An Introduction to the Synergy of Deep Learning and Differential Equations

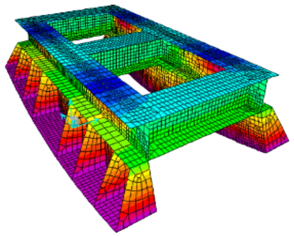
Seminar

Presented by: **Federico Berto**

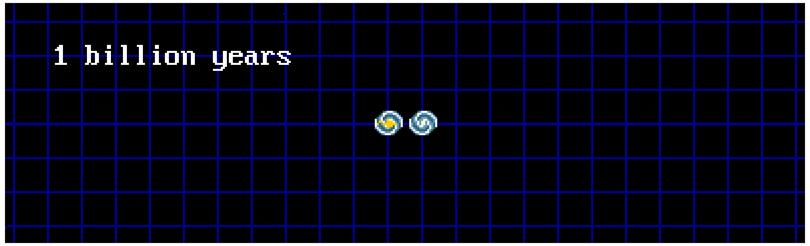
Date: 2021-12-19

Motivation

- Differential equations are the **language of nature**
- Prominent role in many disciplines including engineering, physics, economics, and biology

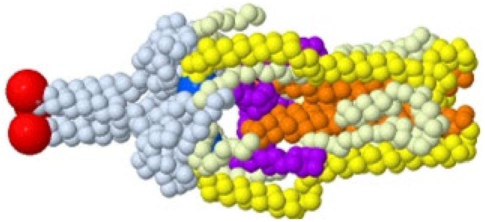


Engineering: Structural analysis with finite element method



Physics: universe expansion

Finance: Black Scholes pricing model



Biology: Spike protein transformation in Covid-19

What is a Differential Equation?

- A function that takes as input the spatial or **spatio-temporal coordinates x** and, optionally, derivatives of the **unknown Φ** with respect to these coordinates

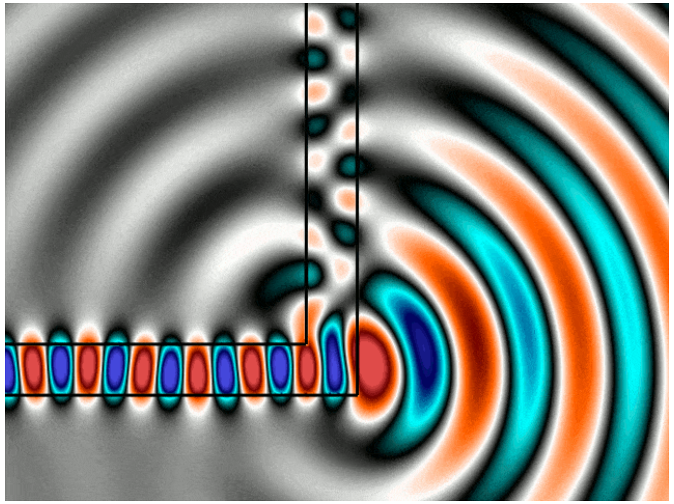
$$F\left(\mathbf{x}, \Phi, \nabla_{\mathbf{x}}\Phi, \nabla^2_{\mathbf{x}}\Phi, \ldots\right) = 0, \quad \Phi : \mathbf{x} \mapsto \Phi(\mathbf{x}).$$

Implicit Formulation of a Differential Equation

- An example from physics

$$\frac{\partial \Phi}{\partial t} - c^2 \frac{\partial \Phi}{\partial \mathbf{x}} = 0.$$

Helmoltz Wave Equation



A More Familiar Example

- Second Newton's law:

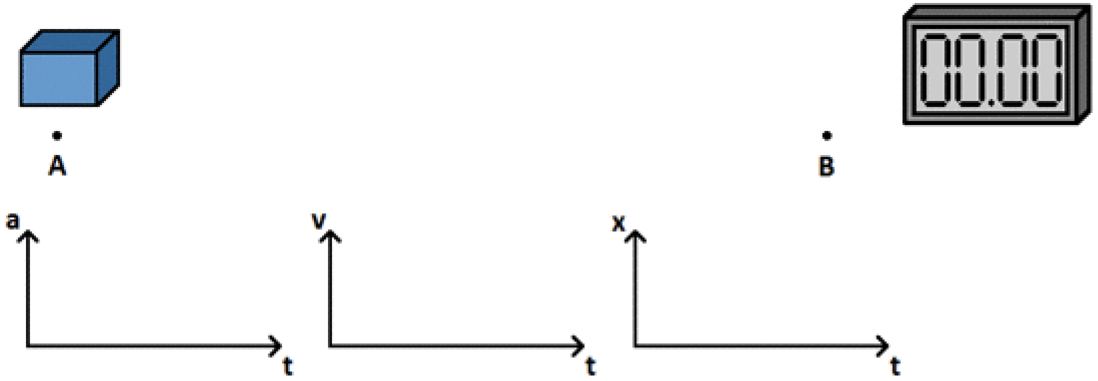
$$F = ma$$

- *Acceleration is the second derivative of the position!*
- So this becomes:

$$F = m\frac{d^2x(t)}{dt^2}$$

In the *implicit (equation=0)* form:

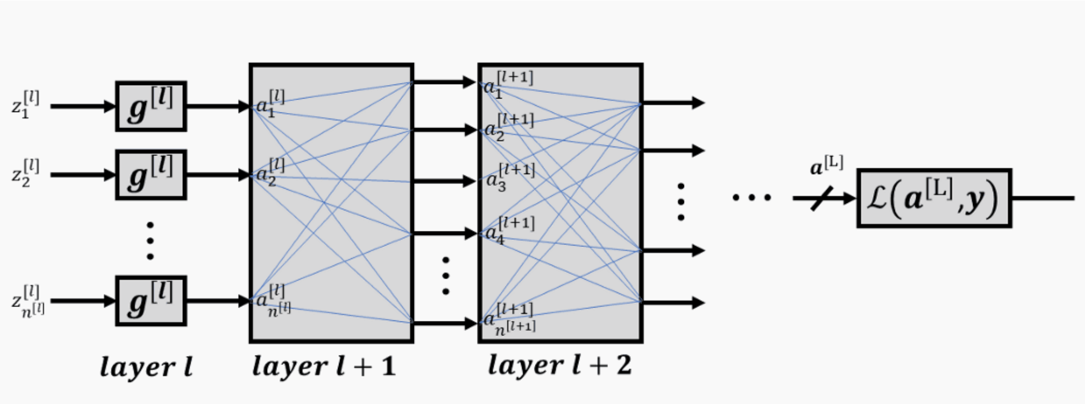
$$F - m\frac{d^2x(t)}{dt^2} = 0$$



Massive object subject to a force

Backpropagation

- Deep Learning is deeply intertwined with Differential Equations with the **backpropagation**

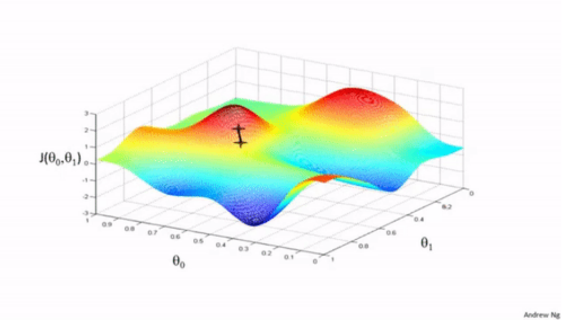


$$\frac{\partial \mathcal{L}}{\partial z^{[l-1]}} = \left[W^{[l]}\right]^{\mathrm{T}} \cdot \frac{\partial \mathcal{L}}{\partial z^{[l]}} * g^{[l-1]\prime}\left(z^{[l-1]}\right)$$

$$\frac{\partial \mathcal{L}}{\partial W^{[l]}} = \frac{\partial \mathcal{L}}{\partial z^{[l]}} \cdot \left[a^{[l-1]}\right]^{\mathrm{T}}$$

$$\frac{\partial \mathcal{L}}{\partial b^{[l]}} = \frac{\partial \mathcal{L}}{\partial z^{[l]}}$$

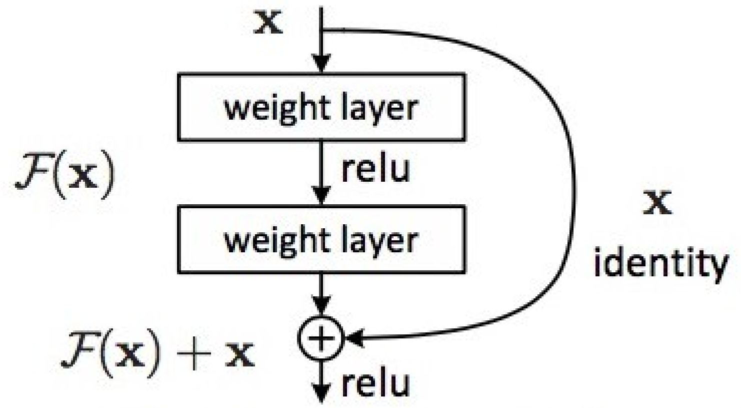
- ➢ We use information of the **gradients** to update a network parameters!



Gradient descent in 2 dimensions

Neural Ordinary Differential Equations

- ResNet [1] have *skip connections* between layers



Residual connection

$$x_{k+1} = F(x_k) + x_k$$

This resembles

$$\frac{x_{k+1} - x_k}{\Delta t} = F(x_k) \approx \frac{dx}{dt} \; for \; \Delta t = 1$$

- ➢ We can build Neural Ordinary Differential Equations [2](Neural ODEs)!
- ➢ Networks tied to numerical solver with inference model:

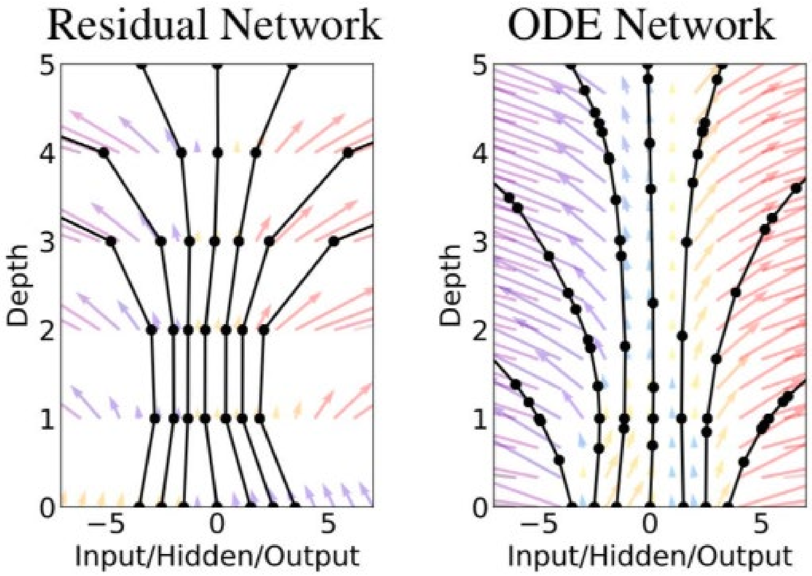
$$x_T = x_0 + \int_0^T \boldsymbol{F_\theta}\big(t, x(t)\big)dt$$

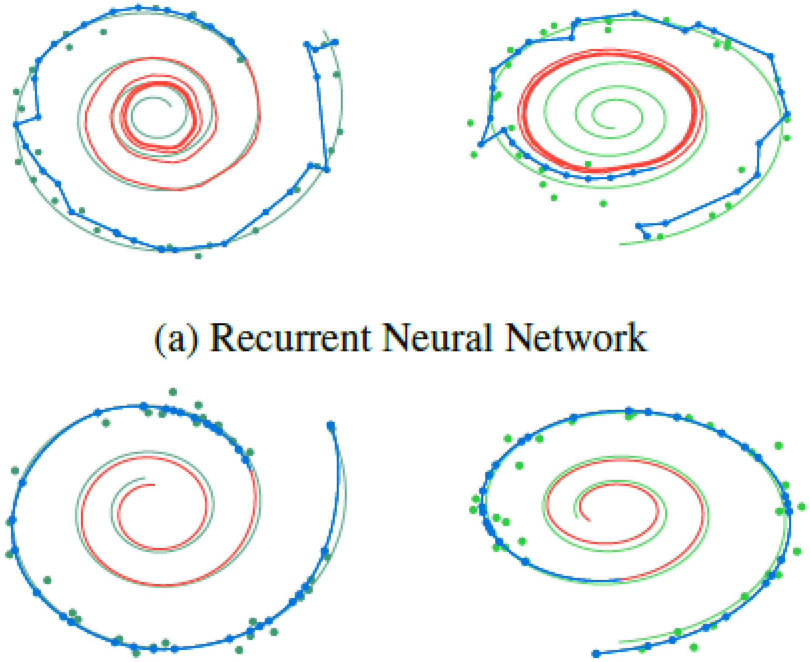
[1] Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.
[2] CHEN, Ricky TQ, et al. Neural ordinary differential equations. NeurIPS 2018

Neural Ordinary Differential Equations

- Networks tied to numerical solver with inference model:

$$x_T = x_0 + \int_0^T \boldsymbol{F}_{\boldsymbol{\theta}}\big(t, x(t)\big)dt$$



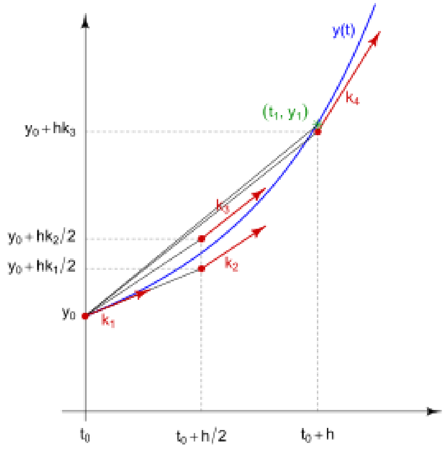


(a) Recurrent Neural Network

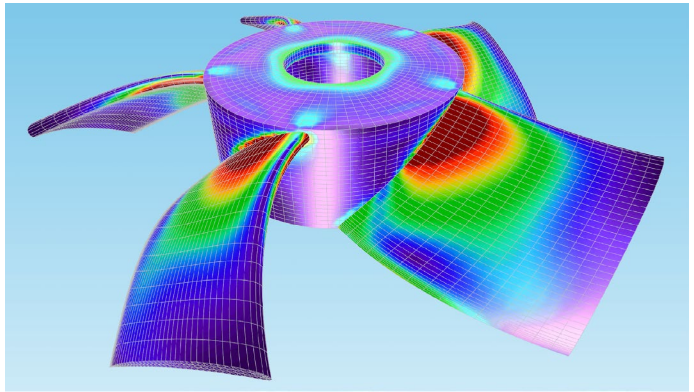
(b) Latent Neural Ordinary Differential Equation

Physics Simulation

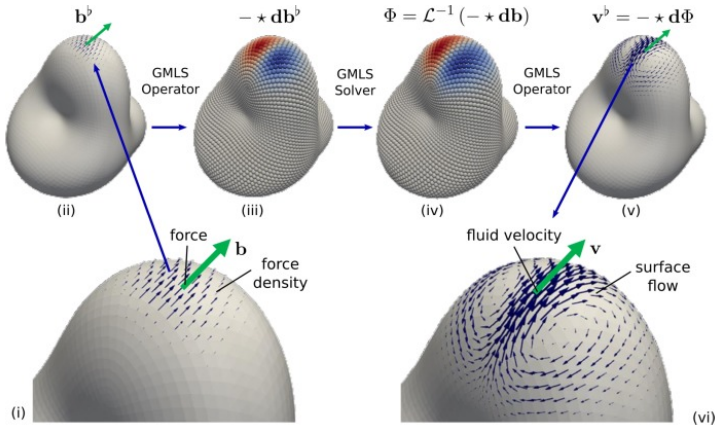
- Classical approaches can be extremely slow or even fail to represent complex behaviors
- Need to choose the right step sizes
- If adaptive solvers are used, it may take too long to solve and simulate a physics process
- Also, **discretizations** are inherently "wrong" → physics are mostly **continuous**!



Runge-Kutta method for ODEs



Finite Element Method (FEM)



Mesh-free methods (e.g. particle)

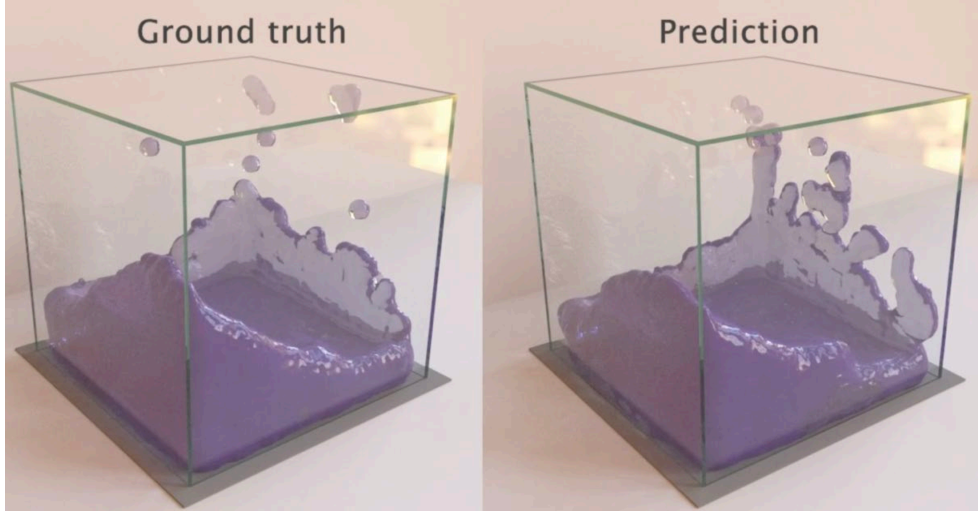
Learning Simulations

Engineered simulators

- Substantial effort to build
- Substantial resources to run
- Only as accurate as the model
- Not always suitable for solving inverse problems

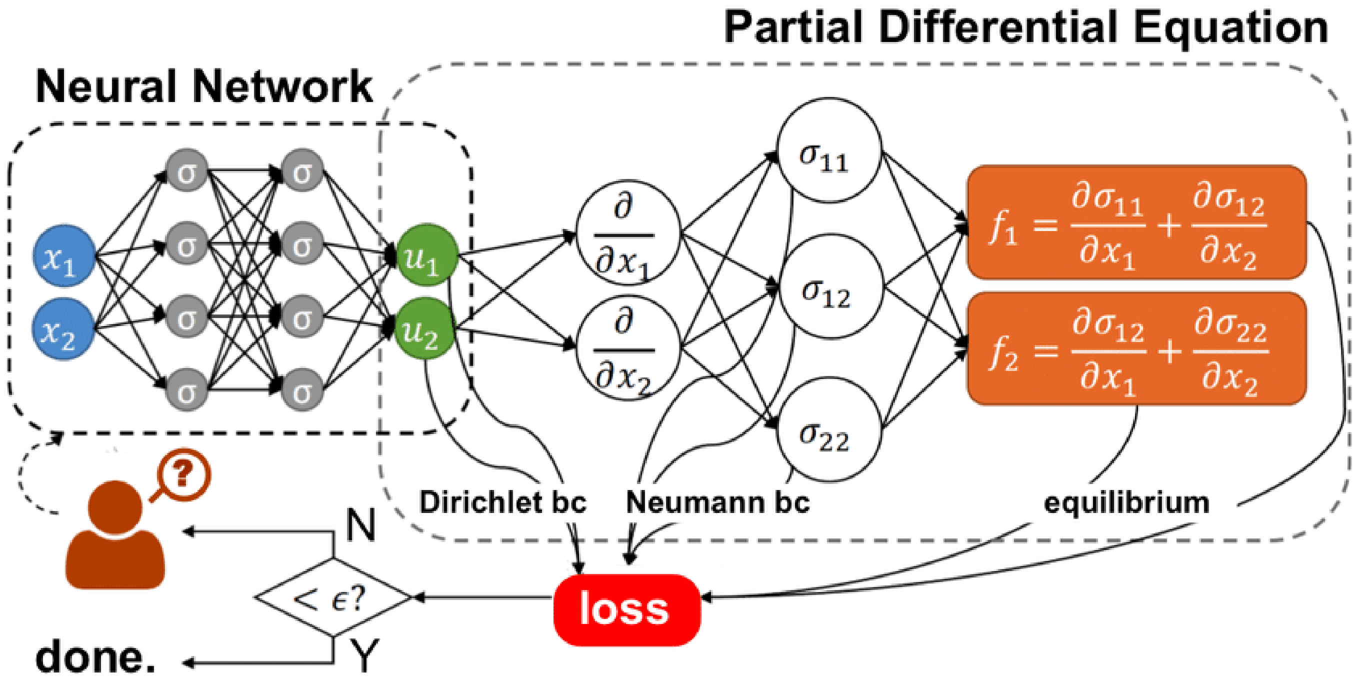
Learned simulators

- Reusable general architectures
- Can be directly optimized for efficiency
- Can be as accurate as the available data
- Gradient-based search for control, inference, etc



Physics Informed Neural Networks

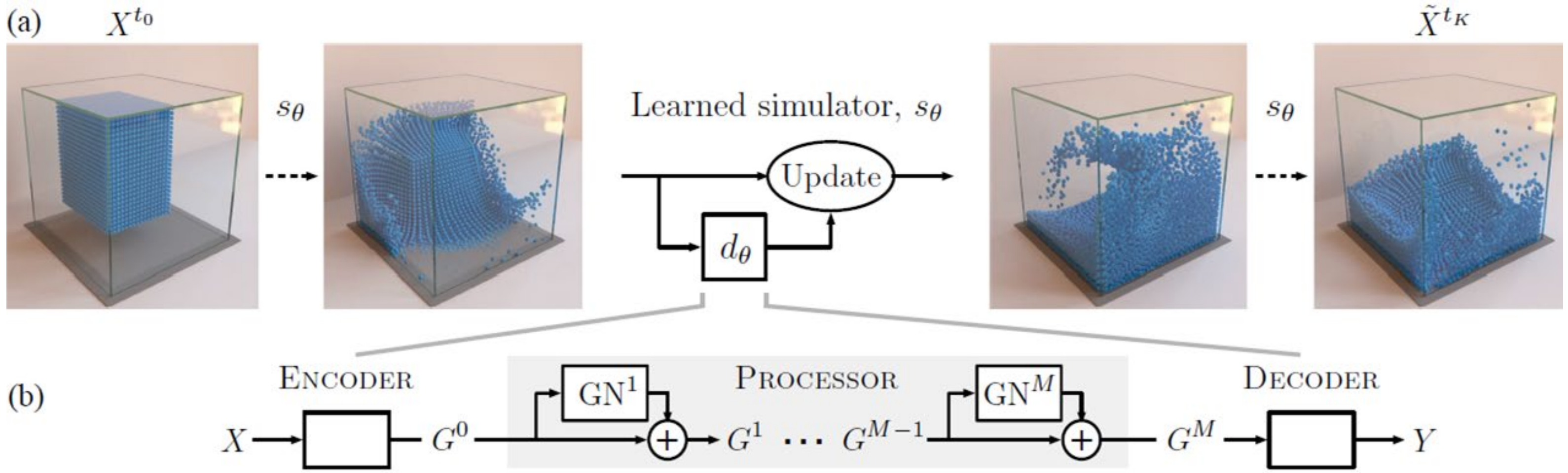
- What if we use Deep Learning to solve Differential Equations?
- ➢ PINNs: Physics Informed Neural Networks[1]



[1] RAISSI, Maziar; PERDIKARIS, Paris; KARNIADAKIS, George E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 2019, 378: 686-707.

Graph Networks Simulators

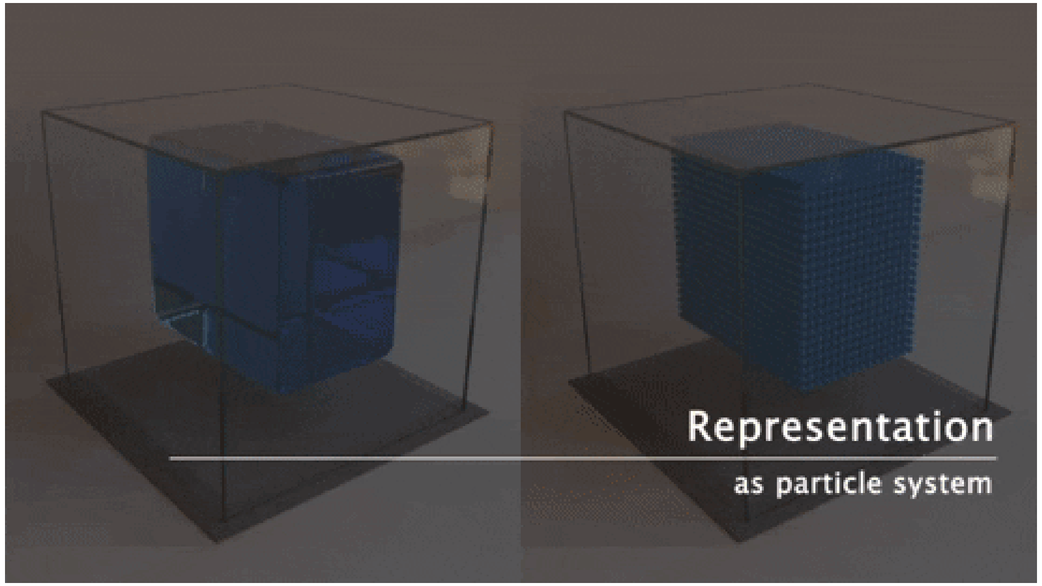
- Many problems can can be discretized in **particles** or **mesh points**
- What If we use Graphs Networks to model these interactions?
- ➢ GNS: Graph-based Neural Simulators [1]
- ➢ A graph of neighboring particles is created ar

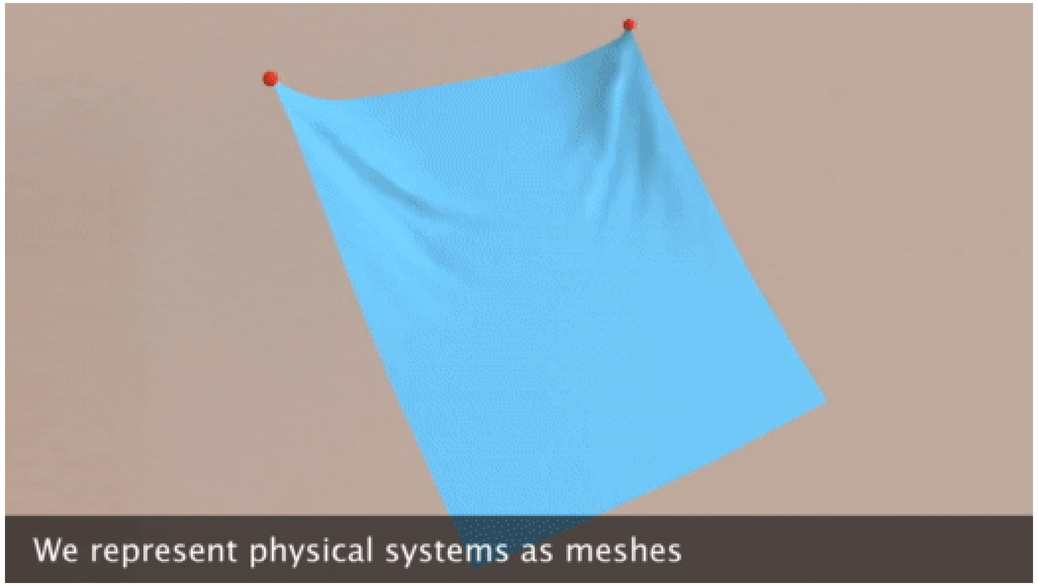


[1] Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." *International Conference on Machine Learning*. PMLR, 2020

Graph Networks Simulators

- GNS: Graph-based Neural Simulators [1]
- We can also expand to mesh-based models [2]





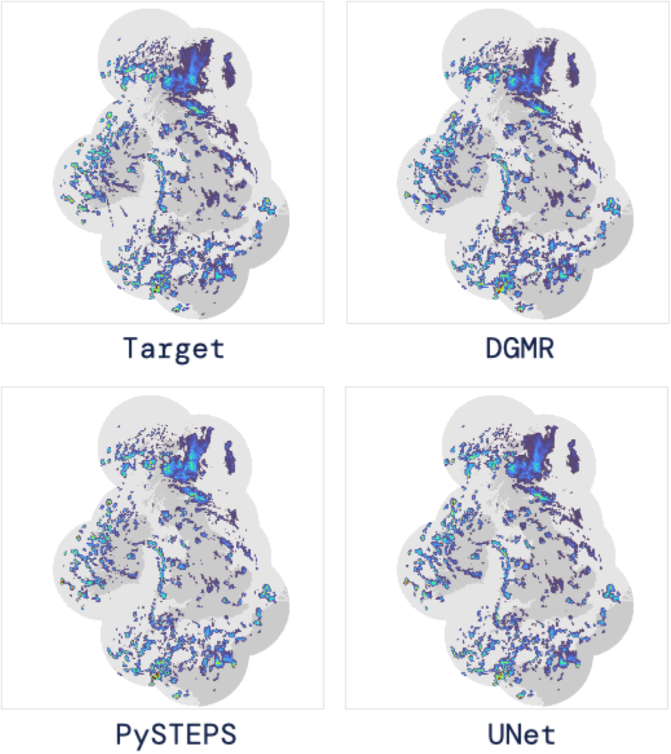
[1] Sanchez-Gonzalez, Alvaro, et al. "Learning to simulate complex physics with graph networks." *International Conference on Machine Learning*. PMLR, 2020
[2] Pfaff, Tobias, et al. "Learning mesh-based simulation with graph networks." *International Conference on Learning Representations.* ICLR, 2021

Latest Advances: Blazing Fast Weather Simulation

- Deepmind has created a weather model which is thousand of times faster than numerical simulations [1]
- Yet, it can forecast weather almost perfectly!

$$P(\mathrm{X}_{M+1:M+N} \mid \mathrm{X}_{1:M}) \\ = \int P(\mathrm{X}_{M+1:M+N} \mid \mathrm{Z}, \mathrm{X}_{1:M}, \theta) P(\mathrm{Z} \mid \mathrm{X}_{1:M}) d\mathrm{Z}.$$

Radar model in differential form

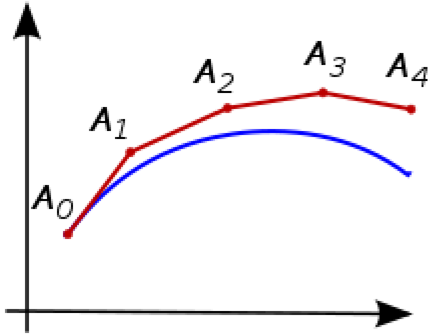


[1] RAVURI, Suman, et al. Skillful Precipitation Nowcasting using Deep Generative Models of Radar. arXiv preprint arXiv:2104.00954, 2021.

.

Hypersolvers: Best of Both Worlds?

- When solving Differential Equations numerically, we need to take **integration steps**
- For each integration step, we have a **discretization error**



- **Hypersolvers:** we can learn to **correct this error** [1]

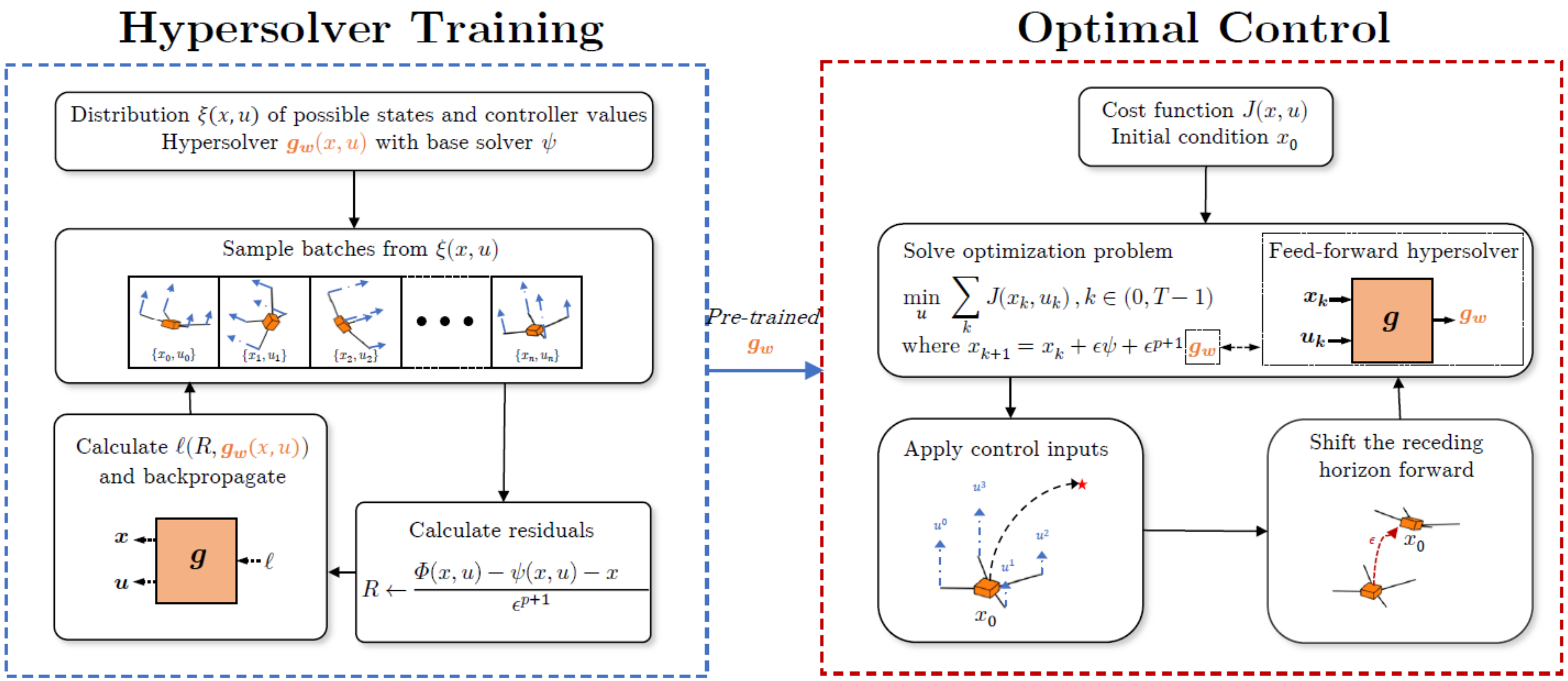
$$x_{k+1} = x_k + \underbrace{\epsilon \psi_\epsilon (t_k, x_k, u_k)}_{\text{base solver step}} + \boxed{\epsilon^{p+1} \underbrace{g_\omega (t_k, x_k, u_k)}_{\text{approximator}}}$$

- The error has *theoretical bounds,* which is very important in practice

[1] Poli M, Massaroli S, Yamashita A, Asama H, Park J. Hypersolvers: Toward fast continuous-depth models. NeurIPS 2020

Hypersolvers: an Application to Control

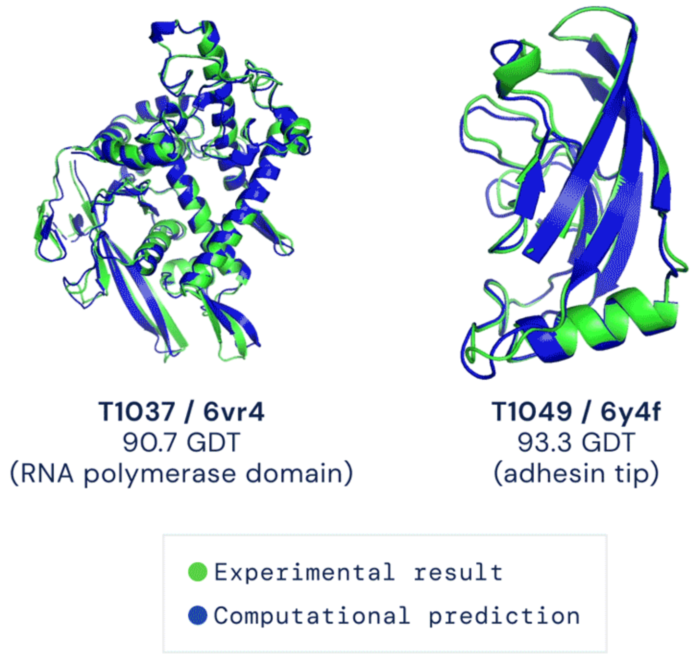
- We can apply the idea to **speed up** optimal control! [1]



[1] Berto, Federico, et al. Neural Solvers for Fast and Accurate Numerical Optimal Control. NeurIPS DLDE Workshop, 2021

Conclusion

- Differential equations describe the reality around us
- Differential Equations and Deep Learning are deeply intertwined
- We can use Deep Learning to simulate and learn the world we live in!



A Final Quote

"Differential equations won't help you much in the design of airplanes — not yet, anyhow."

Nevil Shute, British Aeronautical Engineer, 1923

After a 100 years, the time has come to design airplanes and so much more!

DAEWOONG

Thanks for your attention

Do not hesitate to contact us for any doubts or ideas!

Federico Berto

fberto@kaist.ac.kr