologies (especially when DNN are used)

Part of the image taken from: Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E., 2019. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*.

X-TFC



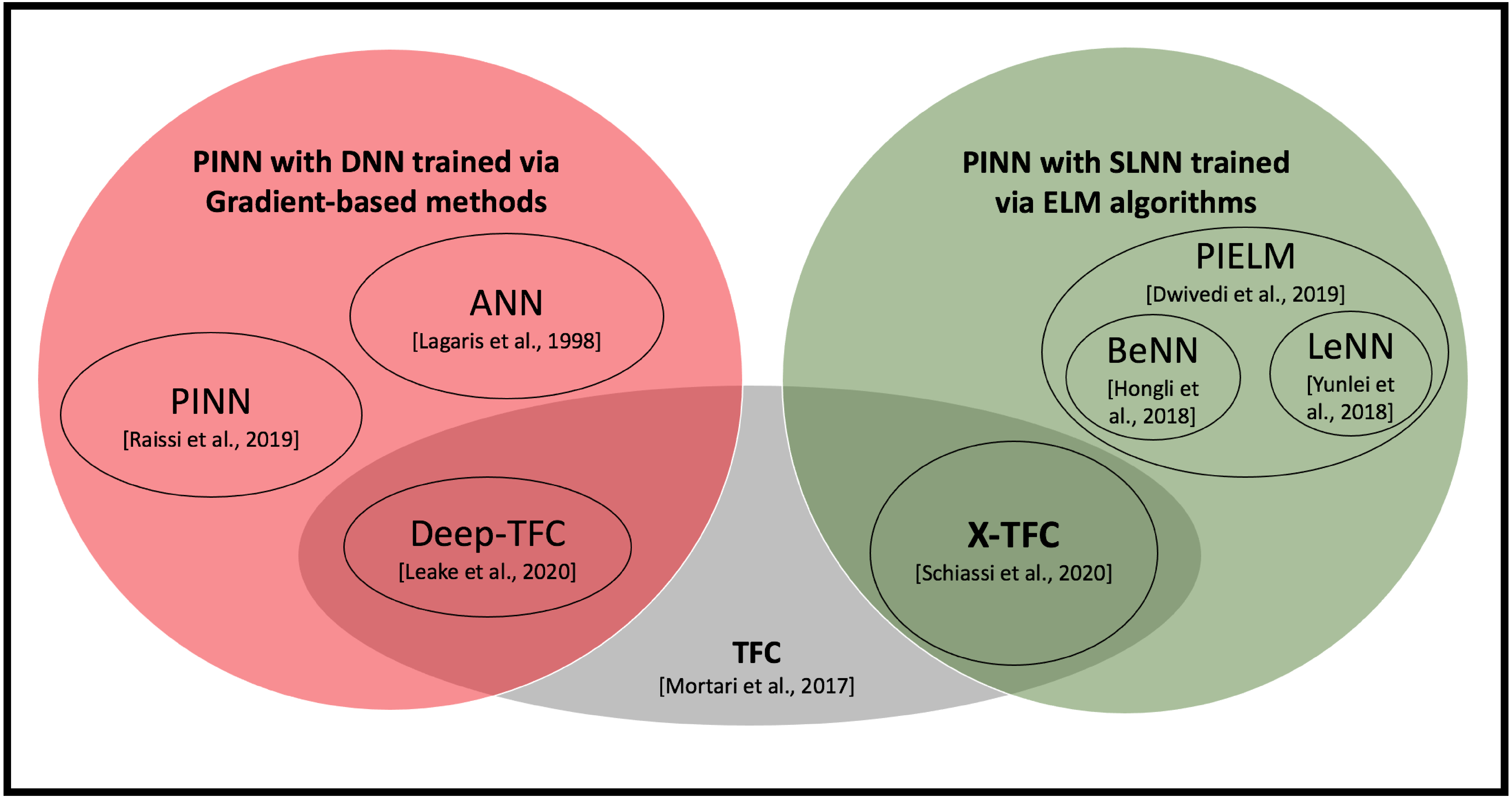
Main Limitations

1. ~~Gradient based methods used to minimize the loss~~
   - ~~Computationally expensive~~
2. ~~Initial Conditions and/or Boundary Conditions are not analytically satisfied~~
   - ~~Increases the computational cost~~
   - ~~Can cause gradient pathologies (especially when DNN are used)~~

Part of the image taken from: Lu, L., Meng, X., Mao, Z. and Karniadakis, G.E., 2019. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*.

Summary of the PINN Frameworks

Space Systems Engineering Laboratory



PINN-based approach to solving generic OCPs

Space Systems Engineering Laboratory

Optimal Control Problem

$$J = \underbrace{\Phi(x(t_0), t_0, x(t_f), t_f)}_{\text{Meyer Cost}} + \underbrace{\int_{t_0}^{t_f} L(x(t), u(t), t)\, dt}_{\text{Lagrange Cost}}$$

$$\dot{x} = f(x(t), u(t), t)$$

$$\begin{cases} \Phi(x(t_0), t_0) = \Phi_0 \\ \Phi(x(t_f), t_f) = \Phi_f \end{cases}$$

→ Pontryagin Maximum/ Minimum Principle →

TPBVP

Boundary Conditions (BCs)

$$y_j(t_0) = y_{0_j}$$
$$y_j(t_f) = y_{f_j}$$
$$\dot{y}_j(t_0) = \dot{y}_{j0}$$
$$\dot{y}_j(t_f) = \dot{y}_{f_j}$$

$$F_i\left(t, y_j(t), \dot{y}_j(t), \ddot{y}_j(t)\right) = 0$$

→ **Solve via PINN**

Hypersensitive Problem

Space Systems Engineering Laboratory

OCP

$$\min \quad \mathcal{J} = \frac{1}{2}\int_0^1 (x^2 + u^2)\, dt$$

subject to

$$\dot{x} = \frac{dx}{dt} = -x + u$$
$$0 \leq t \leq 1$$
$$x(0) = 1.5$$
$$x(1) = 1$$

PMP →

TPBVP

$$\begin{cases} \dot{x} = \dfrac{\partial H}{\partial \lambda} = -x - \lambda \\ \dot{\lambda} = -\dfrac{\partial H}{\partial x} = \lambda - x \end{cases}$$

$$\begin{cases} x(0) = 1.5 \\ x(1) = 1 \end{cases}$$

→ **Solve via PINN**

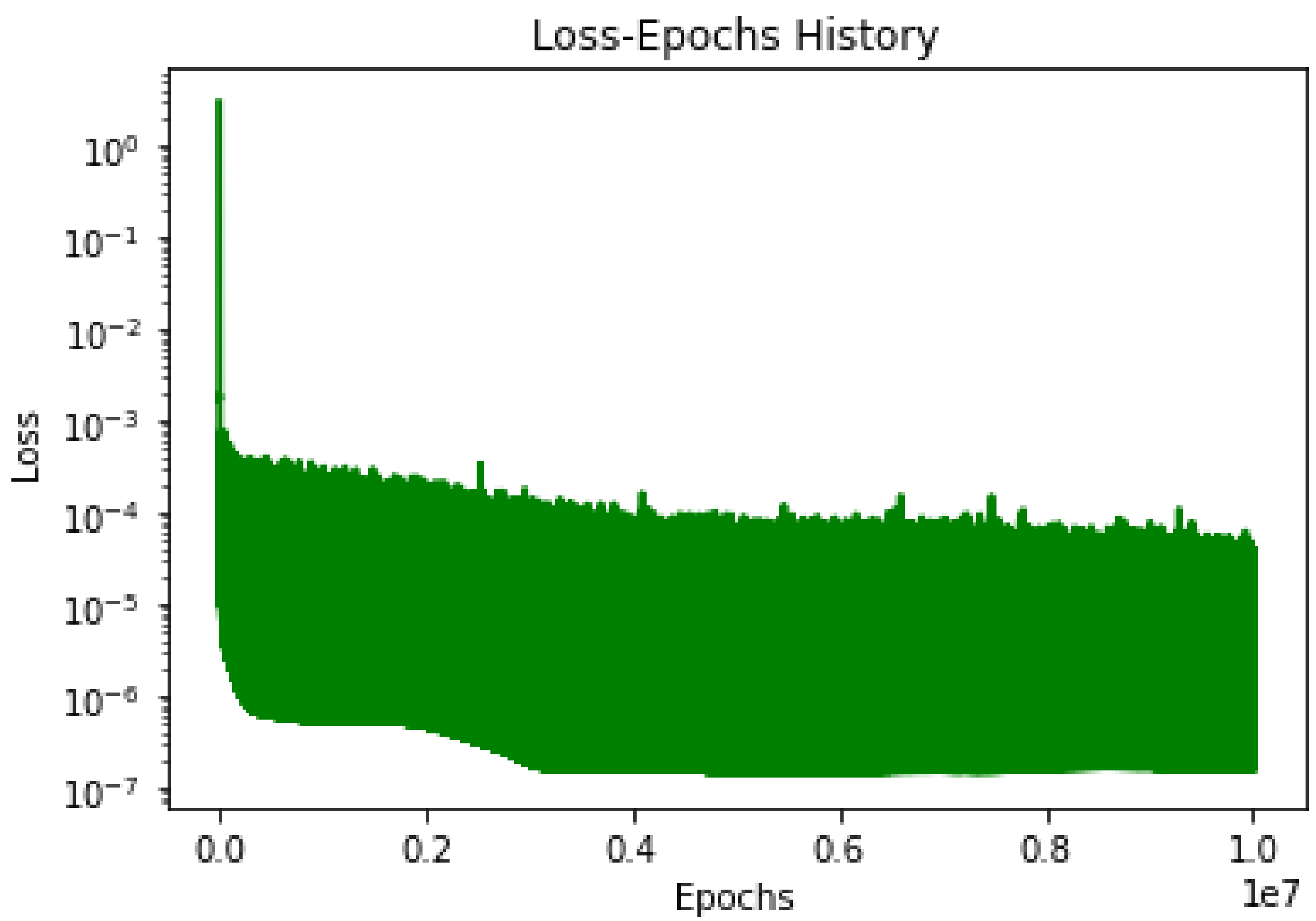
Hypersensitive Problem (cont'd)

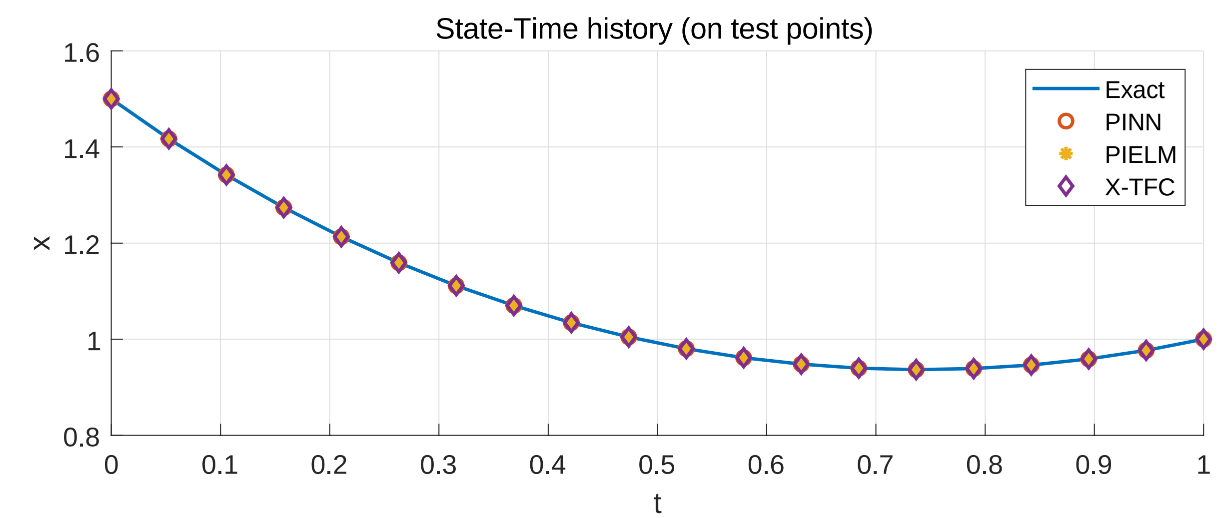
Space Systems Engineering Laboratory

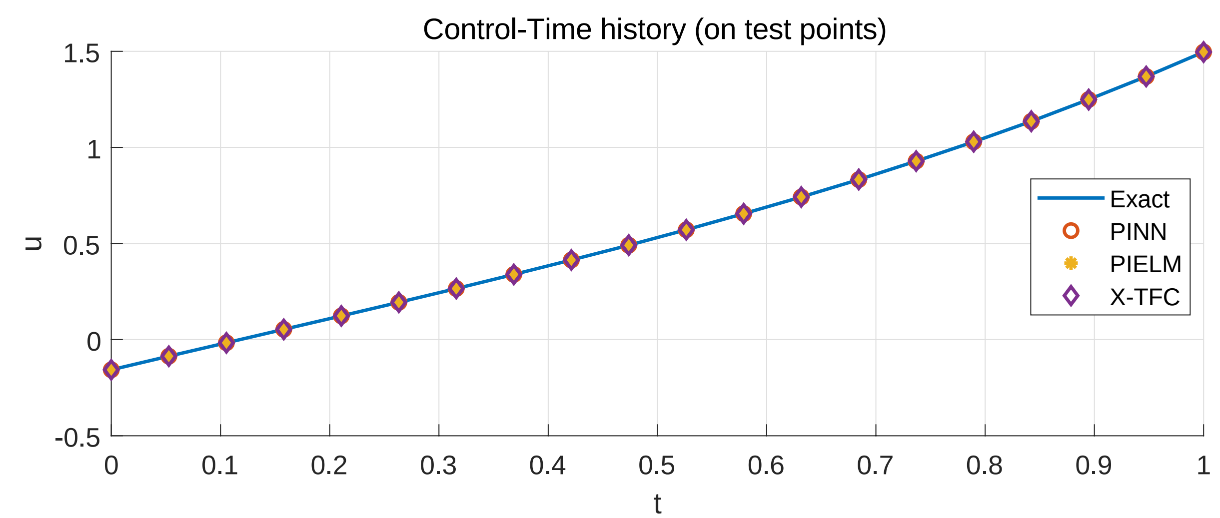
method	layers	neurons	epochs	training time [s]	s_{max}	$\bar{s}$	u_{max}	$\bar{u}$
PINN	1	100	1000	10.43	$7.63 \cdot 10^{-2}$	$2.94 \cdot 10^{-2}$	$2.64 \cdot 10^{-1}$	$1.35 \cdot 10^{-1}$
PINN	1	100	10000	93.09	$5.01 \cdot 10^{-4}$	$2.46 \cdot 10^{-4}$	$1.51 \cdot 10^{-3}$	$5.70 \cdot 10^{-4}$
PINN	1	100	100000	752.04	$1.18 \cdot 10^{-4}$	$5.43 \cdot 10^{-5}$	$1.92 \cdot 10^{-4}$	$7.07 \cdot 10^{-5}$
PINN	1	100	1000000	5968.13	$5.40 \cdot 10^{-5}$	$2.35 \cdot 10^{-5}$	$1.19 \cdot 10^{-4}$	$8.05 \cdot 10^{-5}$
PINN	1	100	10000000	59467.86	$1.54 \cdot 10^{-5}$	$5.21 \cdot 10^{-6}$	$3.87 \cdot 10^{-5}$	$1.32 \cdot 10^{-5}$
PINN	4	25	10000000	94878.03	$2.62 \cdot 10^{-6}$	$8.02 \cdot 10^{-7}$	$5.13 \cdot 10^{-6}$	$1.77 \cdot 10^{-6}$
PINN	10	10	10000000	149527.97	$1.04 \cdot 10^{-5}$	$5.59 \cdot 10^{-6}$	$9.16 \cdot 10^{-6}$	$5.44 \cdot 10^{-6}$
PIELM	1	100	1	0.0029	$6.88 \cdot 10^{-15}$	$3.48 \cdot 10^{-15}$	$8.77 \cdot 10^{-15}$	$7.19 \cdot 10^{-15}$
X-TFC	1	100	1	0.0019	$1.11 \cdot 10^{-15}$	$4.17 \cdot 10^{-16}$	$3.55 \cdot 10^{-15}$	$1.36 \cdot 10^{-15}$

Hyperparameters

- 100 equally distributed training points
- Hyperbolic tangent as activation function
- Adam Optimizer with learning rate 0.001 for PINN
- Inputs and bias sampled from *U*(-3;3) for PIELM and X-TFC







Energy Optimal Rendezvous

Space Systems Engineering Laboratory

OCP

$$\underset{\boldsymbol{a}_c}{\text{minimize}} \quad \frac{1}{2}\int_{t_0}^{t_f} \boldsymbol{a}_c^{\mathrm{T}}\boldsymbol{a}_c \;\mathrm{d}t$$

$$\text{subject to} \quad \dot{\boldsymbol{r}} = \boldsymbol{v}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{M}\boldsymbol{r} + \boldsymbol{N}\boldsymbol{v} + \boldsymbol{a}_c$$

$$\boldsymbol{r}(t_0) = \boldsymbol{r}_0,\ \boldsymbol{r}(t_f) = \boldsymbol{r}_f$$

$$\boldsymbol{v}(t_0) = \boldsymbol{v}_0,\ \boldsymbol{v}(t_f) = \boldsymbol{v}_f$$

→ **PMP** →

TPBVP

$$\begin{cases} \dot{\boldsymbol{r}} = \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}_r} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}_v} = \boldsymbol{M}\boldsymbol{r} + \boldsymbol{N}\boldsymbol{v} - \boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_r = -\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{r}} = -\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{v}} = -\boldsymbol{\lambda}_r - \boldsymbol{N}^{\mathrm{T}}\boldsymbol{\lambda}_v \end{cases}$$

$$\begin{cases} \boldsymbol{r}(t_0) = \boldsymbol{r}_0,\ \boldsymbol{r}(t_f) = \boldsymbol{r}_f \\ \boldsymbol{v}(t_0) = \boldsymbol{v}_0,\ \boldsymbol{v}(t_f) = \boldsymbol{v}_f \end{cases}$$

→ **Solve via PINN**

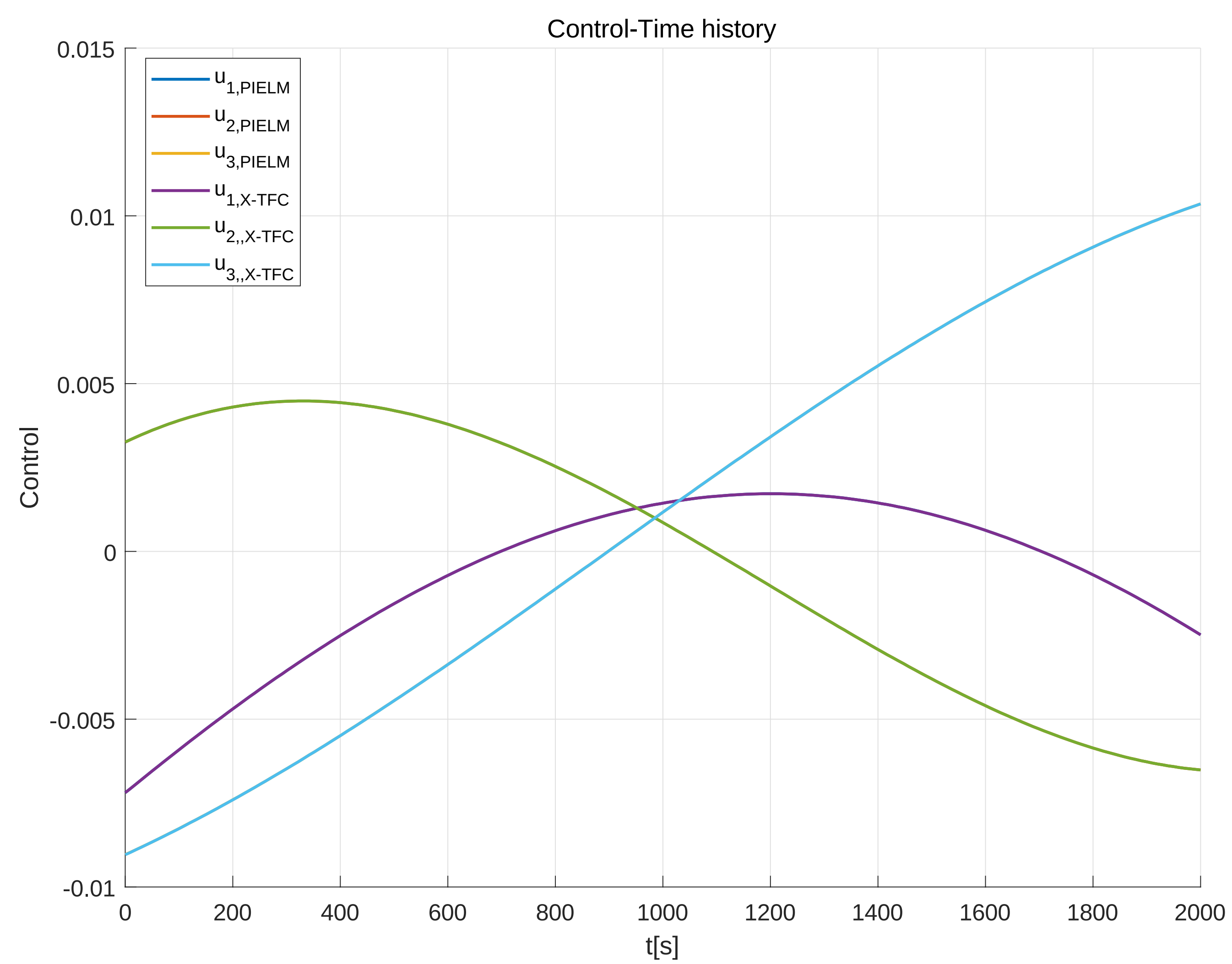
Energy Optimal Rendezvous (cont'd)

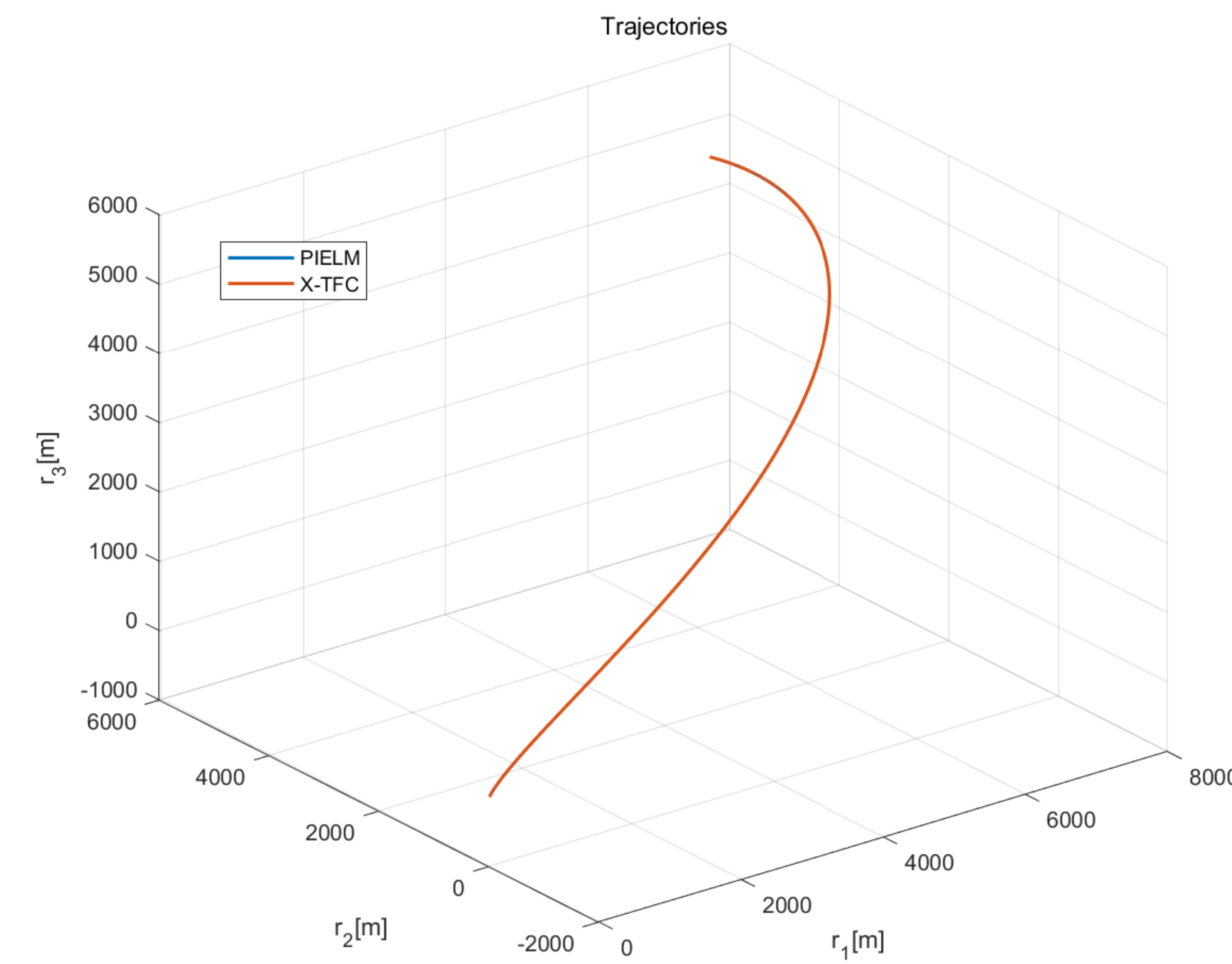
Space Systems Engineering Laboratory

method	training/computational time [s]	$\mathcal{H}(t_f)$	$mean(\mathcal{H})$	$std(\mathcal{H})$	$\mathcal{J}$
LQR [25]	0.1335	$1.5\mathbf{379} \cdot 10^{-6}$	$1.5\mathbf{379} \cdot 10^{-6}$	-	$5.\mathbf{7731} \cdot 10^{-2}$
GPOPS-II [25]	3.4343	$1.5519 \cdot 10^{-6}$	$1.5519 \cdot 10^{-6}$	-	$5.83\mathbf{25} \cdot 10^{-2}$
PIELM	0.0047	$1.551\mathbf{8} \cdot 10^{-6}$	$1.551\mathbf{8} \cdot 10^{-6}$	$4.8415 \cdot 10^{-11}$	$5.38\mathbf{06} \cdot 10^{-2}$
X-TFC	0.0031	$1.5519 \cdot 10^{-6}$	$1.5519 \cdot 10^{-6}$	$2.3052 \cdot 10^{-11}$	$5.8314 \cdot 10^{-2}$

Hyperparameters

- 20 equally distributed training points
- 80 neurons
- Hyperbolic tangent as activation function
- Inputs and bias sampled from *U*(-3;3)





Conclusions and Outlooks

Space Systems Engineering Laboratory

- We presented a new algorithm based on PINNs for solving general OPCs.
  - The class of OCPs with integral quadratic cost is considered
  - The PINN frameworks are used to solve the TPBVP arising from the application of the PMP.
- The algorithm was tested in designing energy optimal rendezvous trajectories (and more).
  - The CPU time, in order of milliseconds, makes the proposed algorithm suitable for on board applications.
  - The performances are comparable with the state-of-the-art software such as GPOPS II.
- Works are in progress to:
  - Employing the PINN-based algorithms to tackle a wide variety of OPCs (especially OPCs for space guidance, navigation, and control).
  - Consider path constraints of the states and inequality constraints on the control

Thanks for watching =)

Space Systems
Engineering Laboratory

https://ssel.arizona.edu/