# Structure-preserving machine learning for dissipative systems: methods and applications

**SciML Workshop CWI**

**December 8th**


Quercus Hernández

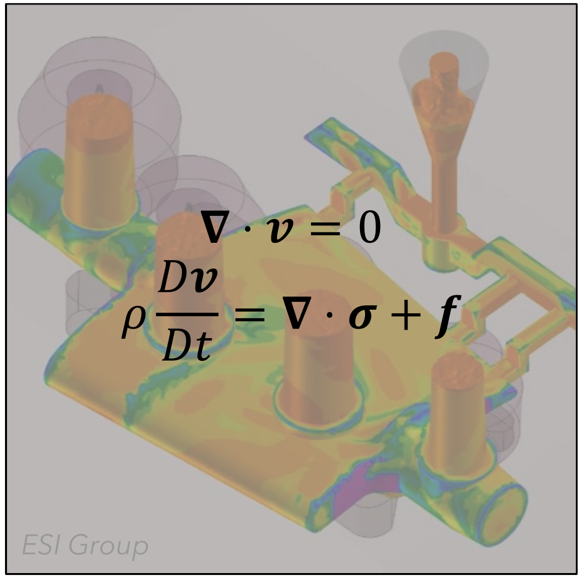
Universidad Zaragoza

# Part I

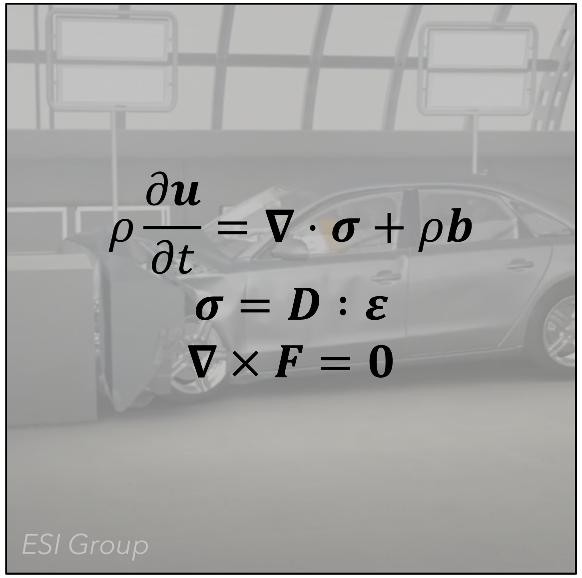
## Introduction and background

# Motivation

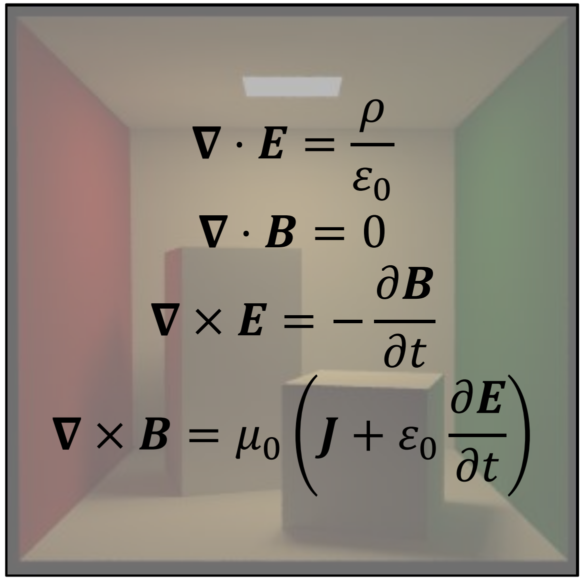
- Computer simulations → **ODEs / PDEs**




Fluid Mechanics




Solid Mechanics




Electromagnetism

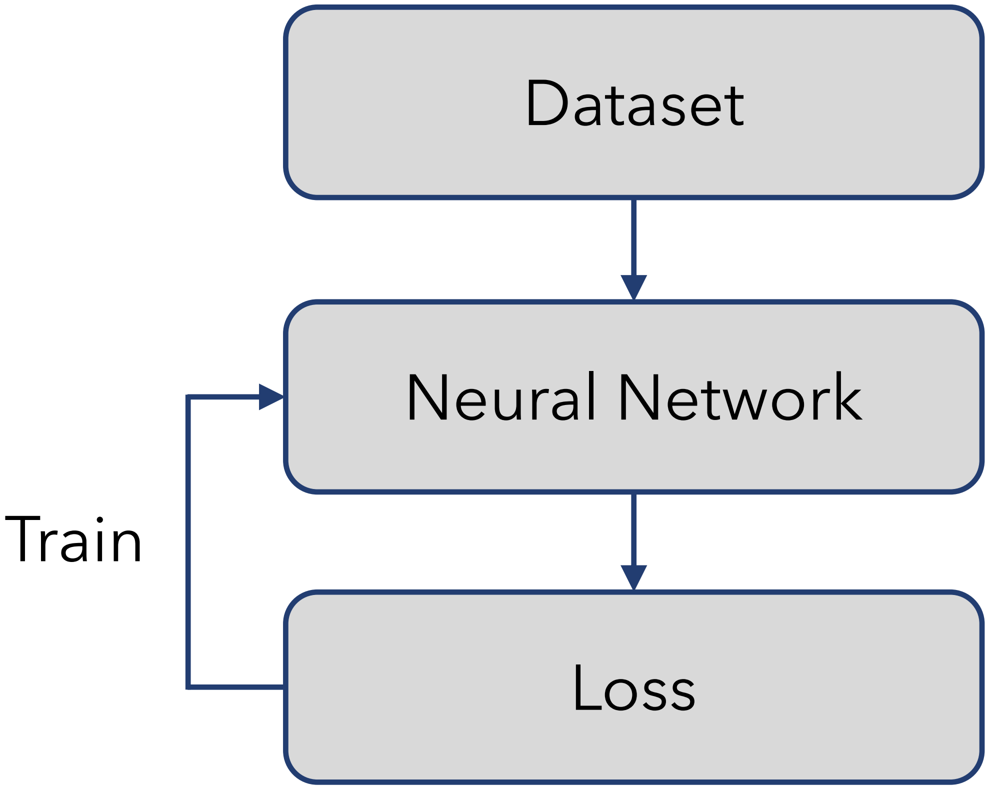
# Motivation

- Numerical methods

| Traditional Methods | Scientific Deep Learning |
| --- | --- |
| Finite Element | Neural Networks |
| Finite Differences | Diffusion Models |
| Finite Volume | Geometric Deep Learning |
| Reduced Order Models | Transformers |
| Mesh-free | |

# Motivation

- Scientific Deep Learning




**Observational bias**

**Inductive bias**

NeuralODE [Chen, 2018]
FNO [Li, 2020]
MP-GNN [Gilmer, 2017]
MeshGraphNet [Pfaff, 2020]

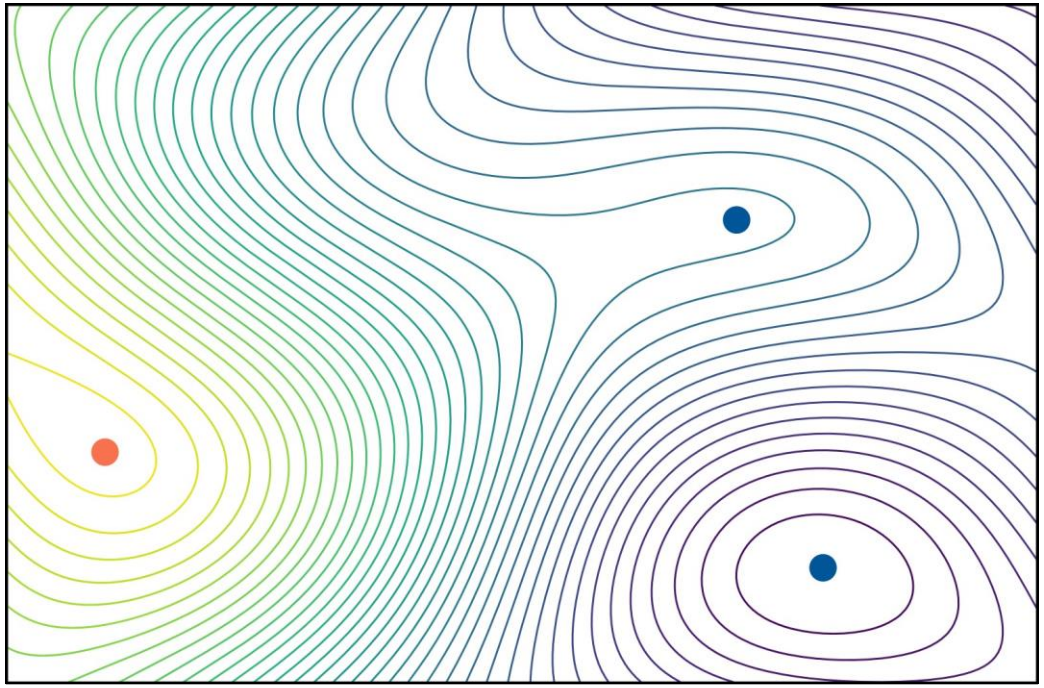
**Learning bias**

PINNs [Raissi, 2019]
DeepONet [Lu, 2019]

# Motivation

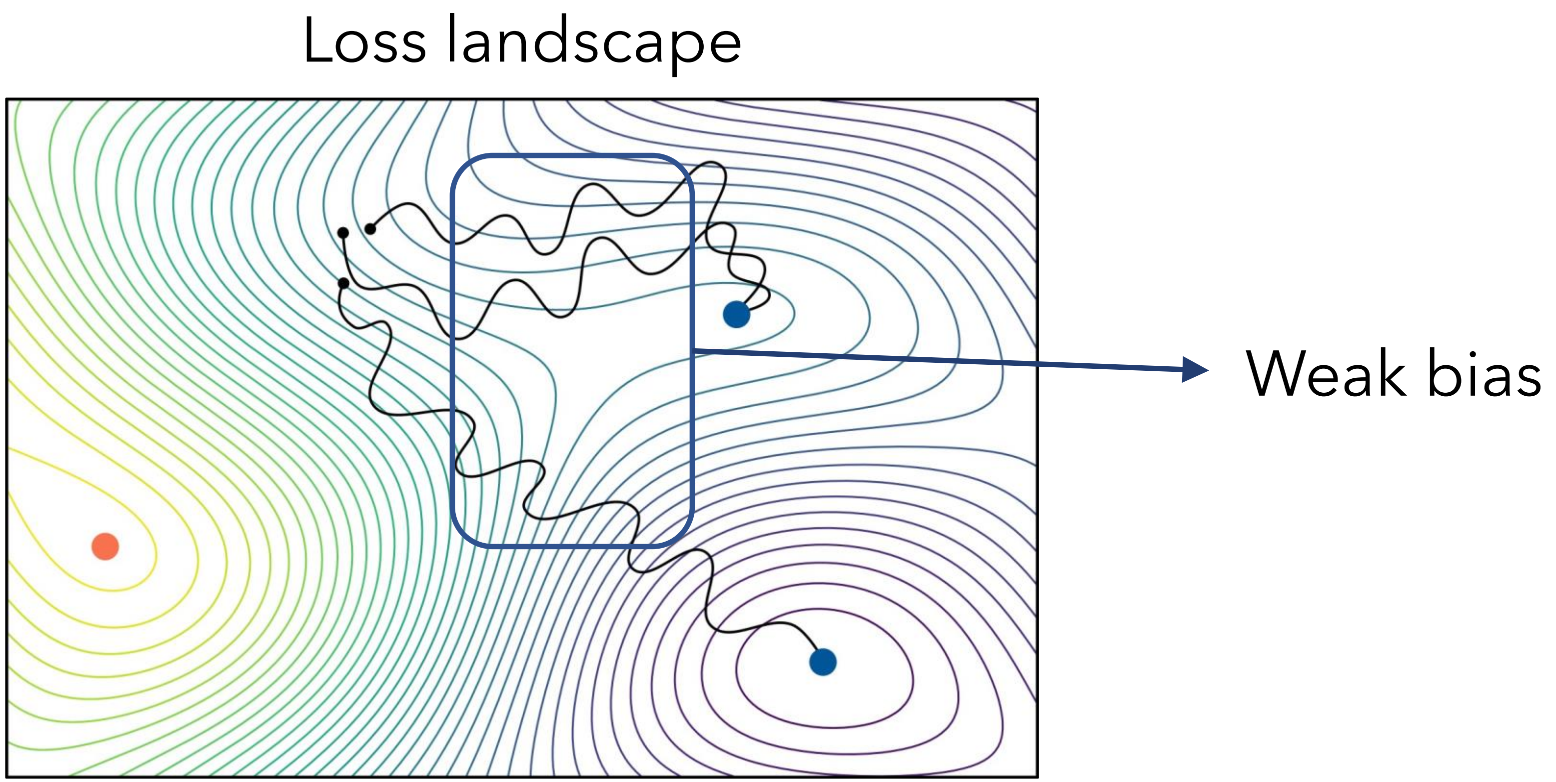
- Biases in deep learning

Loss landscape



# Motivation

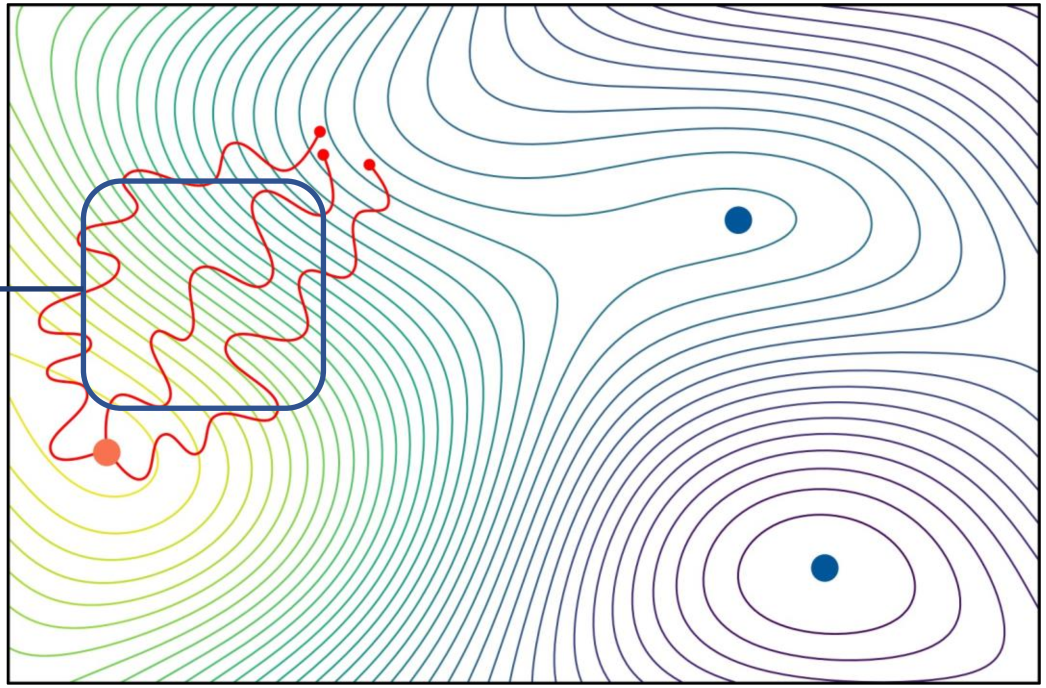
- Biases in deep learning



# Motivation

- Biases in deep learning

Loss landscape



Strong bias

**Physics**

**+**

**Domain**

# Outline

## Part II

Deep Learning of Dynamical Systems


Journal of Computational Physics

Structure-preserving neural networks

Quercus Hernández, Alberto Badías, David González, Francisco Chinesta, Elías Cueto

## Part III

Latent Manifold Learning


ScienceDirect

Computer methods in applied mechanics and engineering

Deep learning of thermodynamics-aware reduced-order models from data

Quercus Hernández, Alberto Badías, David González, Francisco Chinesta, Elías Cueto

## Part IV

Applications to Complex Systems


ScienceDirect

Computer methods in applied mechanics and engineering

Thermodynamics-informed neural networks for physically realistic mixed reality

Quercus Hernández, Alberto Badías, Francisco Chinesta, Elías Cueto

# Part II

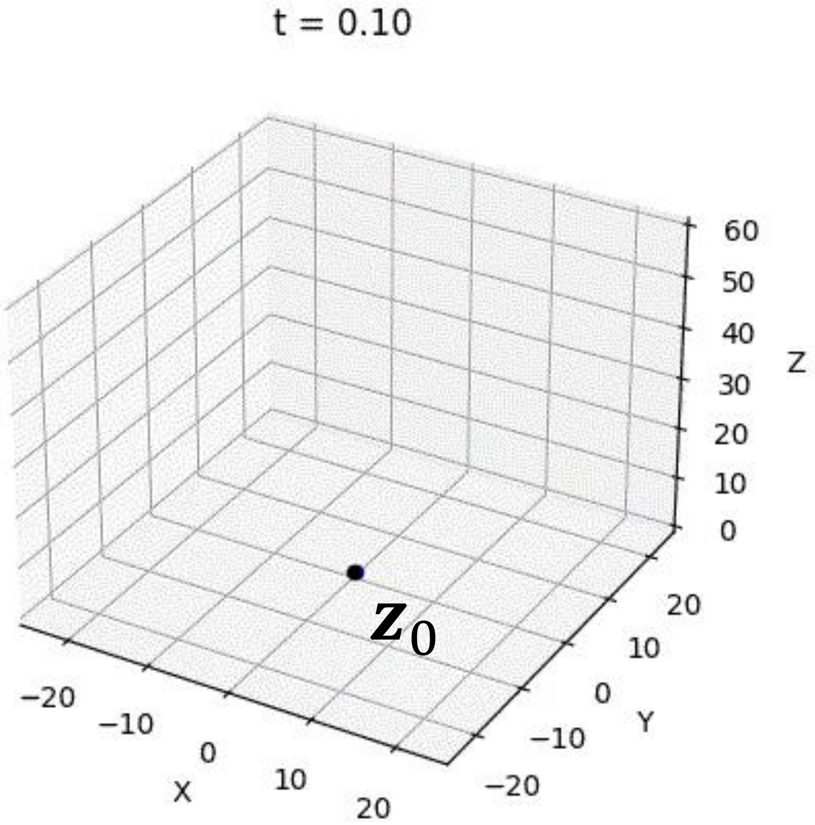
## Deep Learning of Dynamical Systems

# Problem Statement

- Learn a dynamical system from data
- State vector: $\boldsymbol{z} = (z_1, z_2, \dots)$

$$\dot{\boldsymbol{z}} = \frac{d\boldsymbol{z}}{dt} = \boxed{F(\boldsymbol{z}, t)}$$

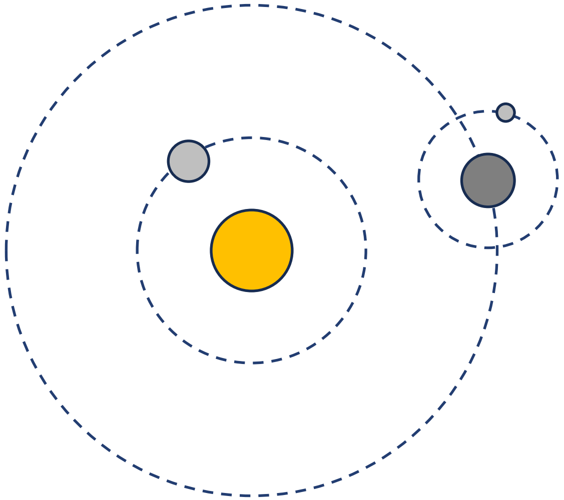
?

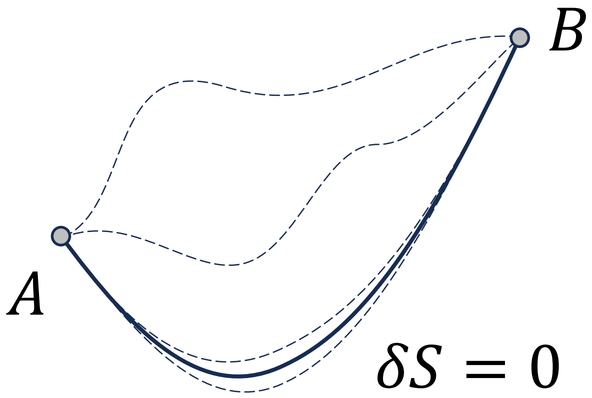


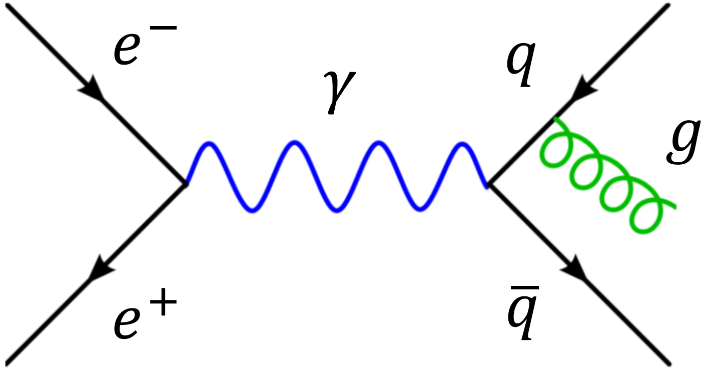

- Time interval: $t \in (0, T]$
- Initial conditions: $\boldsymbol{z}(t=0) = \boldsymbol{z}_0$

# Metriplectic bias

- A brief history of mechanics






**Newton** (1600s) → **Lagrange** (1788) → **Hamilton** (1833)

# Metriplectic bias

- Hamiltonian mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p})$
  - Hamiltonian: $\mathcal{H} = \mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = T(\boldsymbol{p}) + V(\boldsymbol{q})$

Hamilton's equations

$$\begin{cases} \dfrac{d\boldsymbol{p}}{dt} = -\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{q}} & \sim \quad \boldsymbol{F} = m\boldsymbol{a} \\[2ex] \dfrac{d\boldsymbol{q}}{dt} = \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{p}} & \sim \quad \boldsymbol{p} = m\boldsymbol{v} \end{cases}$$

Newtonian mechanics

# Metriplectic bias

- Hamiltonian mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p})$
  - Hamiltonian: $\mathcal{H} = \mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = T(\boldsymbol{p}) + V(\boldsymbol{q})$

Hamilton's equations

$$\begin{cases} \dfrac{d\boldsymbol{p}}{dt} = -\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{q}} \\ \dfrac{d\boldsymbol{q}}{dt} = \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{p}} \end{cases} \longrightarrow \begin{pmatrix} \dfrac{d\boldsymbol{q}}{dt} \\ \dfrac{d\boldsymbol{p}}{dt} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & \boldsymbol{1} \\ -\boldsymbol{1} & 0 \end{pmatrix}}_{\boldsymbol{L}(\boldsymbol{z})} \begin{pmatrix} \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{q}} \\ \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{p}} \end{pmatrix}$$

- Poisson matrix
- Skew-symm

$$\boxed{\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}}}$$

- Symplectic
- Reversible

# Metriplectic bias

- Hamiltonian mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p})$
  - Hamiltonian: $\mathcal{H} = \mathcal{H}(\boldsymbol{q}, \boldsymbol{p}) = T(\boldsymbol{p}) + V(\boldsymbol{q})$

Hamilton's equations

$$\begin{cases} \dfrac{d\boldsymbol{p}}{dt} = -\dfrac{\partial \mathcal{H}}{\partial \boldsymbol{q}} \\[2ex] \dfrac{d\boldsymbol{q}}{dt} = \dfrac{\partial \mathcal{H}}{\partial \boldsymbol{p}} \end{cases}$$

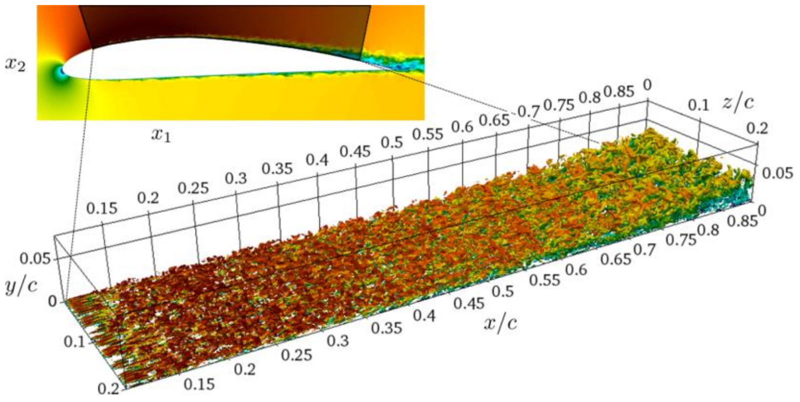
Hamiltonian NN [Sánchez-González, 2019]
SympNets [Jin, 2020]
Lagrangian NN [Bhatoo, 2021]
Poisson NN [Jin, 2023]

$$\boxed{\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}}}$$

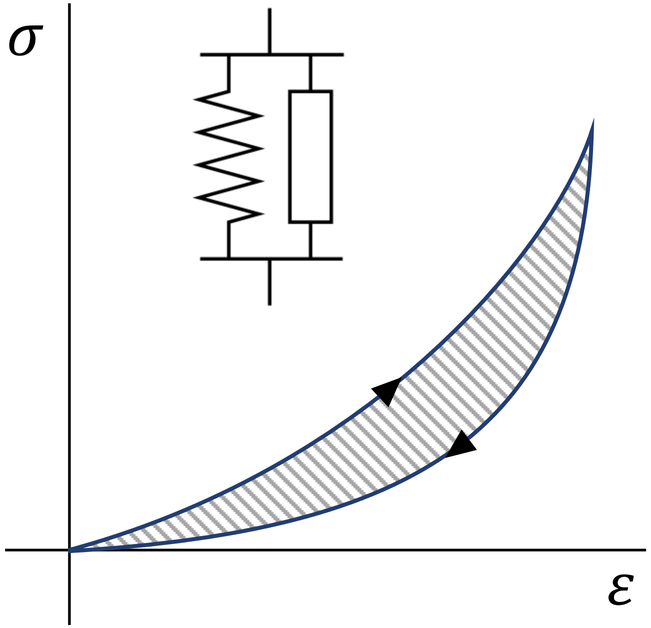
- Symplectic
- Reversible

# Metriplectic bias

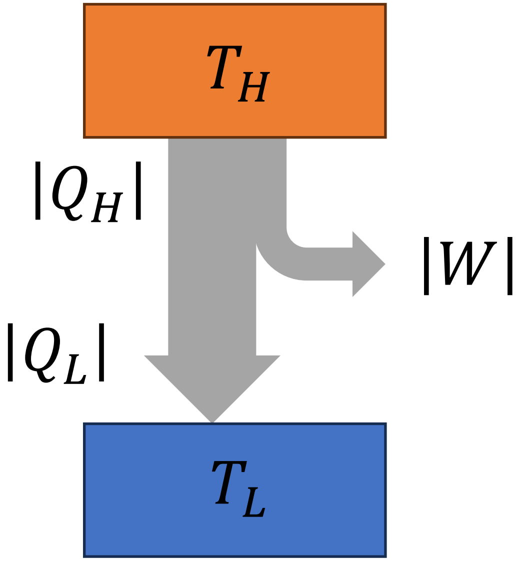
- Dissipative mechanics




Turbulence




Viscoelasticity




Heat transfer

# Metriplectic bias

- Dissipative mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}, \dots)$
  - Energy, Entropy: $E(\boldsymbol{z}),\ S(\boldsymbol{z})$

$$\frac{d\boldsymbol{z}}{dt} = \boxed{\boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}}} + \boxed{\boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}}}$$

- Friction matrix $\boldsymbol{M}(\boldsymbol{z})$
- Metric (Symm + PSD)
- Irreversible

# Metriplectic bias

- Dissipative mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}, \dots)$
  - Energy, Entropy: $E(\boldsymbol{z}),\ S(\boldsymbol{z})$

$$\boxed{\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}}} \longrightarrow$$

**Metriplectic**

**GENERIC**

**G**eneral **E**quation for **N**on-**E**quilibrium **R**eversible-**I**rreversible **C**oupling

[Grmela, 1997]
[Öttinger, 1997]
[Morrison, 1986]

# Metriplectic bias

- Dissipative mechanics
  - State variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}, \dots)$
  - Energy, Entropy: $E(\boldsymbol{z}),\ S(\boldsymbol{z})$

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}} \longrightarrow \begin{array}{l} \boldsymbol{M}\dfrac{\partial E}{\partial \boldsymbol{z}} = \boldsymbol{0} \\[2ex] \boldsymbol{L}\dfrac{\partial S}{\partial \boldsymbol{z}} = \boldsymbol{0} \end{array}$$

Energy conservation: $\dfrac{dE}{dt} = 0$

Entropy inequality: $\dfrac{dS}{dt} \geq 0$

**Thermodynamical guarantees**

# Metriplectic bias

- Why deep learning?
  - Known closed form
    - Navier-Stokes [Morrison, 1984]
    - Rigid bodies [Morrison, 1986]
    - Thermoelasticity [Romero, 2010]

**Solution exists**

# Metriplectic bias

- Why deep learning?
  - Known closed form
    - Navier-Stokes [Morrison, 1984]
    - Rigid bodies [Morrison, 1986]
    - Thermoelasticity [Romero, 2010]
  - Fast surrogate model → **Realtime applications**

# Metriplectic bias

- Why deep learning?
  - Known closed form
    - Navier-Stokes [Morrison, 1984]
    - Rigid bodies [Morrison, 1986]
    - Thermoelasticity [Romero, 2010]
  - Fast surrogate model
  - General inductive bias for scientific discovery [Flashel, 2021]

**Thermodynamics-based**

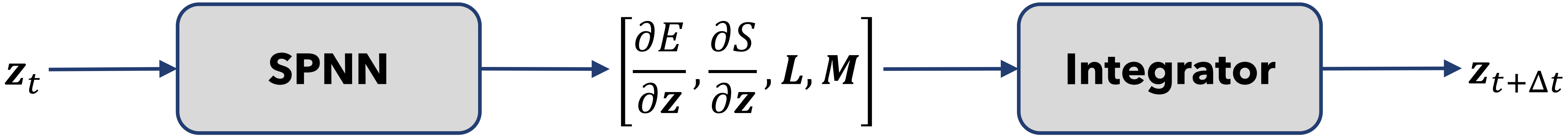
**Equation-free**

# Metriplectic bias

- Why deep learning?
  - Known closed form
    - Navier-Stokes [Morrison, 1984]
    - Rigid bodies [Morrison, 1986]
    - Thermoelasticity [Romero, 2010]
  - Fast surrogate model
  - General inductive bias for scientific discovery [Flashel, 2021]
- Other approaches
  - Port-Hamiltonian [Eidnes, 2022]
  - Symmetry breaking [Wang, 2022]
  - OnsagerNet [Yu, 2021]

# Metriplectic bias

- Structure-Preserving Neural Networks [Hernández, 2021]
  - Learn dynamics from data
  - State Variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}, \dots)$



# Metriplectic bias

- Structure-Preserving Neural Networks [Hernández, 2021]

$\boldsymbol{z}_t \rightarrow$ **SPNN** $\rightarrow \left[\frac{\partial E}{\partial \boldsymbol{z}}, \frac{\partial S}{\partial \boldsymbol{z}}, \boldsymbol{L}, \boldsymbol{M}\right] \rightarrow$ **Integrator** $\rightarrow \boldsymbol{z}_{t+\Delt t}$

- $\boldsymbol{L}$ Skew-symm
- $\boldsymbol{M}$ Symm PSD
- Degeneracies

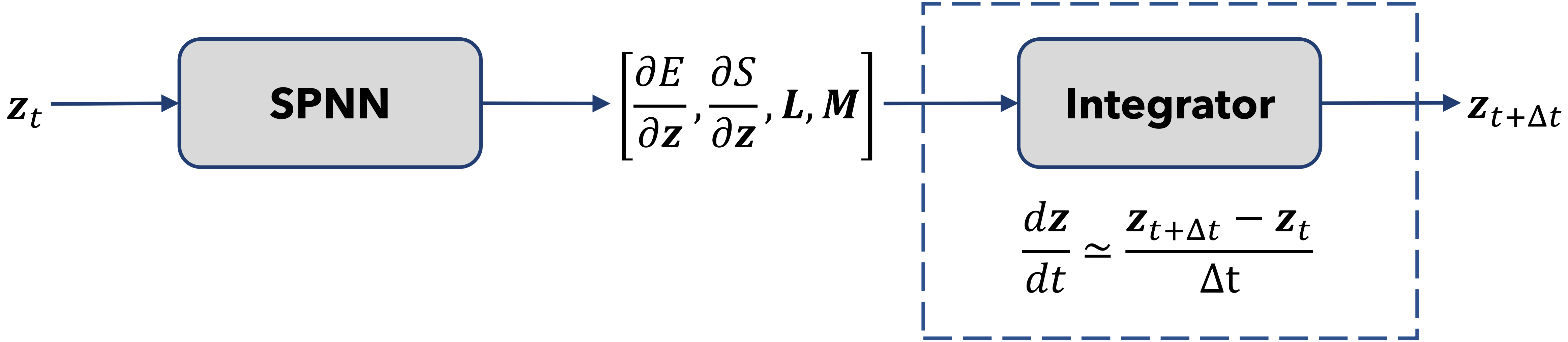
$$\boldsymbol{L}\frac{\partial S}{\partial \boldsymbol{z}} = \boldsymbol{M}\frac{\partial E}{\partial \boldsymbol{z}} = \boldsymbol{0}$$

$$\boldsymbol{L} = \boldsymbol{l} - \boldsymbol{l}^T$$
$$\boldsymbol{M} = \boldsymbol{m}^T\boldsymbol{m}$$

Autograd from
$E$ and $S$

# Metriplectic bias

- Structure-Preserving Neural Networks [Hernández, 2021]



# Metriplectic bias

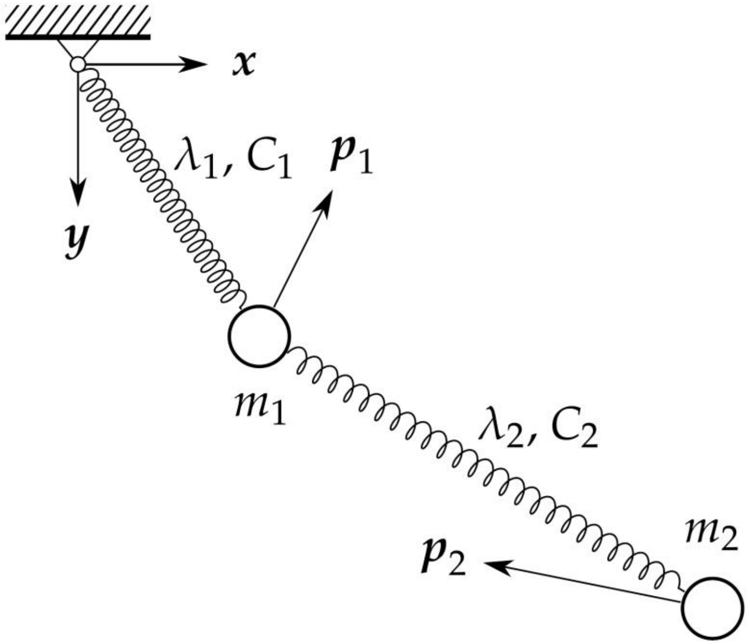
- Structure-Preserving Neural Networks [Hernández, 2021]

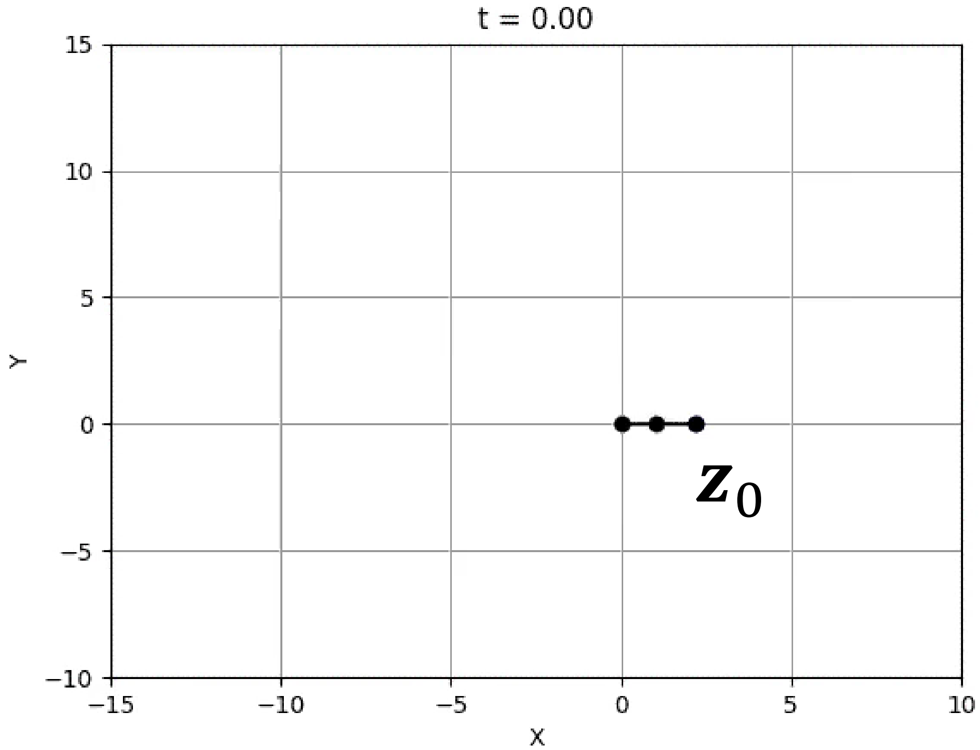
$$\boldsymbol{z}_t \longrightarrow \boxed{\textbf{SPNN}} \longrightarrow \left[\frac{\partial E}{\partial \boldsymbol{z}}, \frac{\partial S}{\partial \boldsymbol{z}}, \boldsymbol{L}, \boldsymbol{M}\right] \longrightarrow \boxed{\textbf{Integrator}} \longrightarrow \boldsymbol{z}_{t+\Delta t}$$

$$\mathcal{L} = \underbrace{\boxed{\lambda_d \left\| \boldsymbol{z}_{t+\Delta t}^{net} - \boldsymbol{z}_{t+\Delta t}^{GT} \right\|_2^2}}_{\text{Data loss}} + \underbrace{\boxed{\left\| \boldsymbol{L} \frac{\partial S}{\partial \boldsymbol{z}} \right\|_2^2 + \left\| \boldsymbol{M} \frac{\partial E}{\partial \boldsymbol{z}} \right\|_2^2}}_{\text{Degeneracy loss}}$$

# Experiments

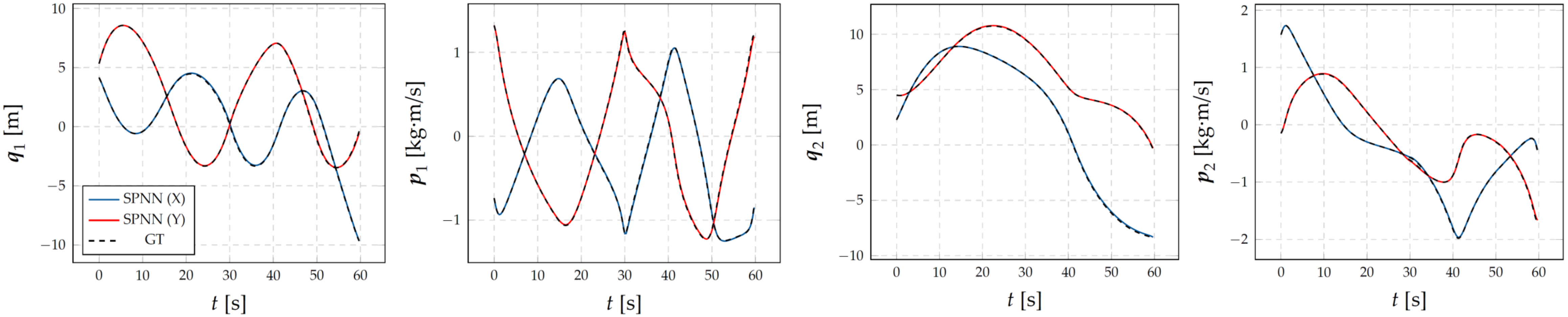
- Double thermoelastic-pendulum
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{p}_1, \boldsymbol{p}_2, s_1, s_2)\}$
  - Dataset: 50 ICs



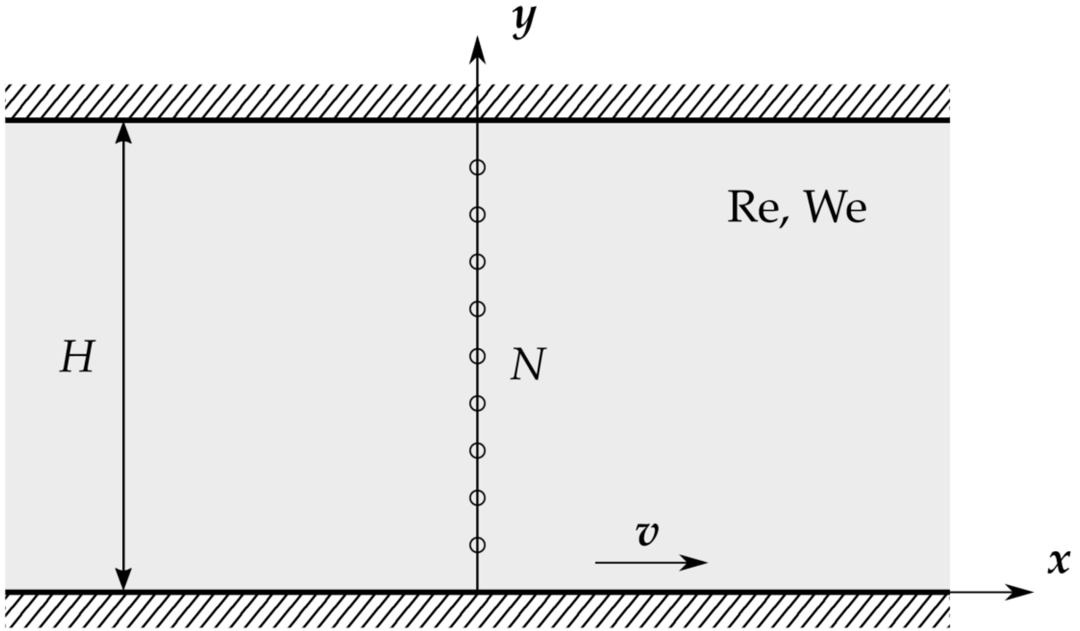
# Experiments

- Double thermoelastic-pendulum
  - Results: Time evolution

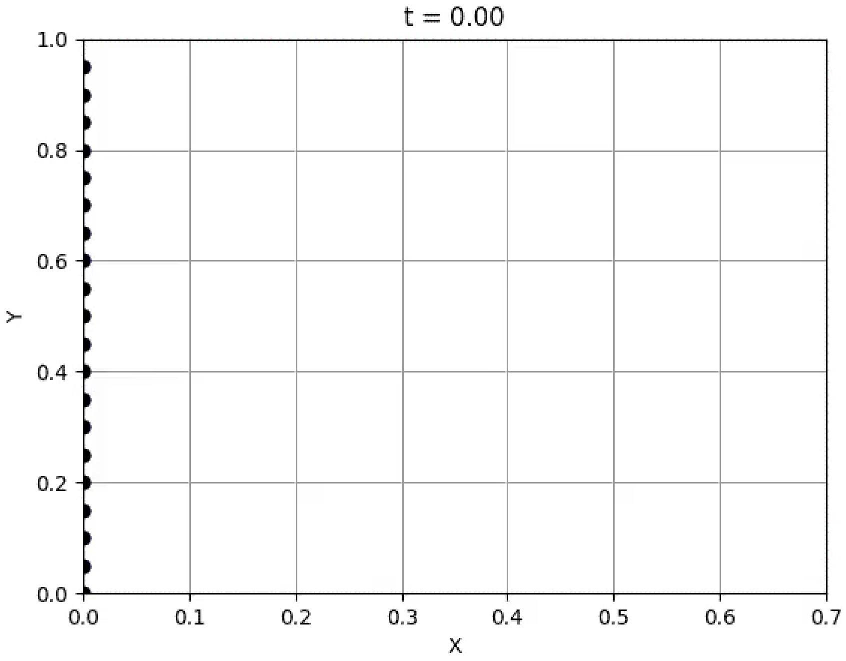


# Experiments

- Couette flow in an Oldroyd-B fluid
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, v, e, \tau)\}$
  - Dataset: 100 ICs

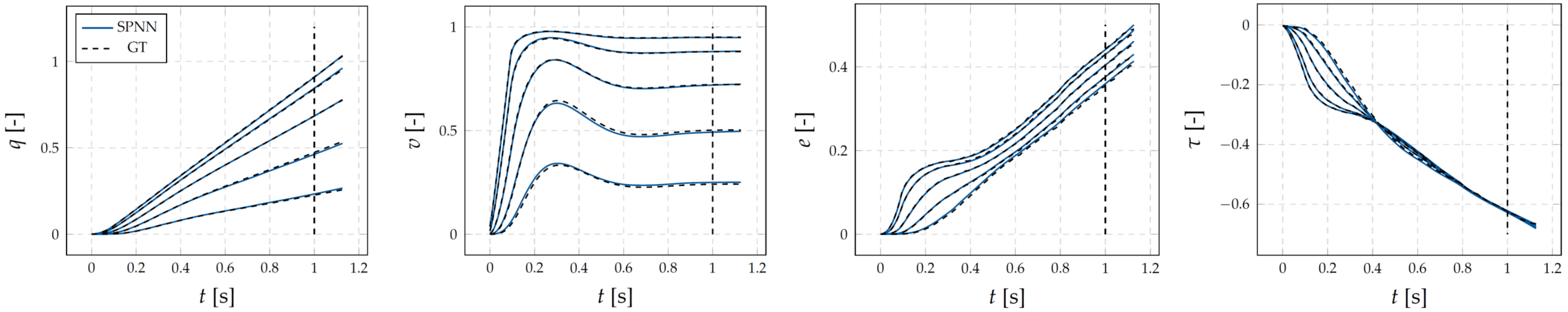



$\boldsymbol{z}_0$



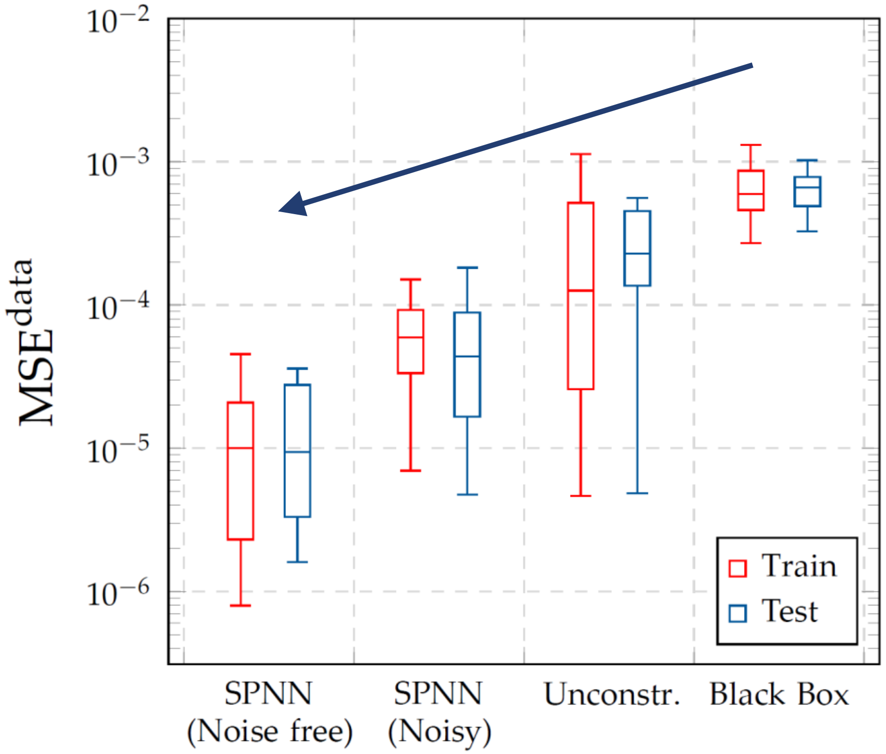
# Experiments

- Couette flow in an Oldroyd-B fluid
  - Results: Time evolution



# Experiments

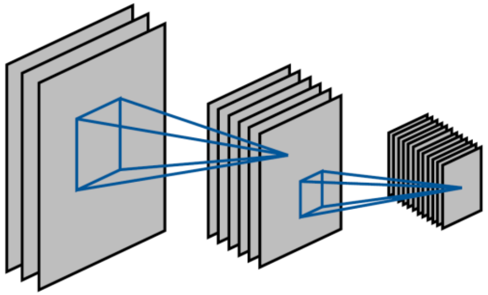
- Couette flow in an Oldroyd-B fluid
  - Results: Ablation study



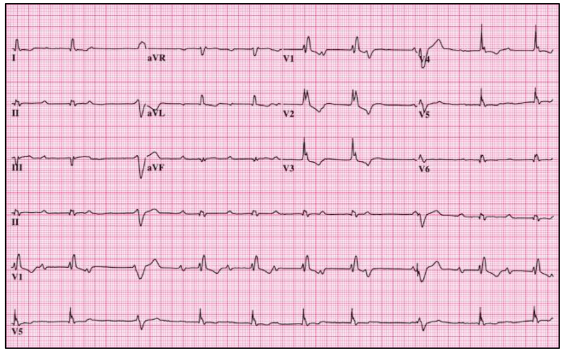

**The more physics, the less error**

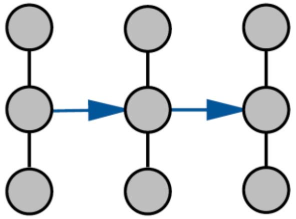
# Geometric bias

- Data are not random

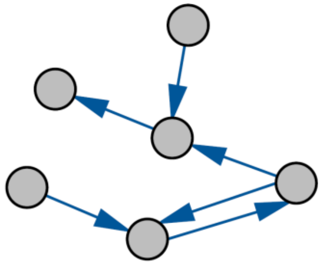


**CNN**
(Locality)





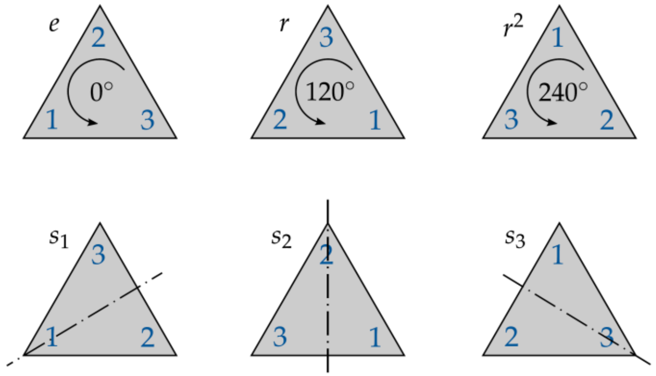
**RNN**
(Sequentiality)



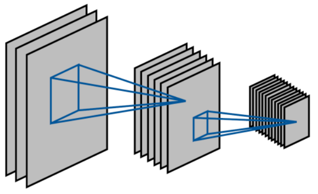
**??**
(Specified)

# Geometric bias

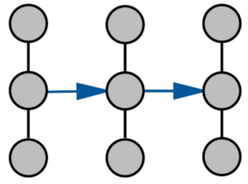
- Graph Neural Networks
  - Geometric Deep Learning [Bronstein, 2021]



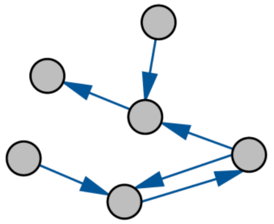

$$G = \{e, r, r^2, s_1, s_2, s_3\}$$



$T(n)$



$W$

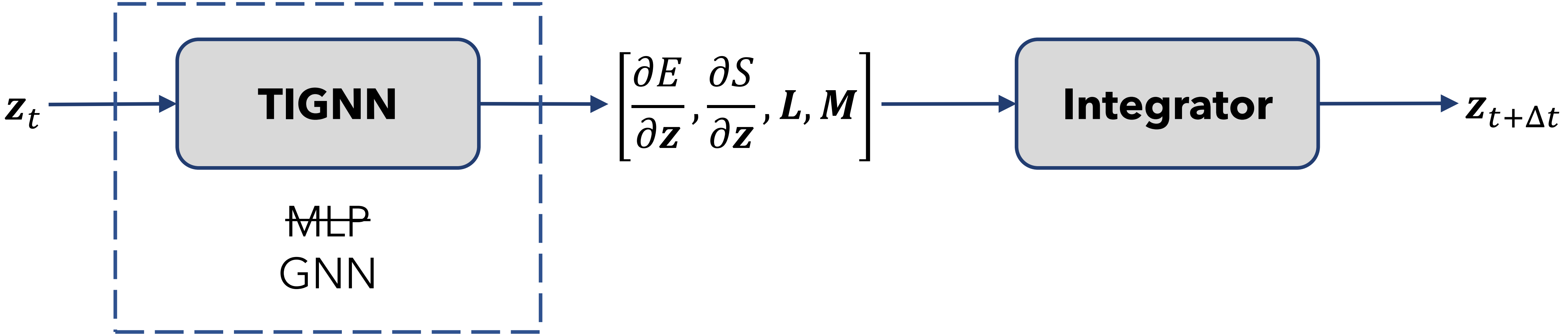


$\Sigma_n$

**Symmetry** → **Group** → **Equivariance / Invariance**

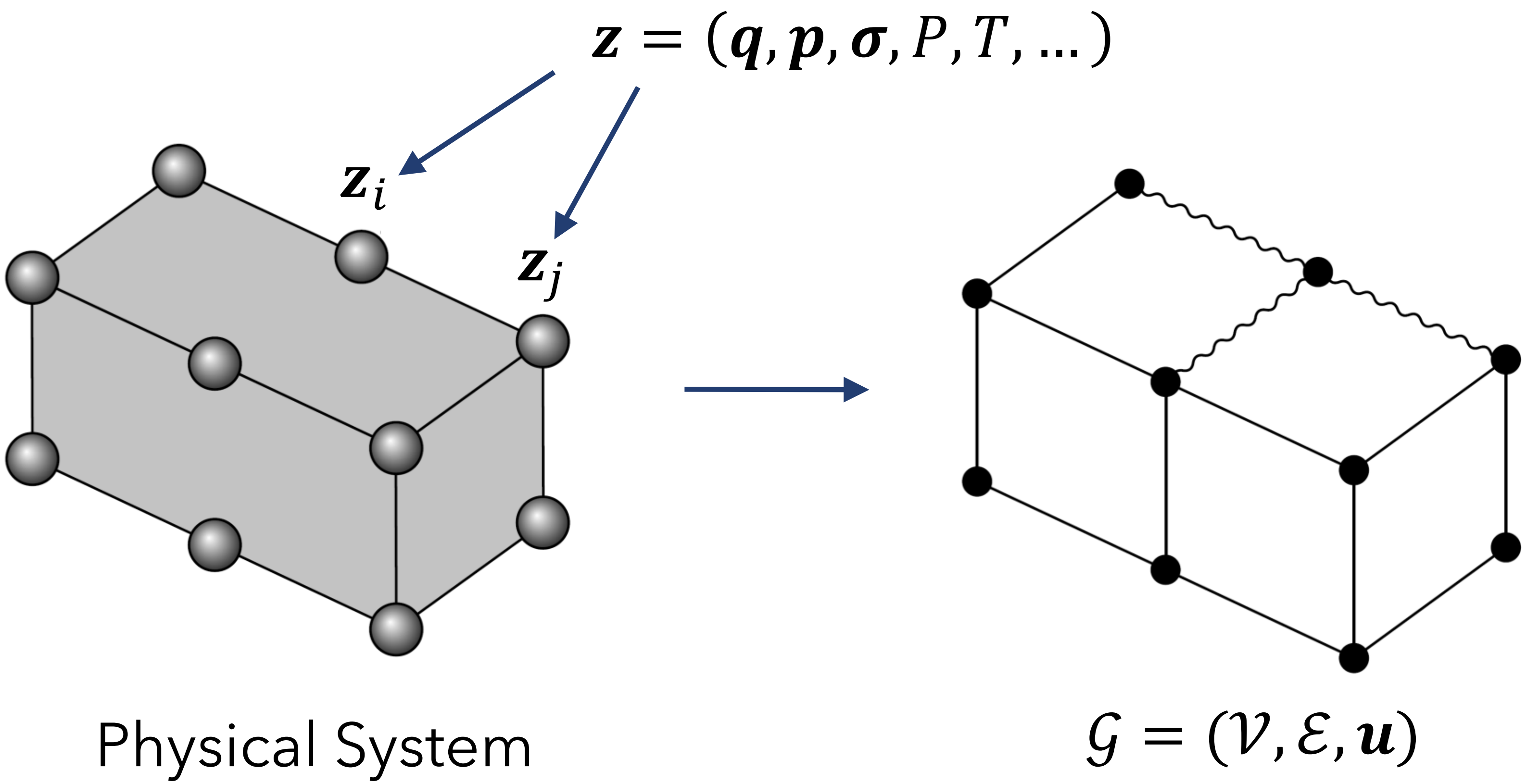
# Geometric bias

- Thermodynamics-Informed Graph Neural Networks [Hernández, 2022]
  - Learn dynamics from data
  - State Variables: $\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{p}, \dots)$



# Geometric bias

- Graph construction

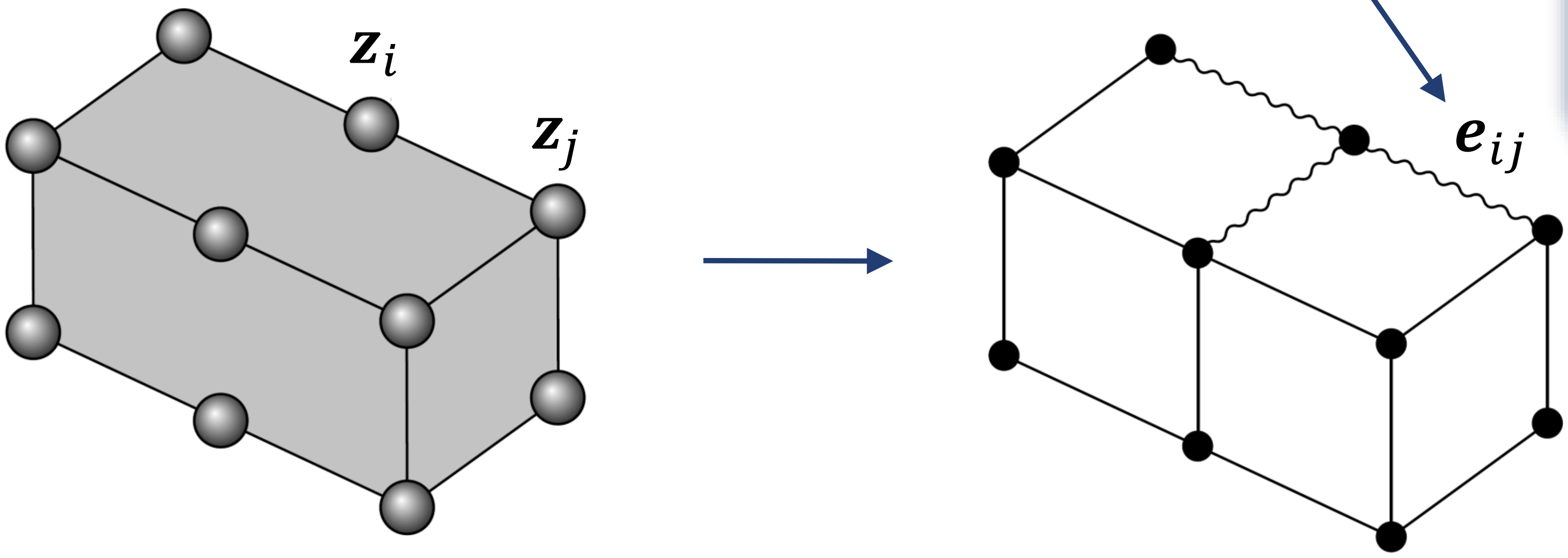


# Geometric bias

- Graph construction

$$\boldsymbol{e}_{ij} = (\boldsymbol{q}_i - \boldsymbol{q}_j, |\boldsymbol{q}_i - \boldsymbol{q}_j|)$$

**Translation Equivariant**

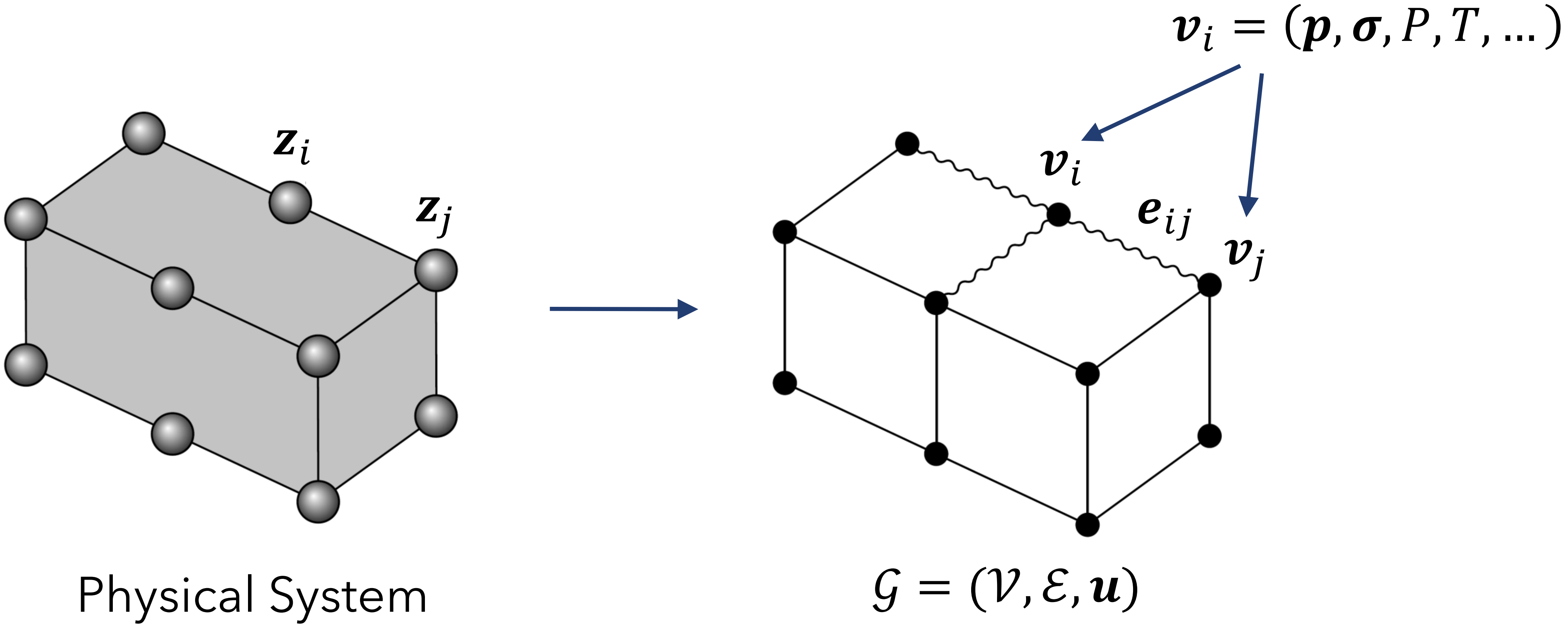



Physical System

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{u})$

# Geometric bias

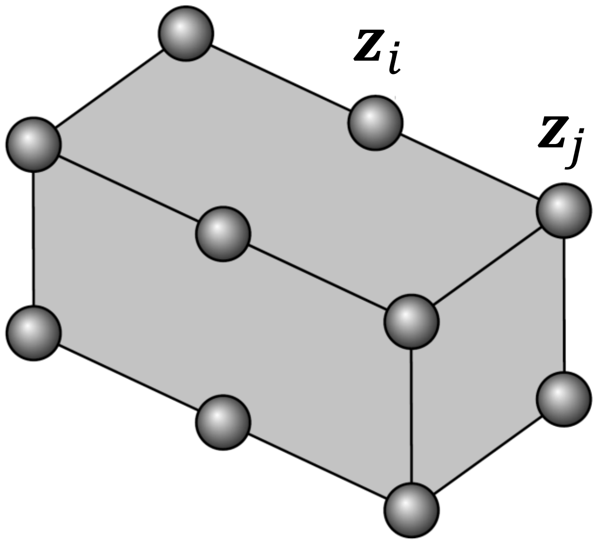
- Graph construction



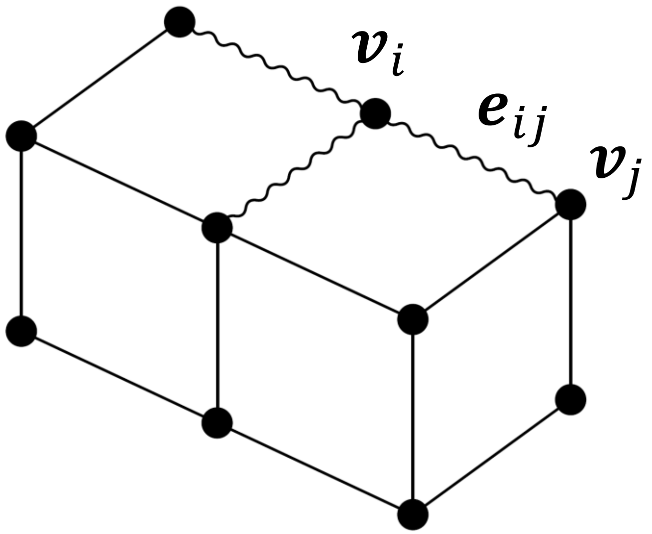
# Geometric bias

- Graph construction

$$\boldsymbol{u} = (g, v, Re, \dots)$$



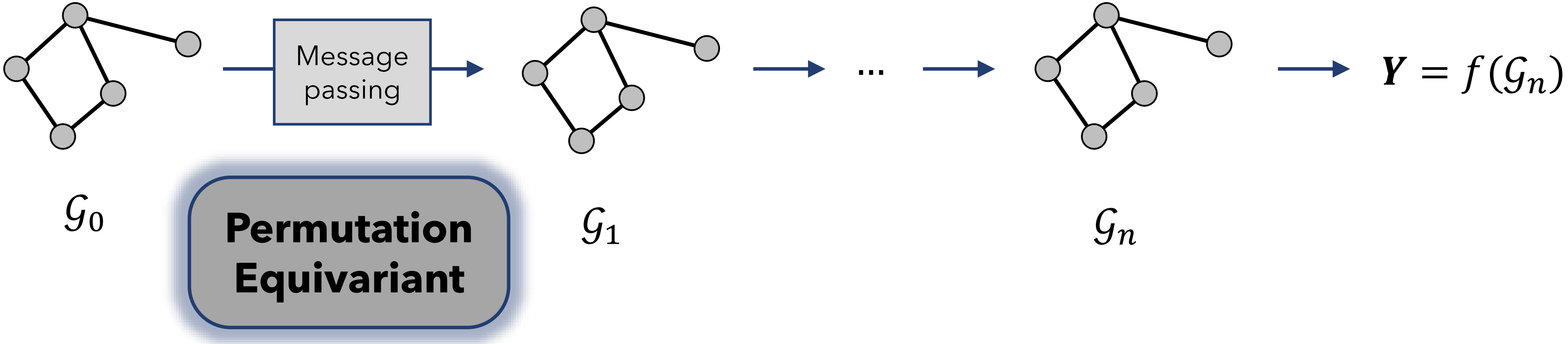

Physical System




$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{u})$$

# Geometric bias

- Encode – Process – Decode [Battaglia, 2018]
  - Node and edge MLPs
  - Message passing algorithm



# Experiments

- Ablation study

| Method | Physics | Geometry |
|---|---|---|
| SPNN [Hernández, 2021] | ✓ | ✗ |
| | | |
| | | |

# Experiments

- Ablation study

| Method | Physics | Geometry |
|---|---|---|
| SPNN [Hernández, 2021] | ✓ | ✗ |
| GNN [Pfaff, 2021] | ✗ | ✓ |
| | | |

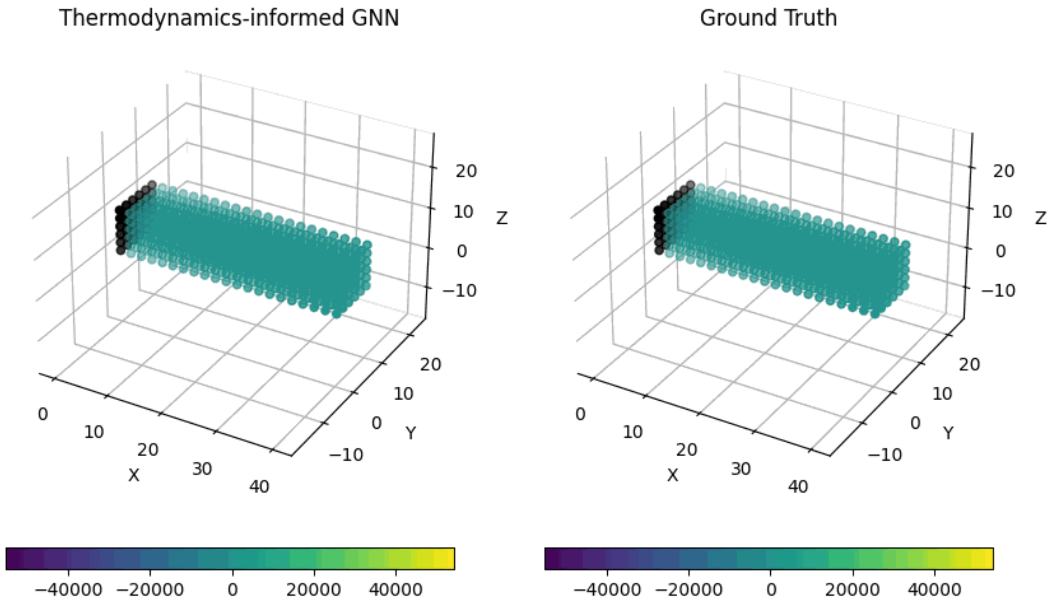
# Experiments

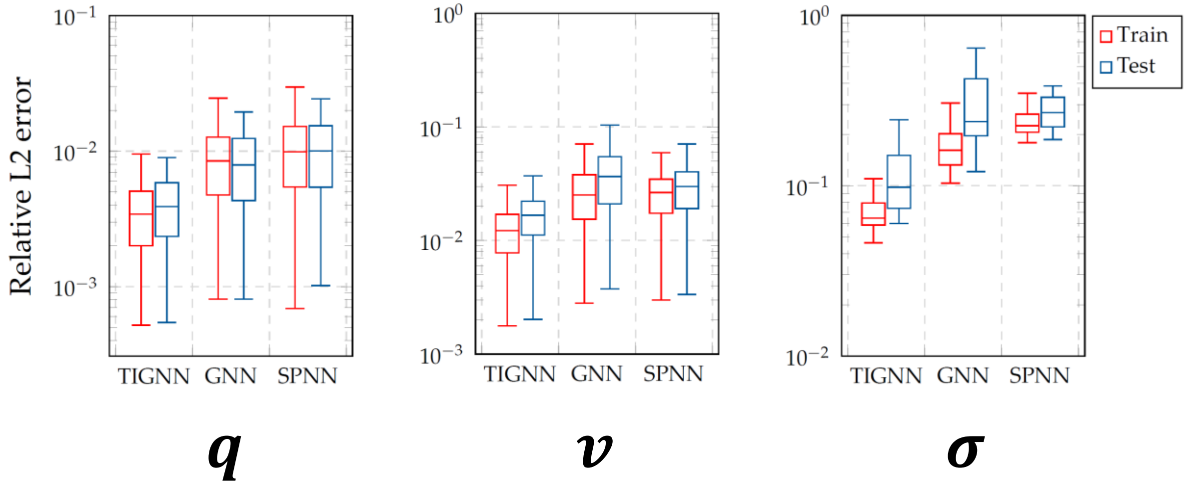
- Ablation study

| Method | Physics | Geometry |
|---|---|---|
| SPNN [Hernández, 2021] | ✓ | ✗ |
| GNN [Pfaff, 2021] | ✗ | ✓ |
| TIGNN [Hernández, 2022] | ✓ | ✓ |

# Experiments

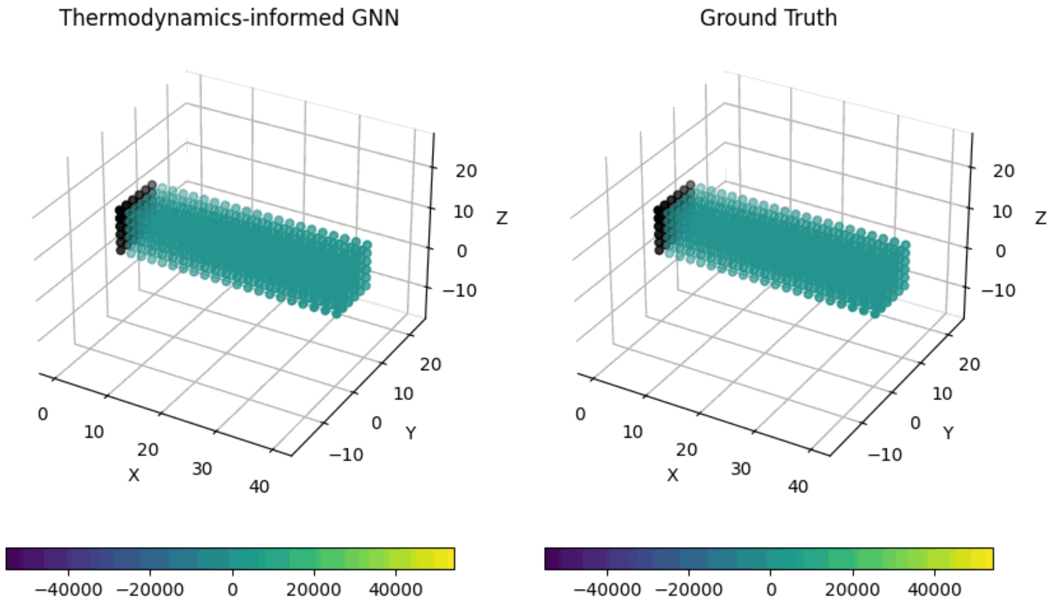
- Bending viscoelastic beam
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Dataset: 52 load positions

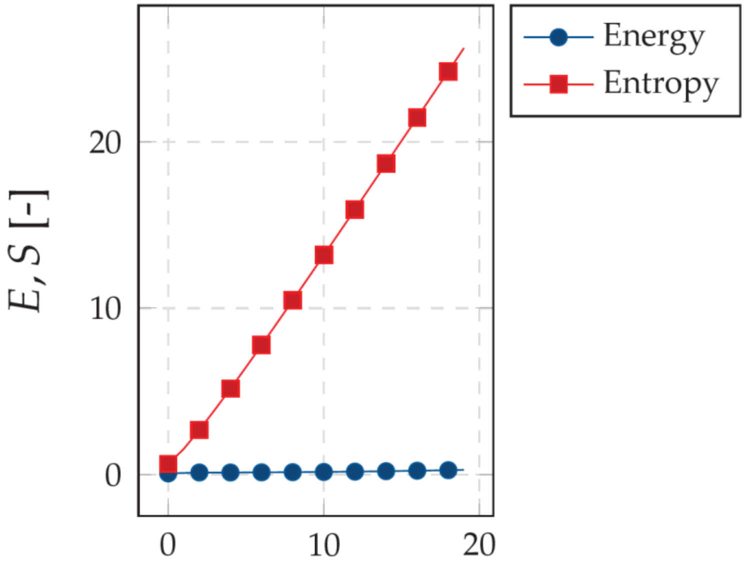


# Experiments

- Bending viscoelastic beam
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Dataset: 52 load positions

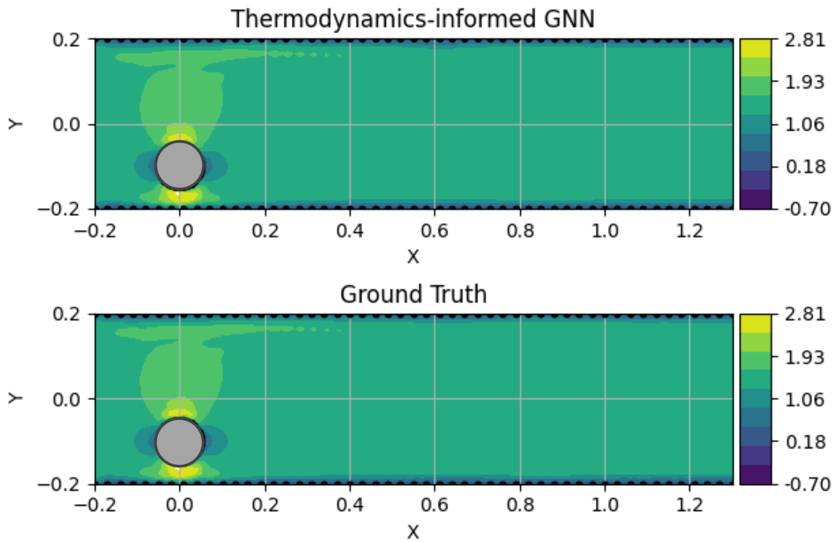


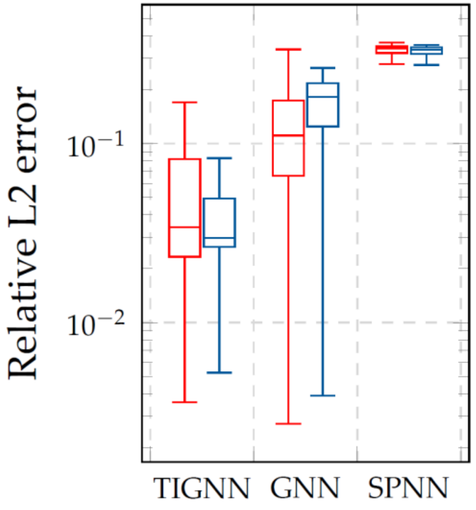



# Experiments

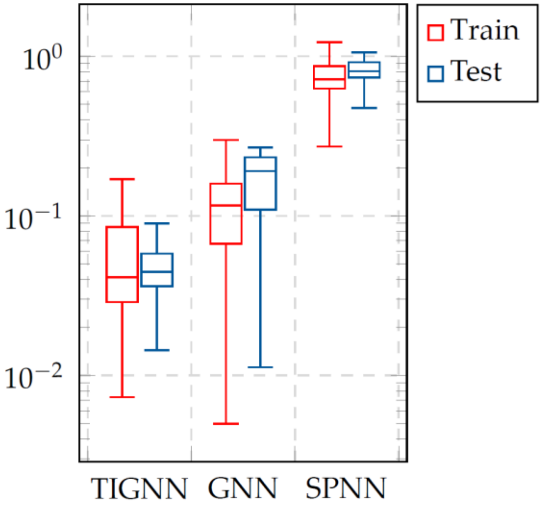
- Flow past a cylinder
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{v}, P)\}$
  - Dataset: 30 geometries + $\boldsymbol{v}$

**Mesh agnostic**



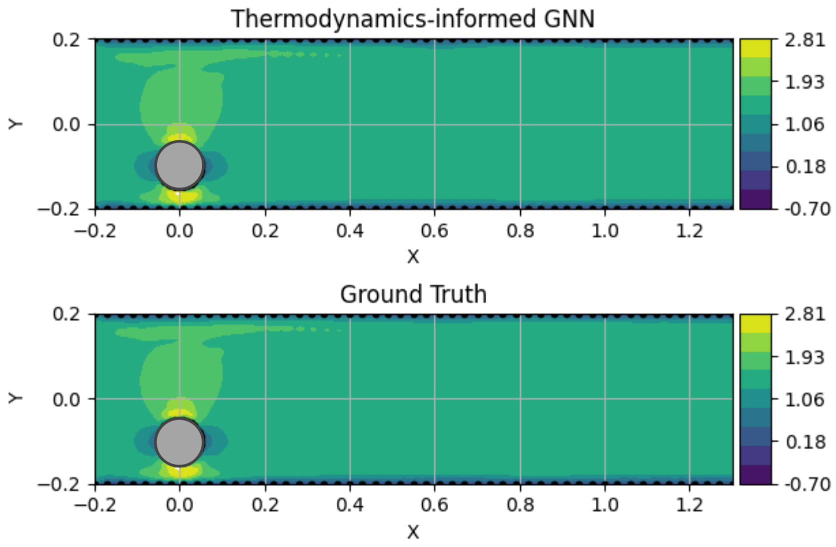

$\boldsymbol{v}$

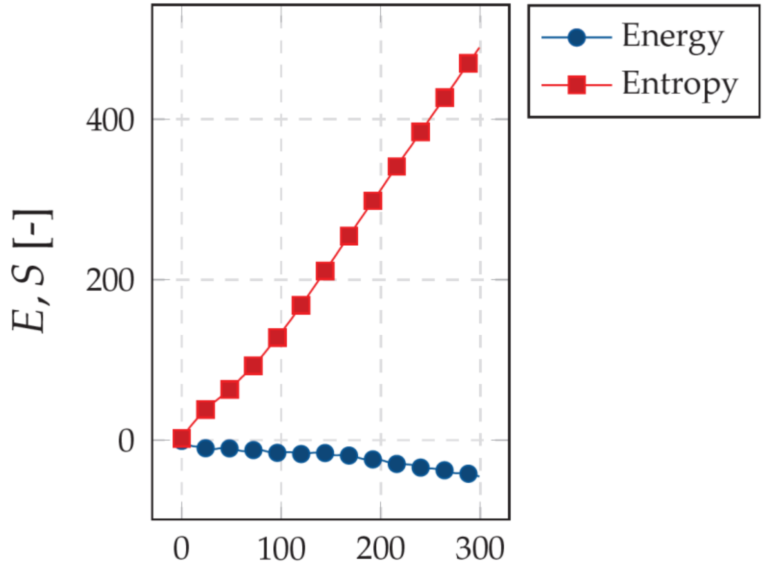



$P$

# Experiments

- Flow past a cylinder
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{v}, P)\}$
  - Dataset: 30 geometries + $\boldsymbol{v}$



# Discussion

- Methodology to learn **dissipative** dynamics from data
- Structure-preserving inductive biases
  - GENERIC structure ensures **thermodynamical consistency**
  - Graph-based computations enforce **geometric priors**
- Outperform black-box approaches

**Limitations**

- Message passing
- Large degrees of freedom
- Fixed meshes in time

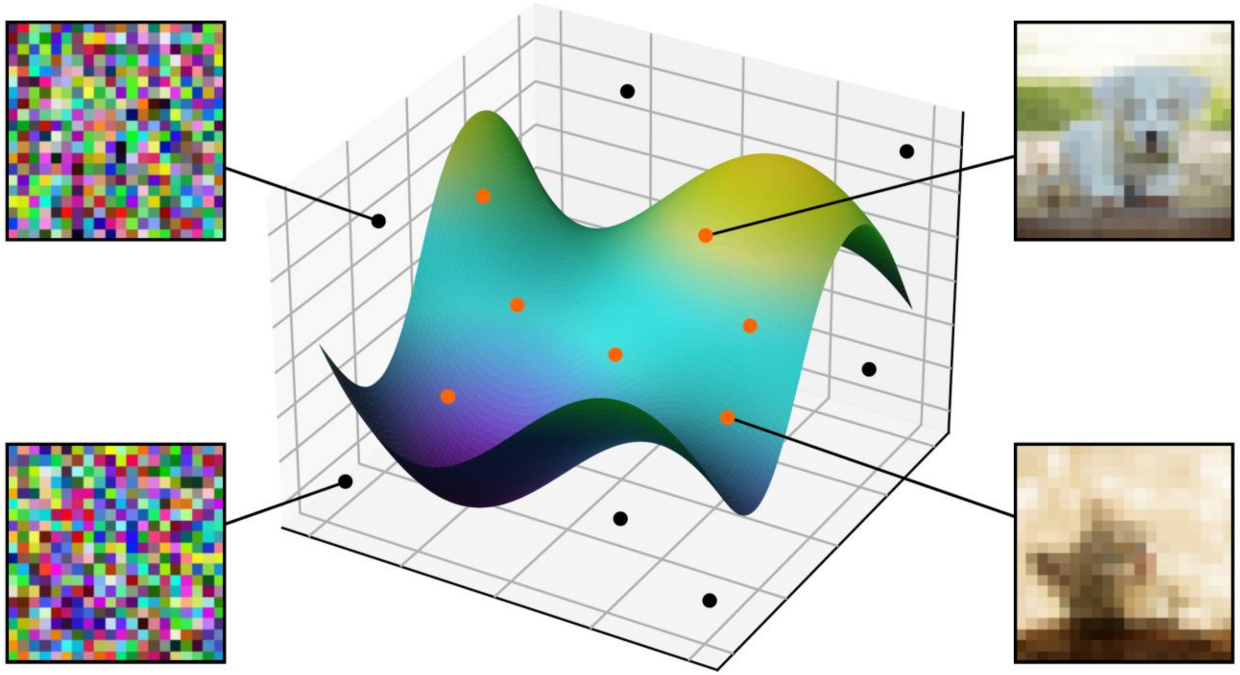
# Part III

## Latent Manifold Learning

# Motivation

- Curse of dimensionality
  - 20 x 20 RGB images

$$D = 7.7 \cdot 10^{2889}$$

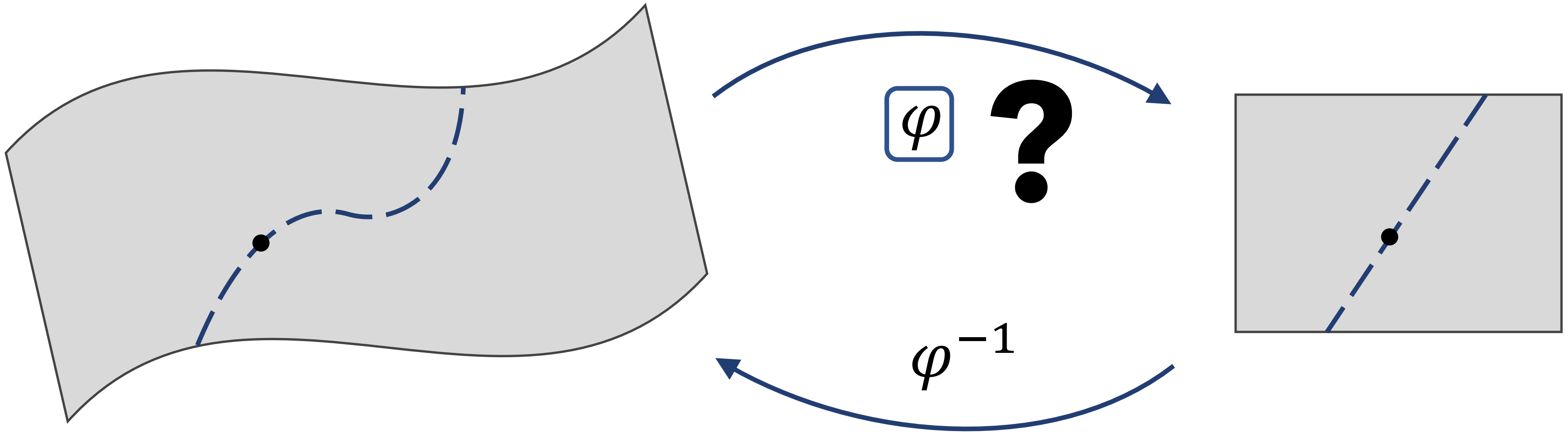


**Manifold Hypothesis**

# Problem Statement

- Curse of dimensionality

$$\boldsymbol{z} = (z_1, z_2, \ldots) \in \mathbb{R}^D \qquad \boldsymbol{x} = (x_1, x_2, \ldots) \in \mathbb{R}^d$$




$$\dot{\boldsymbol{z}} = \frac{d\boldsymbol{z}}{dt} = F(\boldsymbol{z}, t) \qquad \dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt} = \boxed{f(\boldsymbol{x}, t)}\ ?$$

# Problem Statement

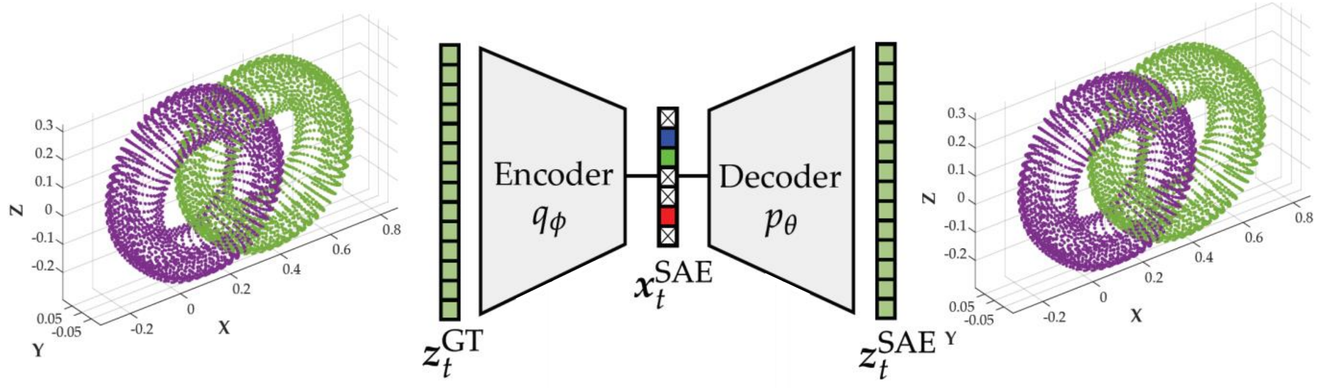
- Model order reduction

| Linear | Non-Linear |
|---|---|
| POD [Niroomandi, 2008] | LLE [Badías, 2019] |
| RB [Prud'homme, 2002] | TDA [Moya, 2019] |
| Galerkin [Rowley, 2004] | k-PCA [Moya, 2020] |
| PGD [Ammar, 2007] | $\boxed{\text{AE [Hernández, 2021]}}$ |
| $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{z}$ | $\boldsymbol{x} = \varphi(\boldsymbol{z})$ |

Reduced basis

# Methodology

- Sparse Autoencoder

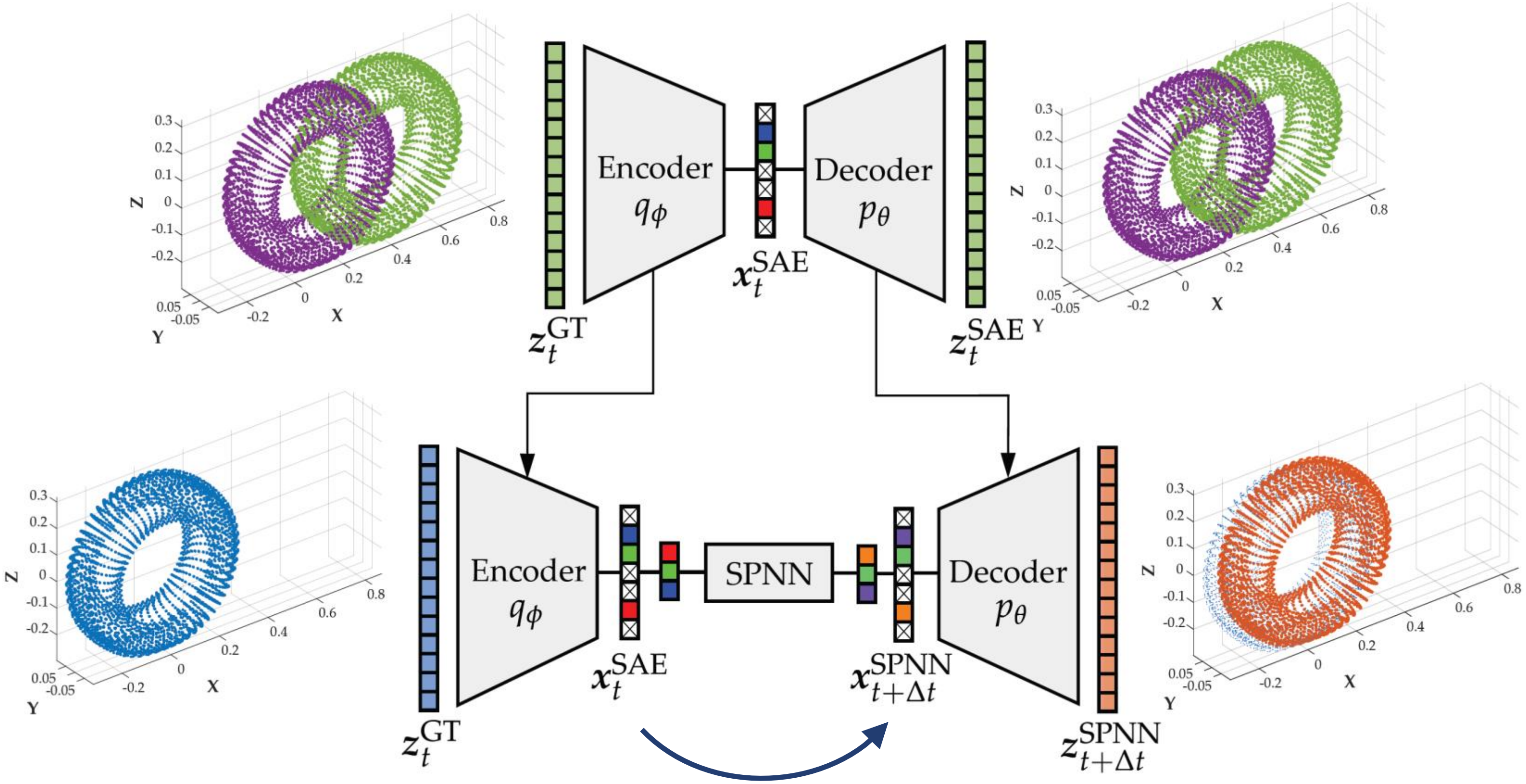



$$\left.\begin{aligned} \boldsymbol{x} &= q_\phi(\boldsymbol{z}) \\ \boldsymbol{z} &= p_\theta(\boldsymbol{x}) \end{aligned}\right\} \quad p_\theta \circ q_\phi = \boldsymbol{I}$$

**+ Sparsity Constraints**

# Methodology

- Reduced integration



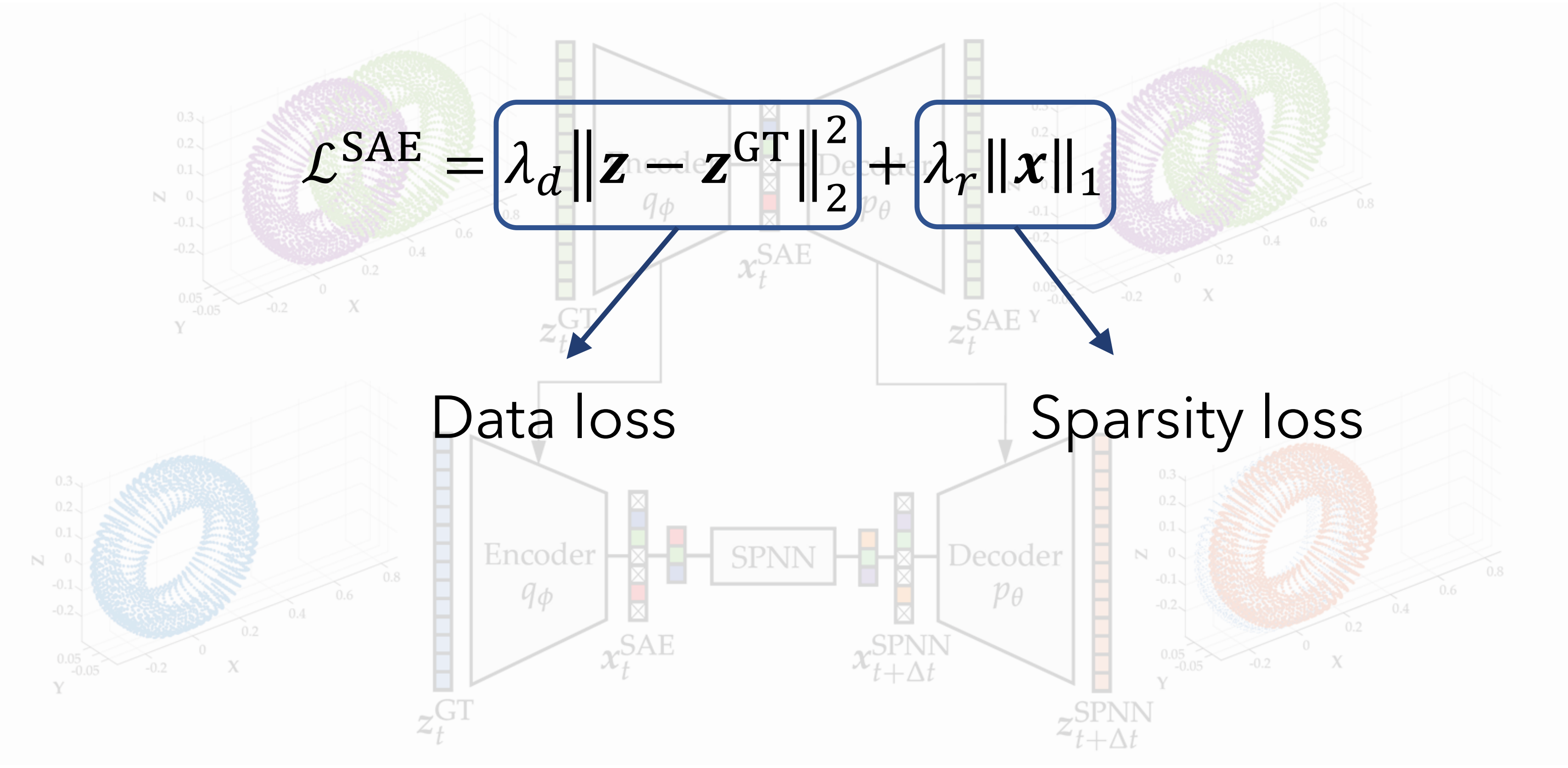
# Methodology

- Learning procedure

$$\mathcal{L}^{\mathrm{SAE}} = \boxed{\lambda_d \left\| \boldsymbol{z} - \boldsymbol{z}^{\mathrm{GT}} \right\|_2^2} + \boxed{\lambda_r \left\| \boldsymbol{x} \right\|_1}$$

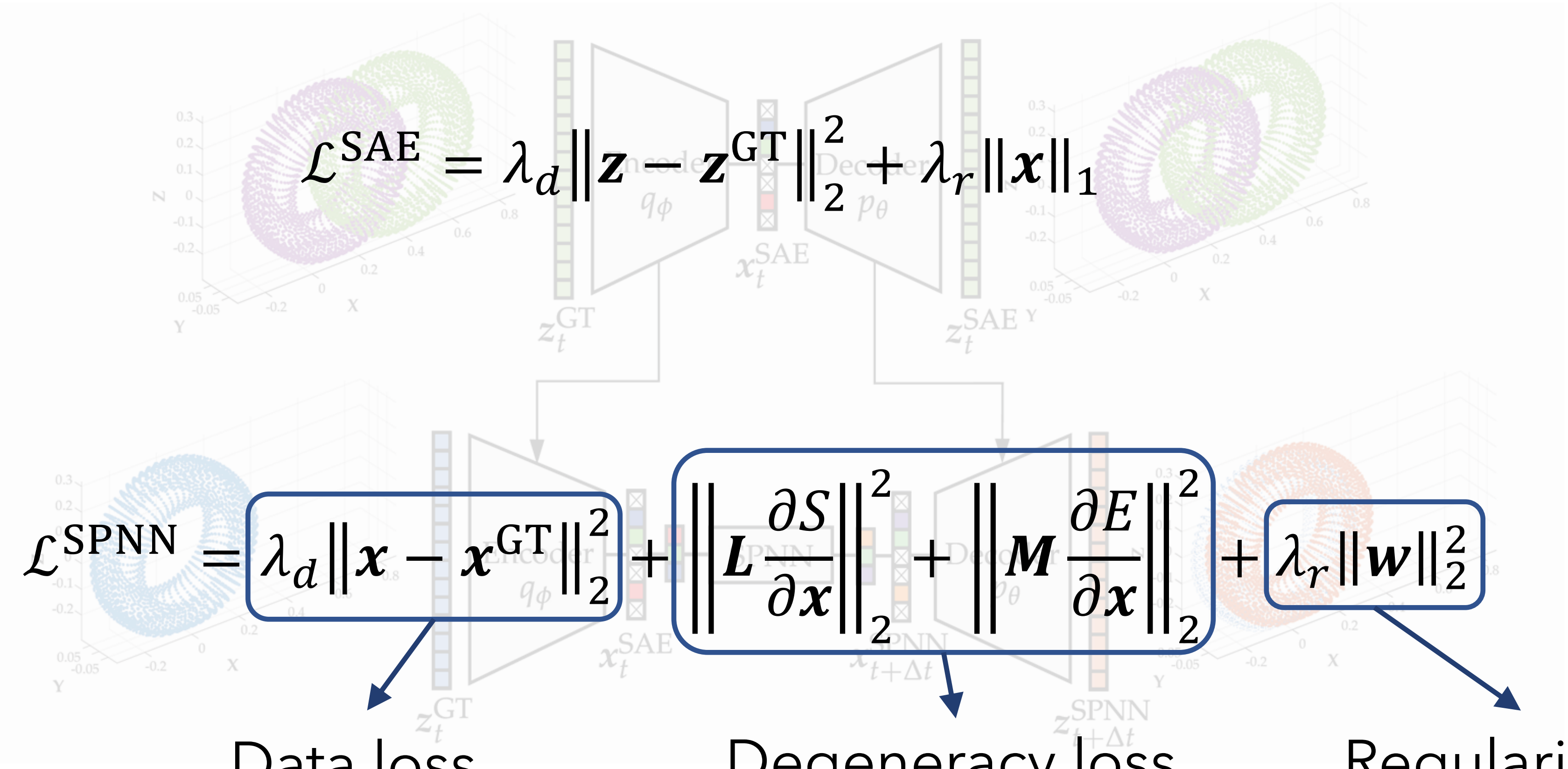
Data loss

Sparsity loss



# Methodology

- Learning procedure



$$\mathcal{L}^{\text{SAE}} = \lambda_d \left\| \boldsymbol{z} - \boldsymbol{z}^{\text{GT}} \right\|_2^2 + \lambda_r \|\boldsymbol{x}\|_1$$

$$\mathcal{L}^{\text{SPNN}} = \boxed{\lambda_d \left\| \boldsymbol{x} - \boldsymbol{x}^{\text{GT}} \right\|_2^2} + \boxed{\left\| \boldsymbol{L} \frac{\partial S}{\partial \boldsymbol{x}} \right\|_2^2 + \left\| \boldsymbol{M} \frac{\partial E}{\partial \boldsymbol{x}} \right\|_2^2} + \boxed{\lambda_r \|\boldsymbol{w}\|_2^2}$$

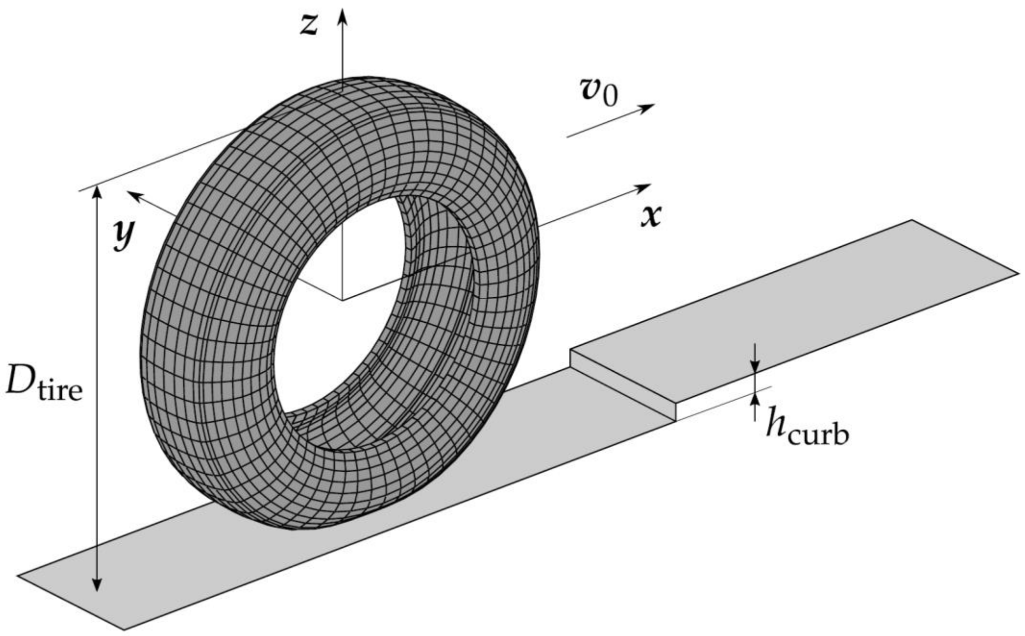
Data loss

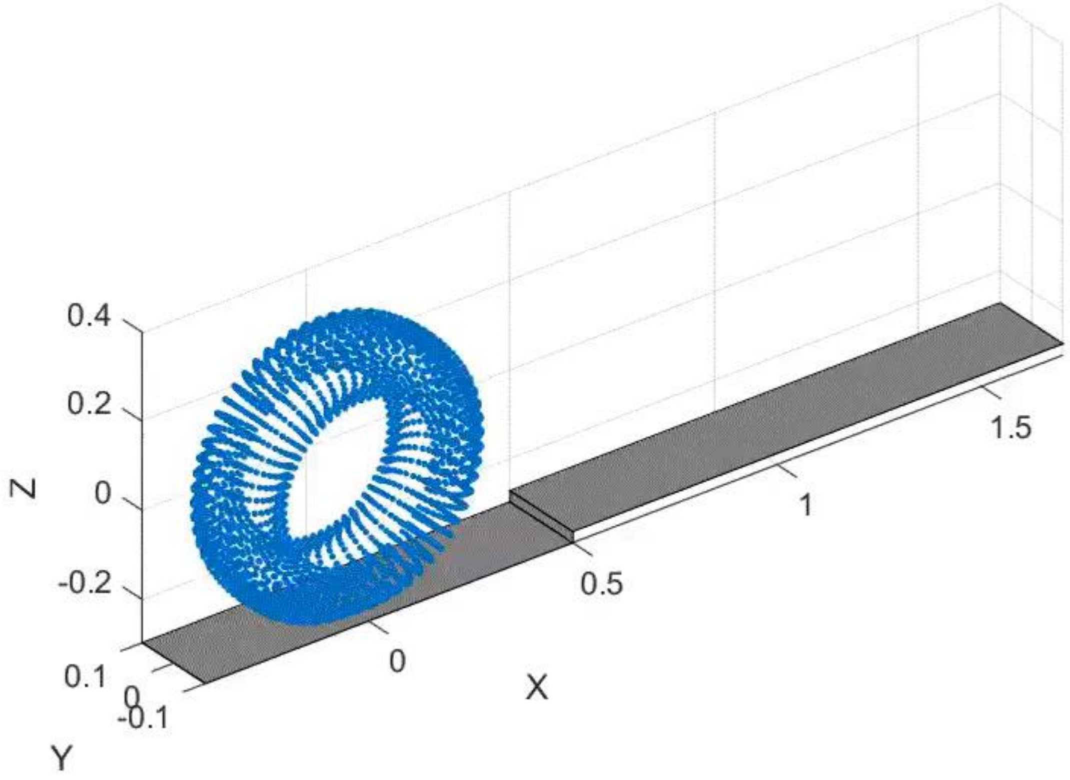
Degeneracy loss

Regularization

# Experiments

- Rolling hyperelastic tire
  - State Space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Dataset: 200 snapshots

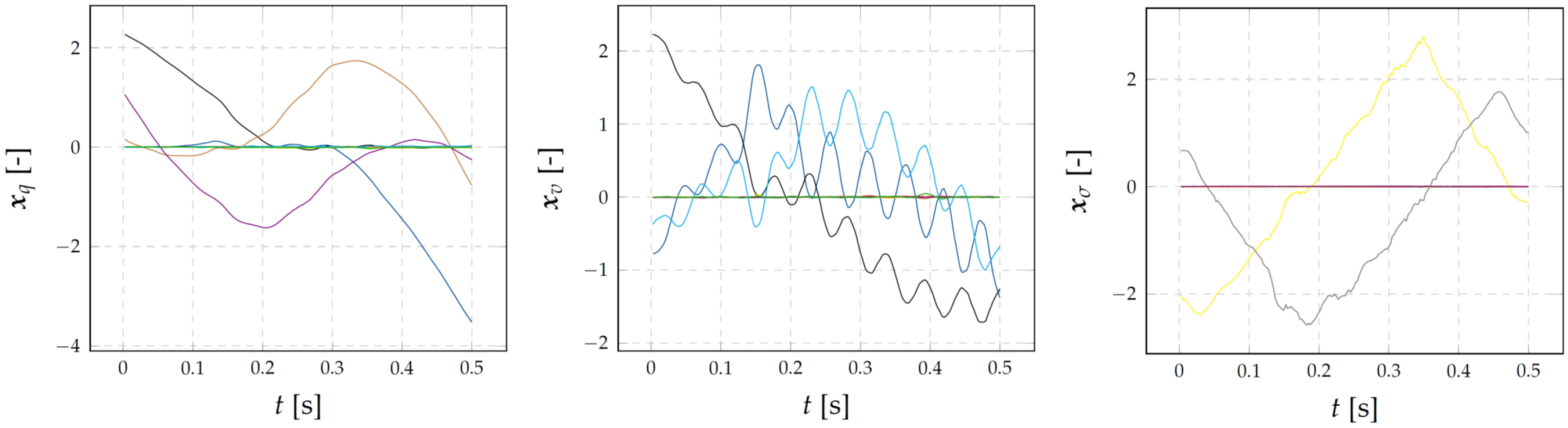


# Experiments

- Rolling hyperelastic tire
  - Results: Latent variables

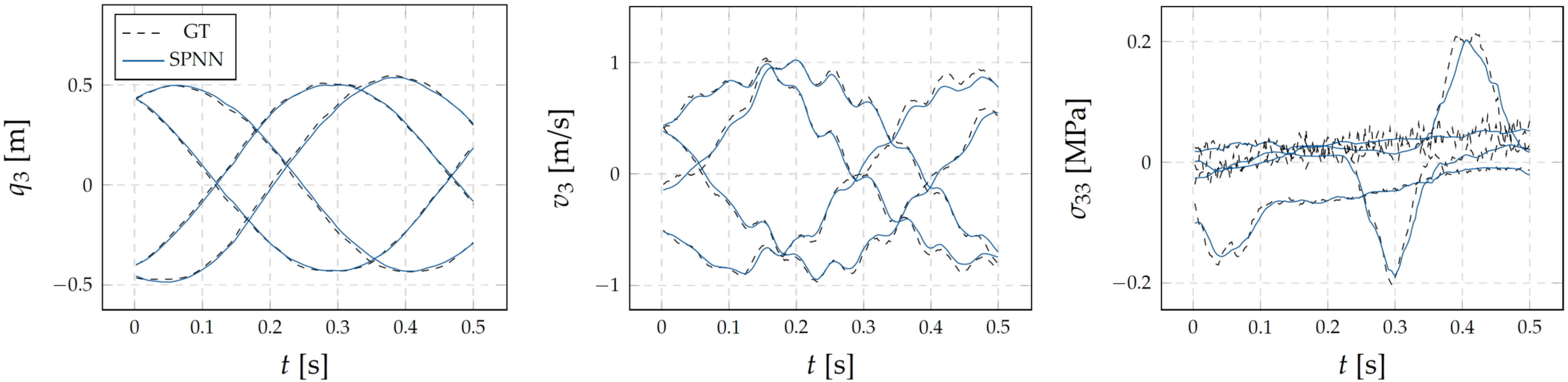
$$D = 50k \longrightarrow d = 9$$




4 variables | 3 variables | 2 variables

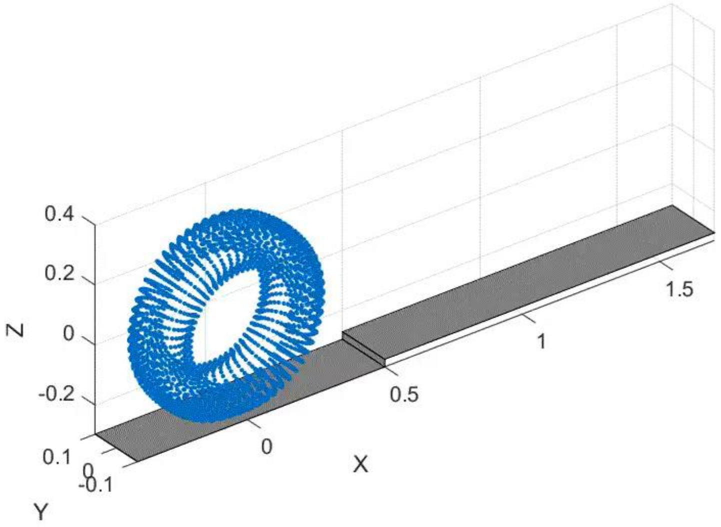
# Experiments

- Rolling hyperelastic tire
  - Results: Time evolution

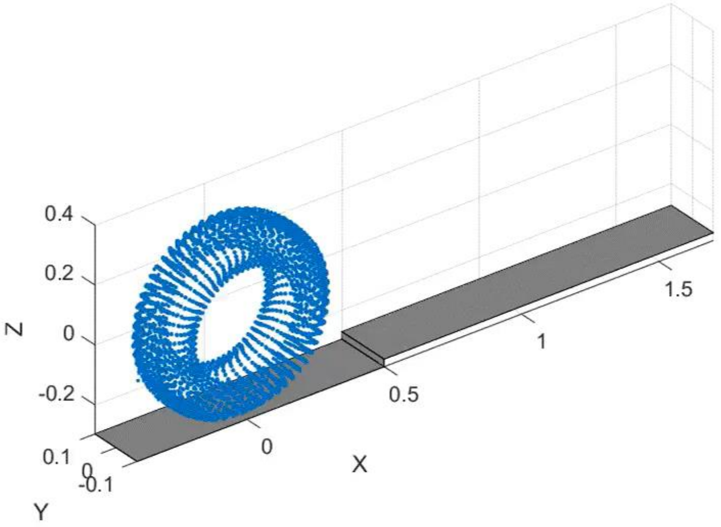


# Experiments

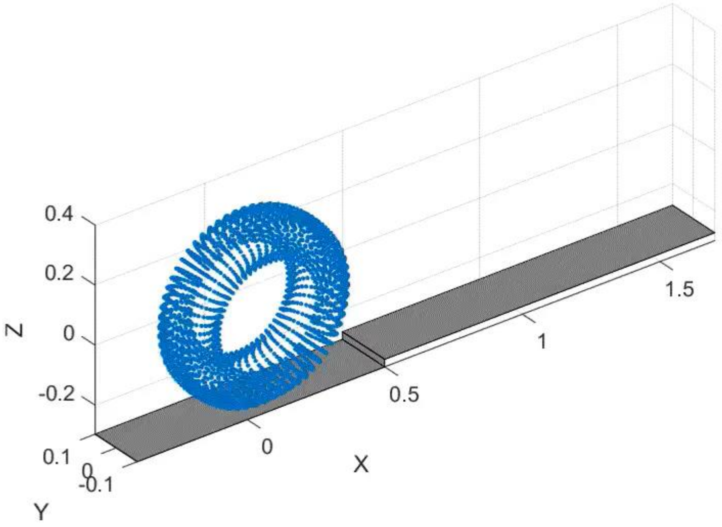
- Rolling hyperelastic tire
  - Results: Rollout animation




Ground Truth



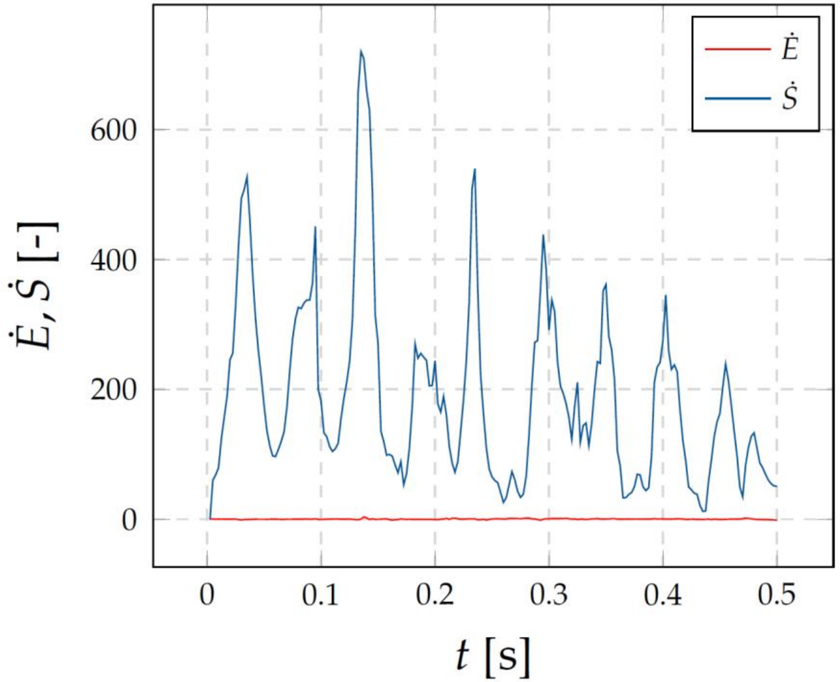

Ours




POD

# Experiments

- Rolling hyperelastic tire
  - Results: Thermodynamic consistency



# Discussion

- Structure-preserving **reduced order model** from data
- **Unsupervised** identification of the latent dimensionality
- Great compression ratio with low global error

**Limitations**

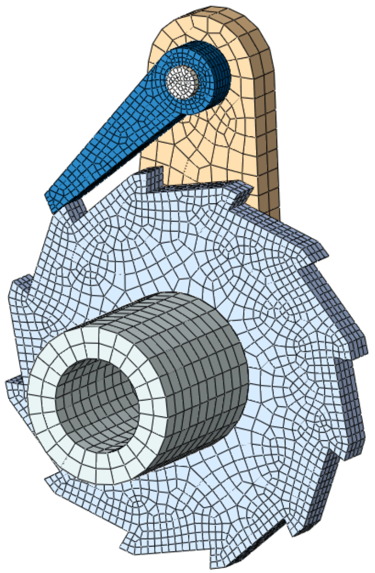
- Fixed mesh
- Spectral bias
- Hyperparameter tuning

# Part IV

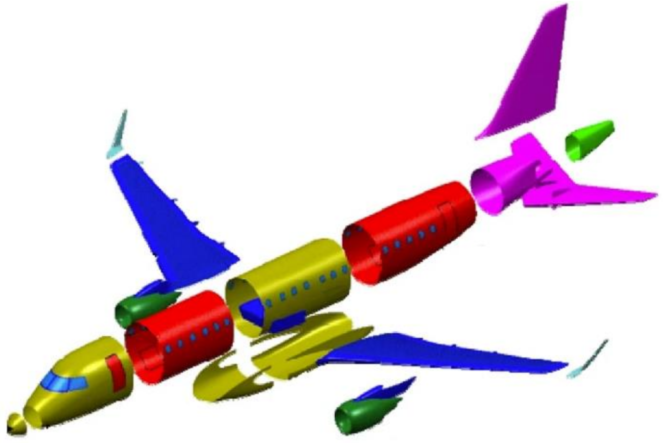
## Applications to Complex Systems

# Motivation

- Complex coupled systems

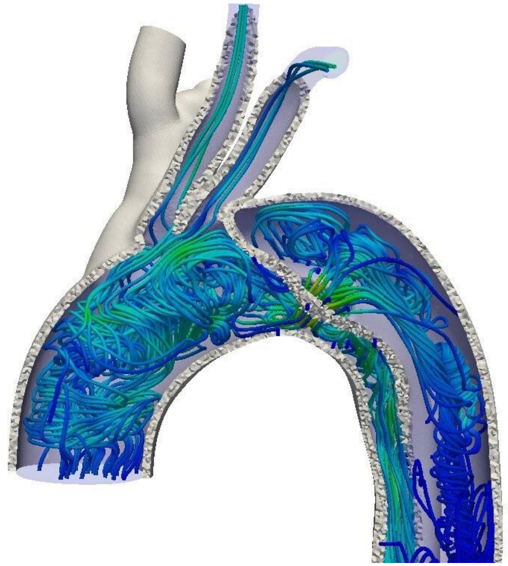


High degrees of freedom



[Britto Maria, 2019]

System coupling

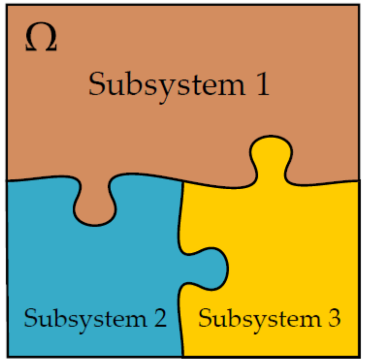


Eostrack

Fluid-Structure Interaction

# Methodology

- Coupled systems
  - Full domain: $\Omega = \bigcup_i \Omega_i$

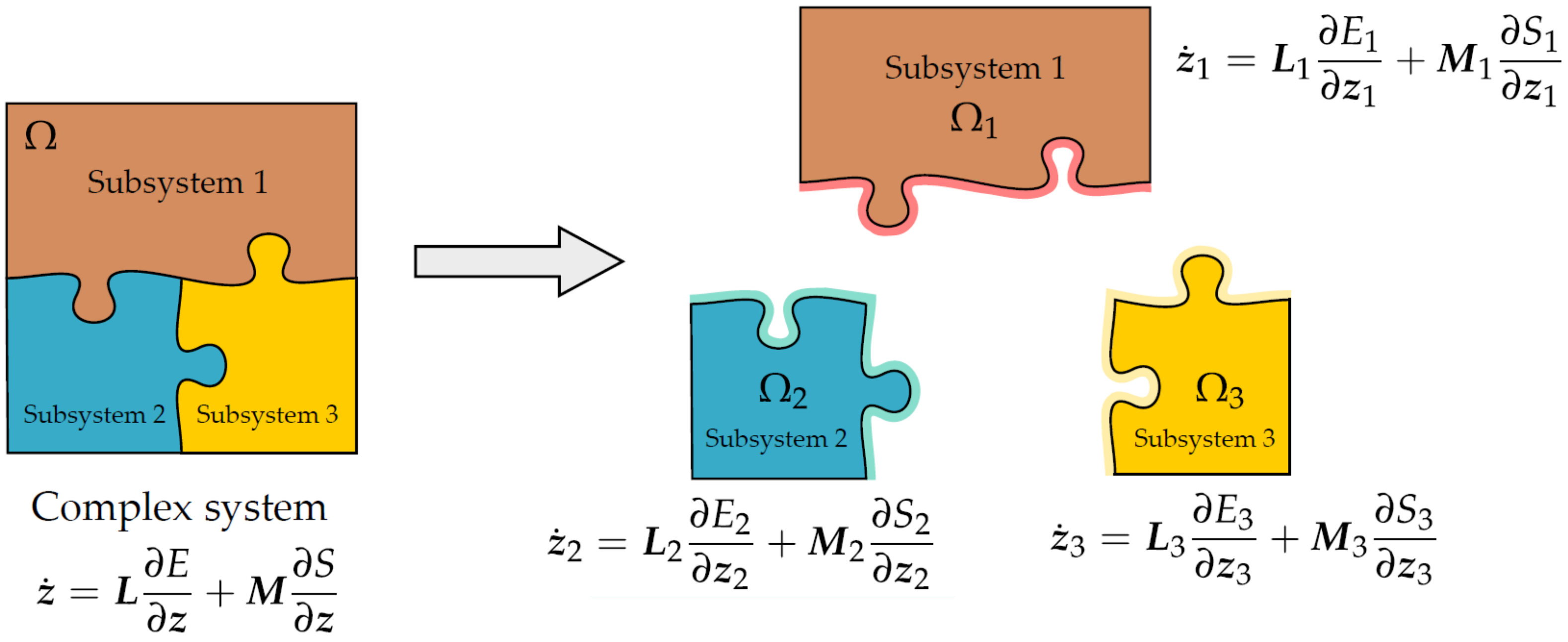



Complex system

$$\dot{z} = \boldsymbol{L}\frac{\partial E}{\partial z} + \boldsymbol{M}\frac{\partial S}{\partial z}$$

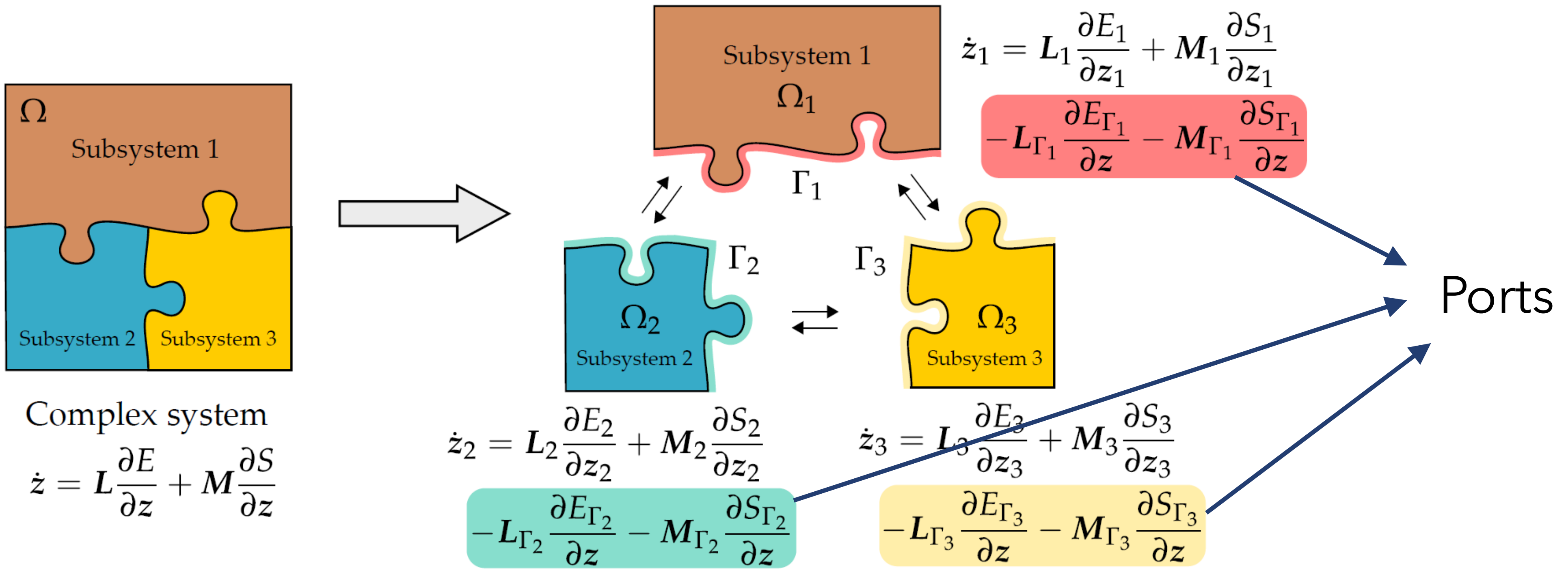
# Methodology

- Coupled systems
  - Full domain: $\Omega = \cup_i \Omega_i$



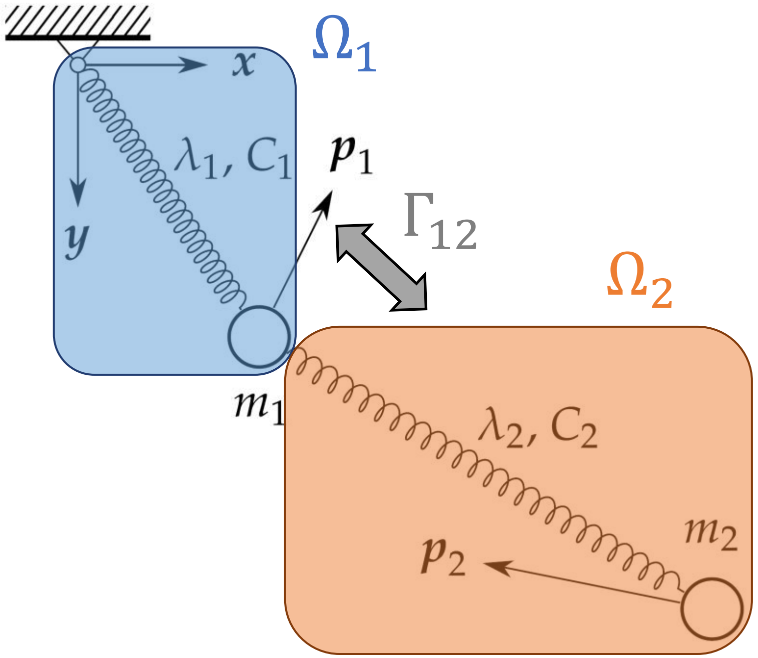
# Methodology

- Coupled systems
  - Full domain: $\Omega = \cup_i \Omega_i$

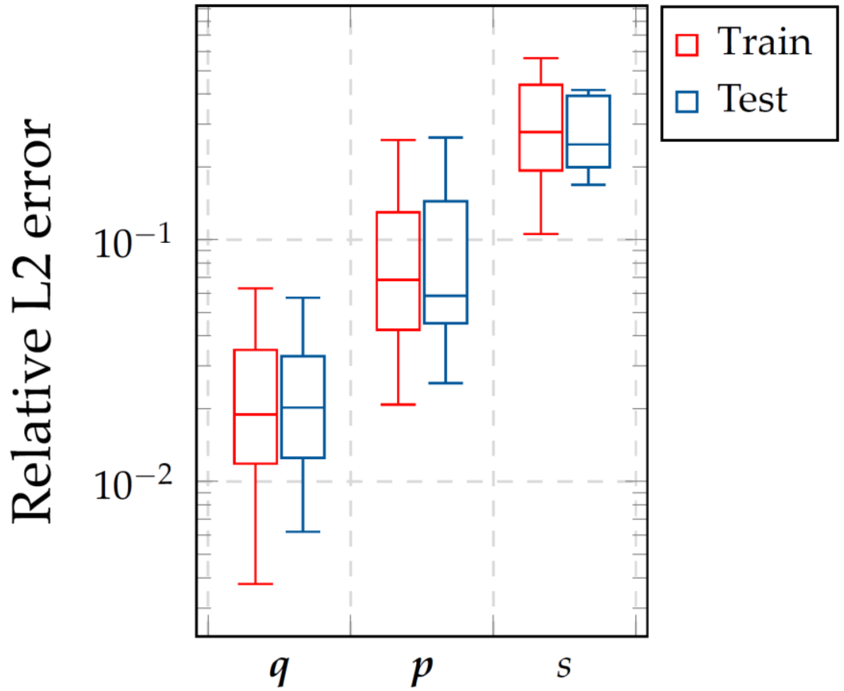


# Experiments

- Double thermoelastic pendulum
  - State space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, s)\}$
  - Database: 50 ICs

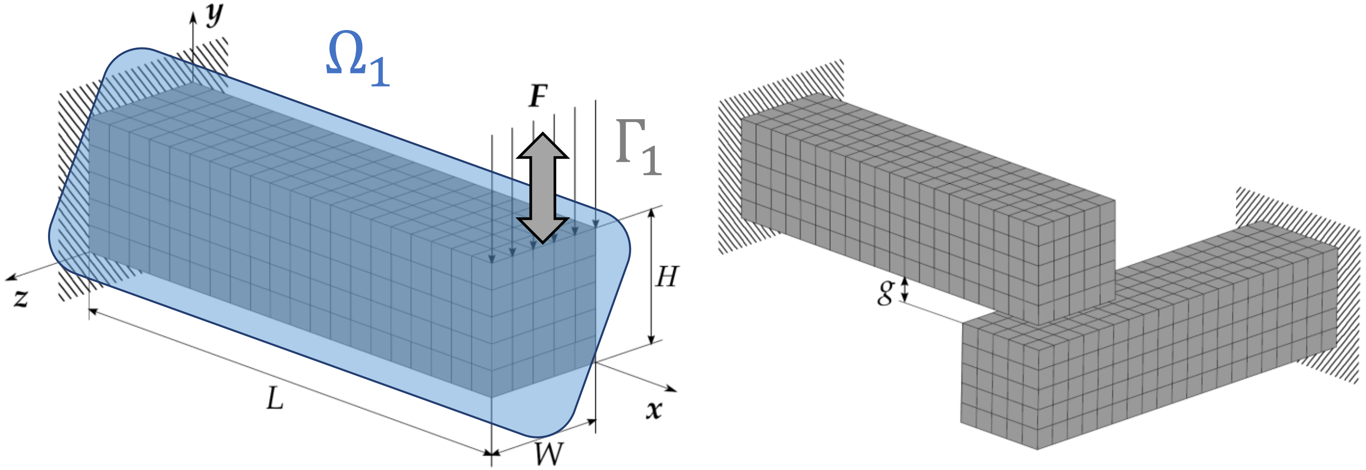


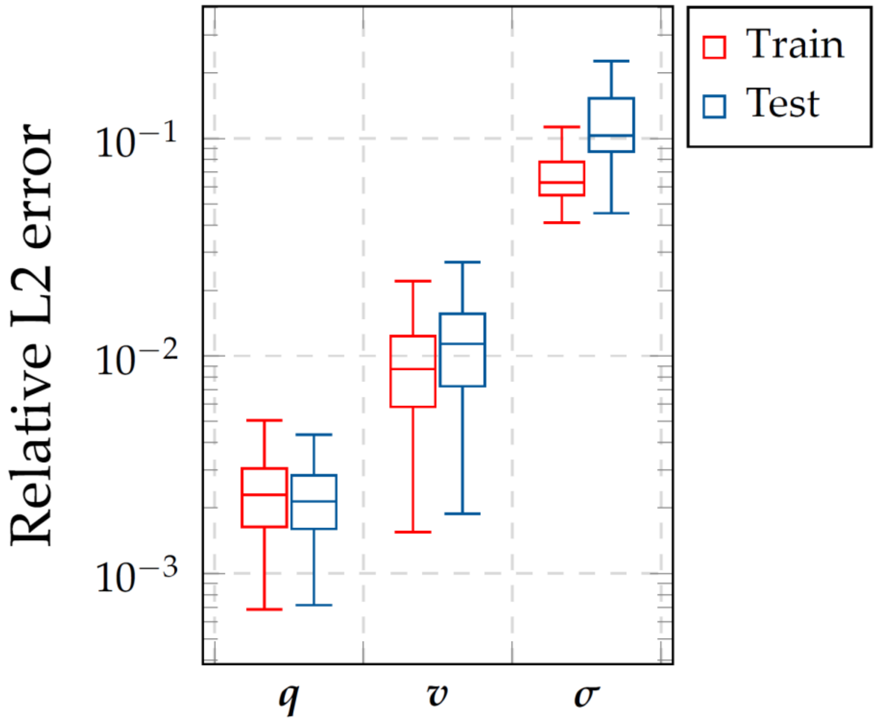

**x2 faster**



# Experiments

- Bending beams
  - State space: $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Database: 52 load positions





Universidad Zaragoza
1542

# Motivation

- Real-time simulations in emerging technologies: VR, AR


Digital twins


AR design


VR Metaverse

**This work**

# Motivation

- Need robust and fast solvers

| Simulator | Physics | Cost |
|---|---|---|
| Traditional (CFD, FEM) | ✓ | ✗ |
| | | |
| | | |
| | | |

# Motivation

- Need robust and fast solvers

| Simulator | Physics | Cost |
|---|---|---|
| Traditional (CFD, FEM) | ✓ | ✗ |
| Engines (Havok, Unreal) | ~ | ✓ |
| | | |
| | | |

[D'Andrea, 2013]

[Wang, 2015]

# Motivation

- Need robust and fast solvers

| Simulator | Physics | Cost |
|---|---|---|
| Traditional (CFD, FEM) | ✓ | ✗ |
| Engines (Havok, Unreal) | ~ | ✓ |
| Deep learning (NNs) | ✗ | ✓ |
| | | |

[Fresca, 2022]

[Romero, 2022]

# Motivation

- Need robust and fast solvers

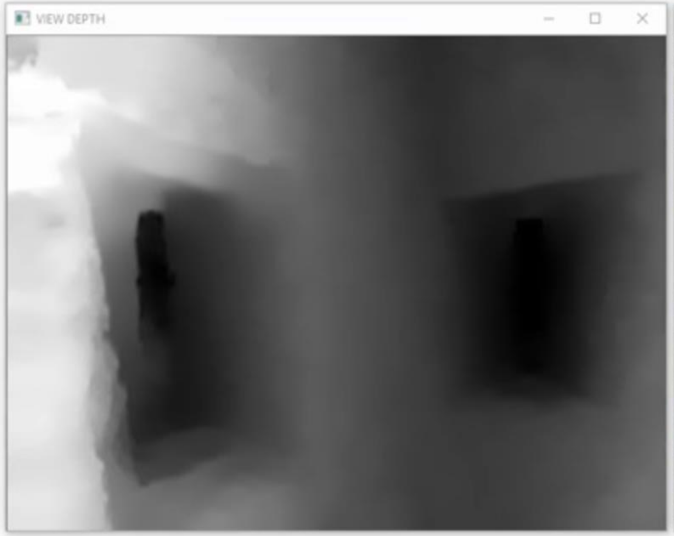
| Simulator | Physics | Cost |
| --- | --- | --- |
| Traditional (CFD, FEM) | ✓ | ✗ |
| Engines (Havok, Unreal) | ~ | ✓ |
| Deep learning (NNs) | ✗ | ✓ |
| Structure-Preserving | ✓ | ✓ |

[Hernández, 2022]

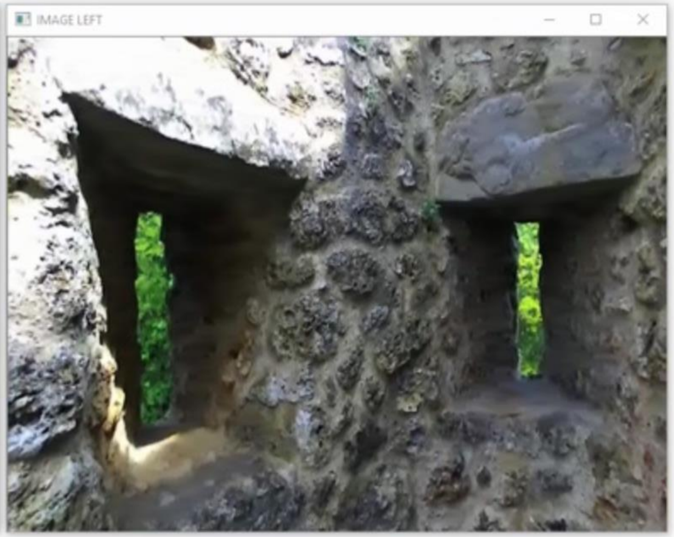
# Vision system

- Environment input: Stereo camera

*Stereolabs*



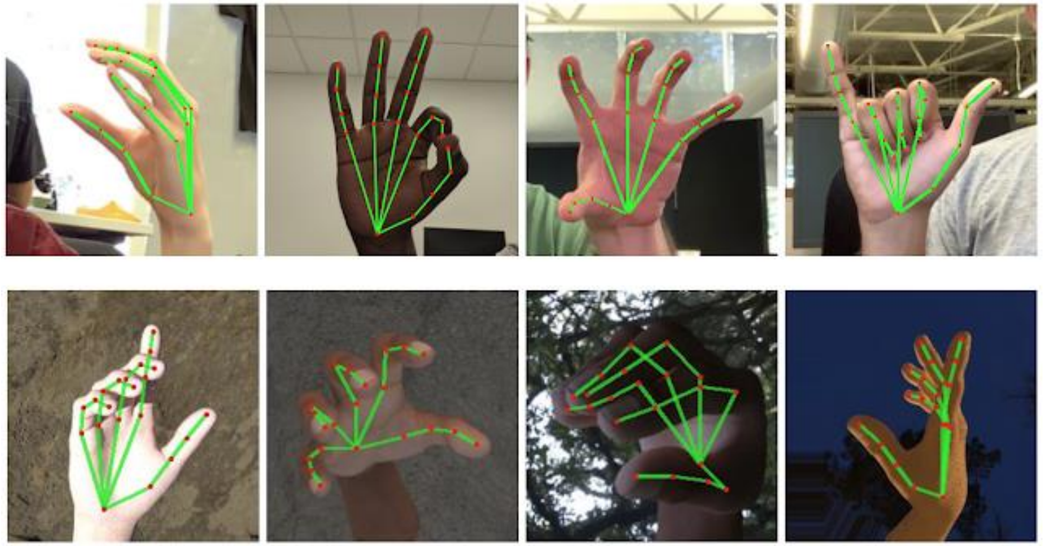

Depth map




RGB Image

# Vision system

- Environment input: Stereo camera
- User input: Hand tracking

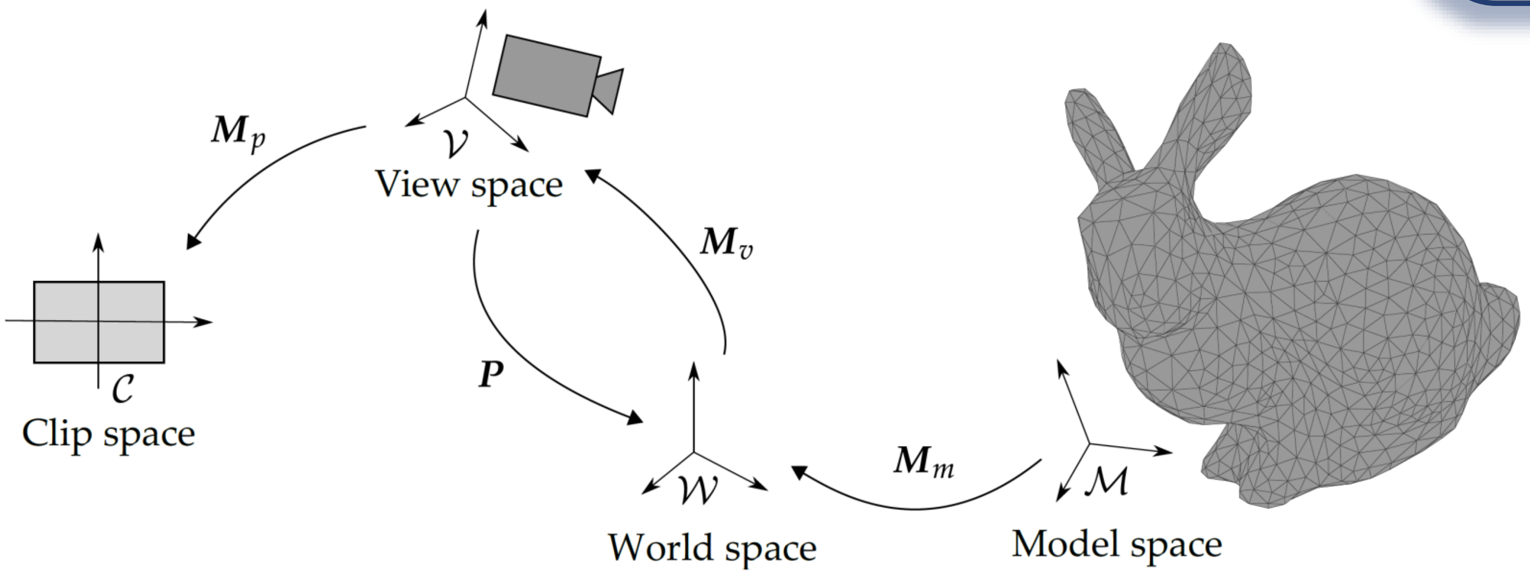


[Zhang, 2020]

# Vision system

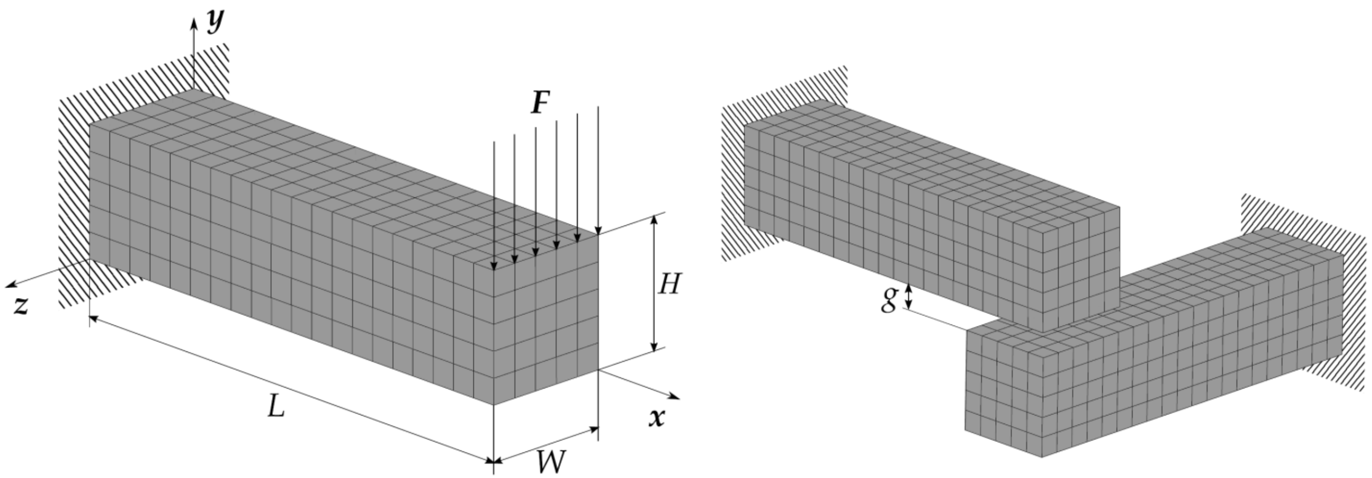
- Environment input: Stereo camera
- User input: Hand tracking
- Interactive object: MVP transformations

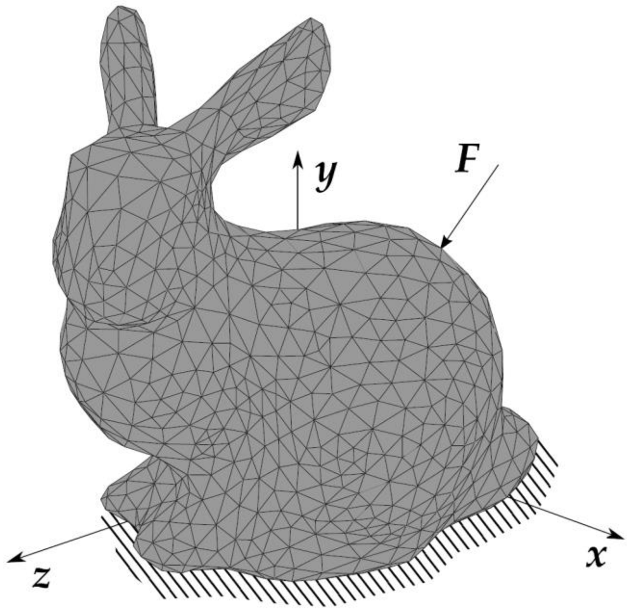
**Rotation Equivariant**



# Experiments

- Viscoelastic beam
  - $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Load positions: 52
- Stanford bunny
  - $\mathcal{S} = \{\boldsymbol{z} = (\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\sigma})\}$
  - Load positions: 100



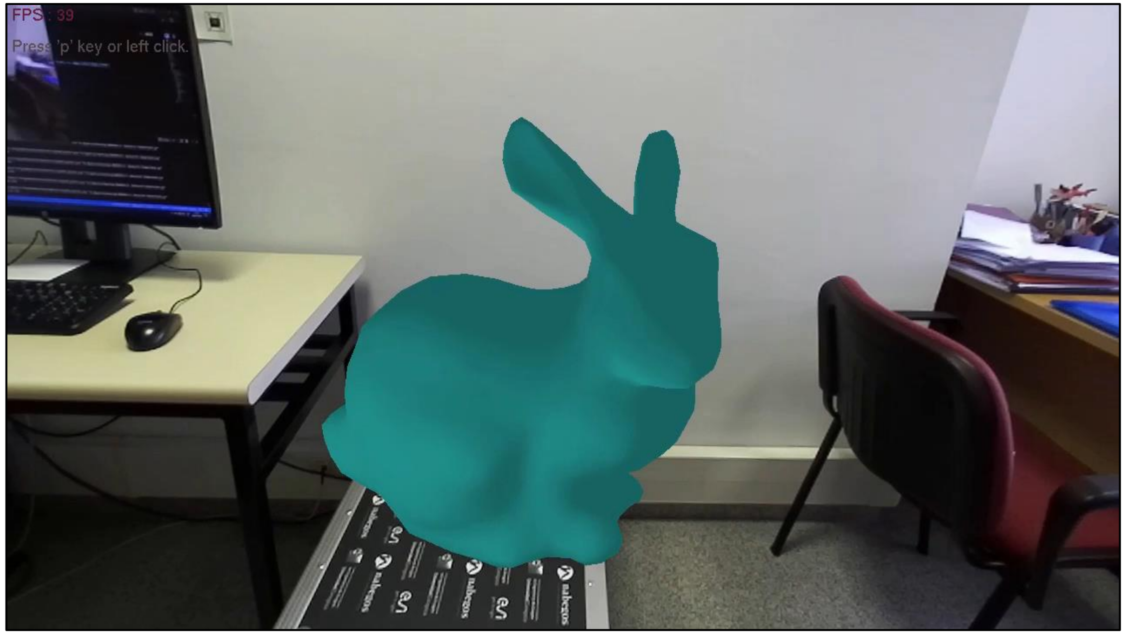



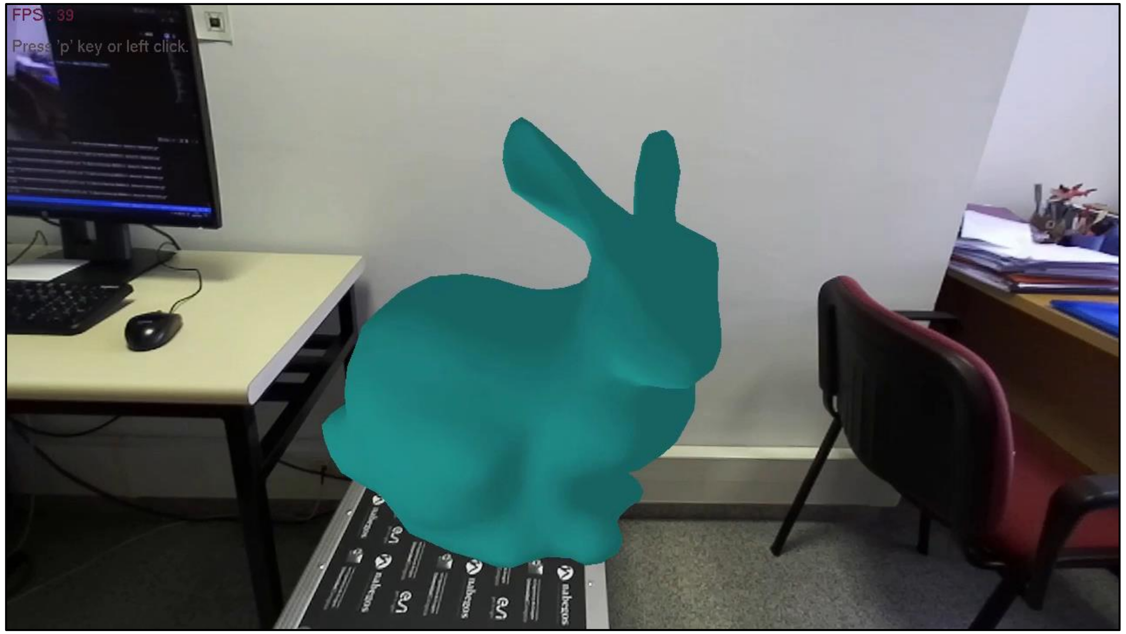
# Experiments

- Viscoelastic beam
  - Results: Video sequence


# Experiments

- Stanford bunny
  - Results: Video sequence





# Discussion

- Port-metriplectic networks as inductive bias for **coupled** systems
- Reduced computational cost in complex applications
- Structure-preserving neural networks in **realtime**
- Realistic interaction with thermodynamical consistency

**Limitations**

- One point of contact
- Computational efficiency
- Depth artifacts

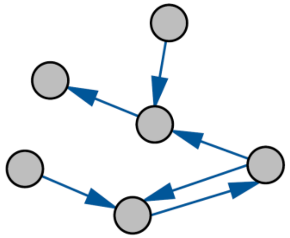
# Part V
## Conclusions

# Conclusions

## Part II

Deep Learning of Dynamical Systems

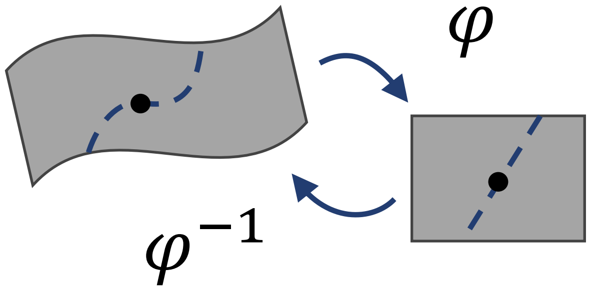
$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}}$$



## Part III

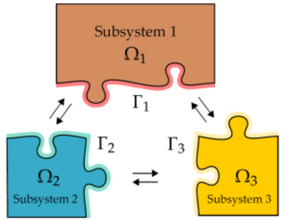
Latent Manifold Learning

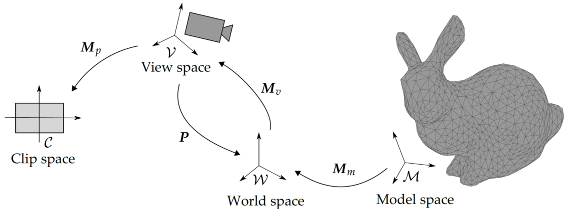
$\boldsymbol{z} \in \mathbb{R}^D$ $\quad$ $\boldsymbol{x} \in \mathbb{R}^d$




## Part IV

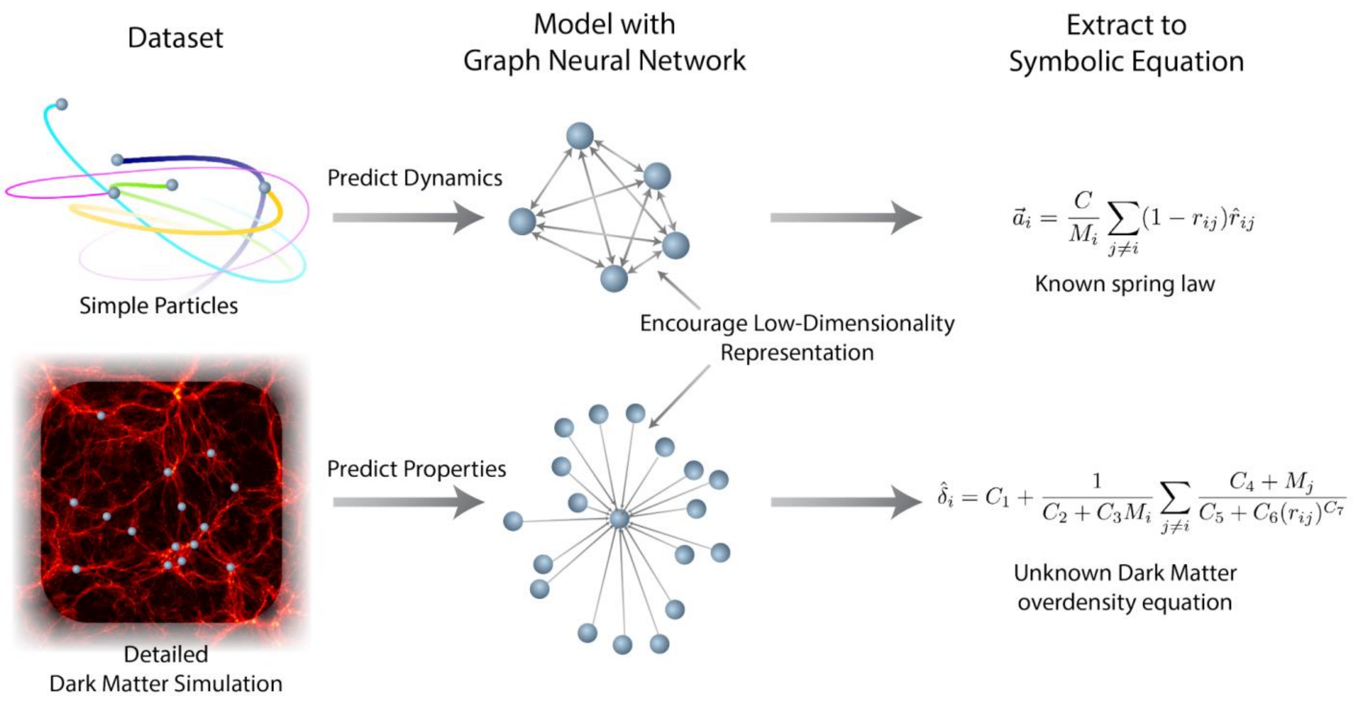
Applications to Complex Systems



# Future Work

- Improve interpretability with symbolic regression [Brunton, 2016]

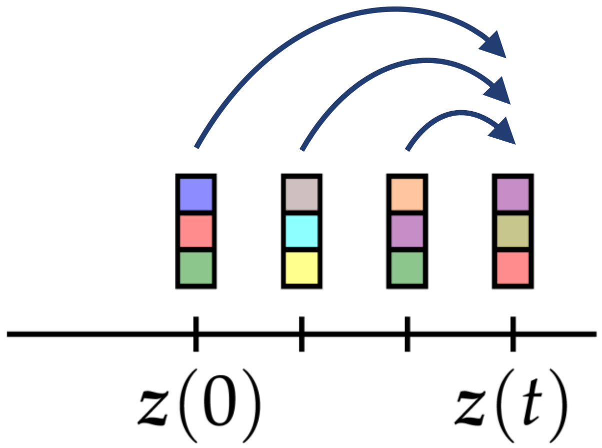



*[Cramner, 2020]*

# Future Work

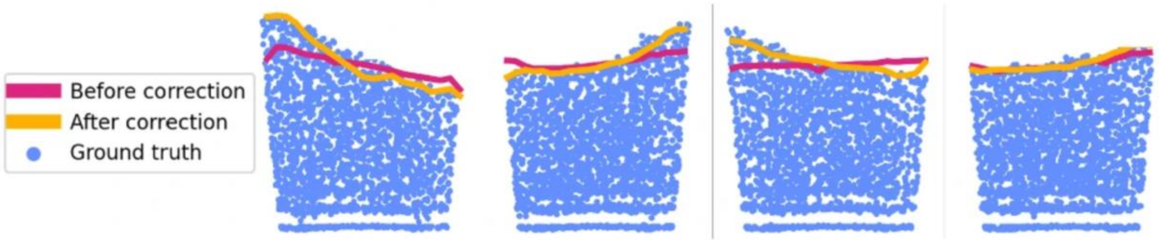
- Improve interpretability with symbolic regression [Brunton, 2016]
- Higher order time discretization
  - Runge-Kutta [Wang, 1998]
  - Transformers [Vaswani, 2017]

$$\boldsymbol{z}_{t+\Delta t} = \boldsymbol{z}_t + \Delta t \sum_i c_i \boldsymbol{k}_i$$



# Future Work

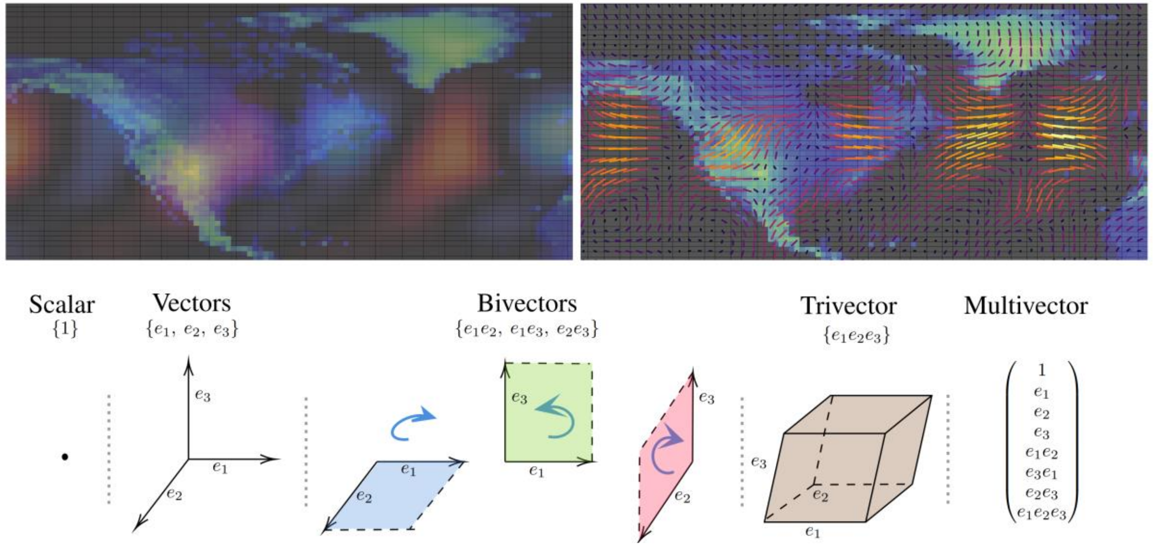
- Improve interpretability with symbolic regression [Brunton, 2016]
- Higher order time discretization
  - Runge-Kutta [Wang, 1998]
  - Transformers [Vaswani, 2017]
- Use experimental data
  - Perturbation + Noise models
  - Incremental learning [Moya, 2023]




*[Moya, 2023]*

# Future Work

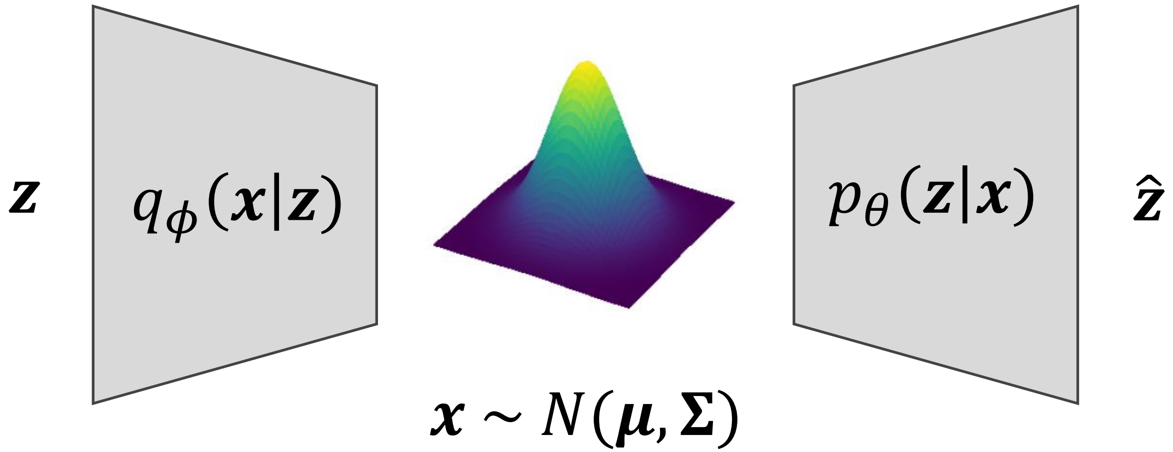
- Improve interpretability with symbolic regression [Brunton, 2016]
- Higher order time discretization
  - Runge-Kutta [Wang, 1998]
  - Transformers [Vaswani, 2017]
- Use experimental data
  - Perturbation + Noise models
  - Incremental learning [Moya, 2023]
- Equivariant architectures
  - Exterior algebra [Brandstetter, 2023]




*[Brandstetter, 2023]*

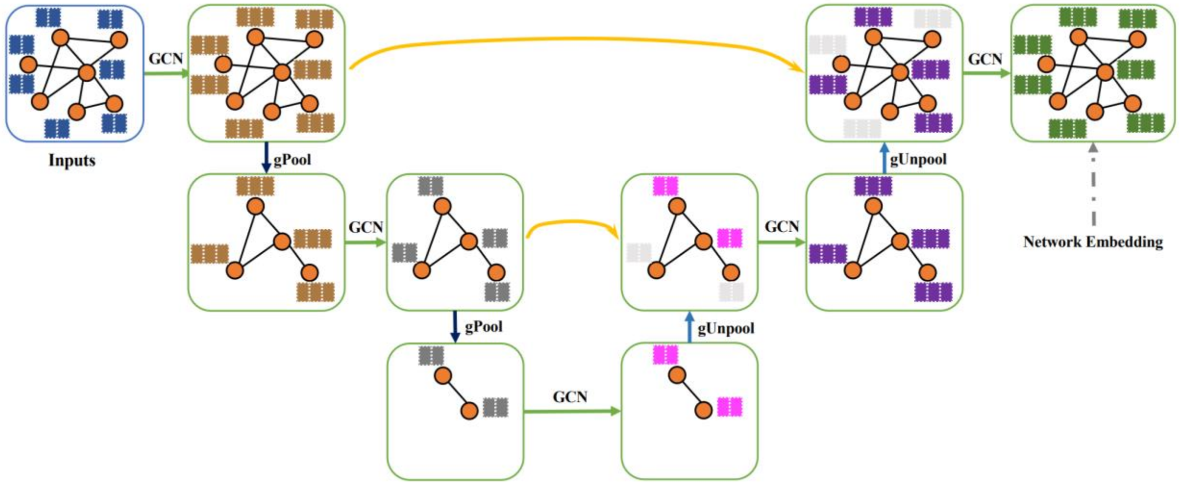
# Future Work

- Uncertainty quantification
  - Bayesian frameworks [Kingma, 2013]



# Future Work

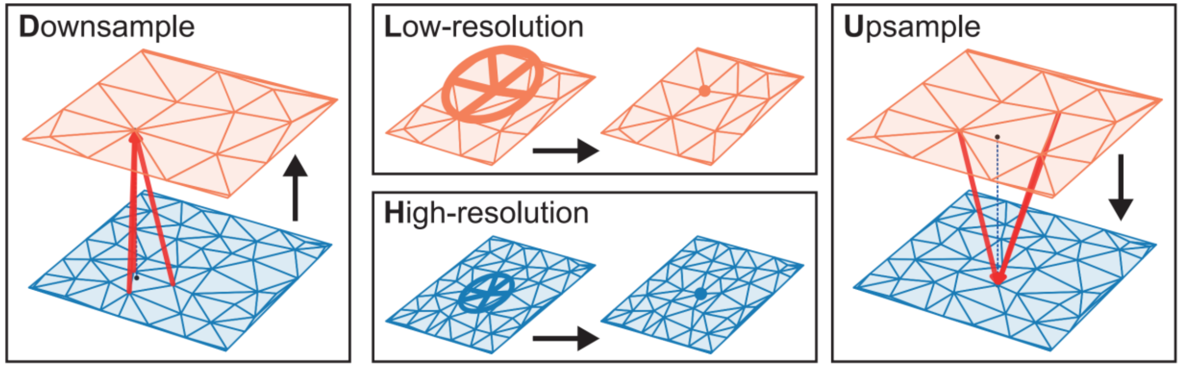
- Uncertainty quantification
  - Bayesian frameworks [Kingma, 2013]
- Architecture improvements
  - Graph U-nets [Gao, 2019]




*[Gao, 2019]*

# Future Work

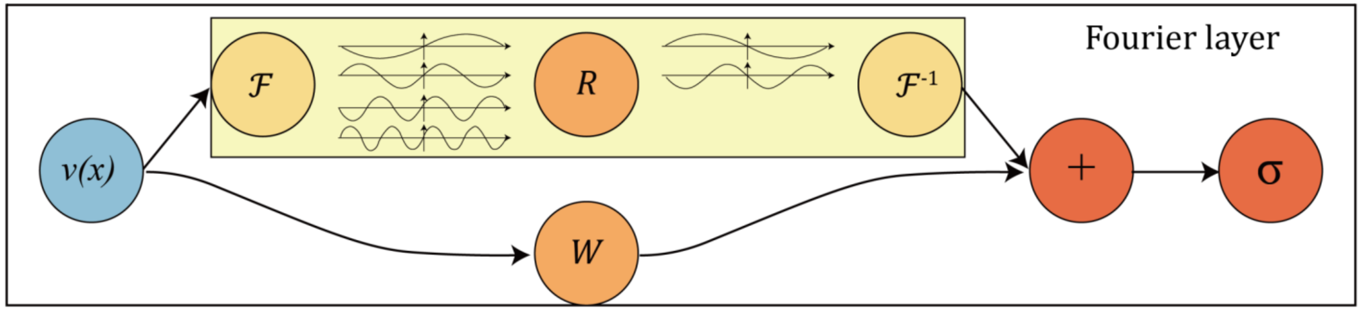
- Uncertainty quantification
  - Bayesian frameworks [Kingma, 2013]
- Architecture improvements
  - Graph U-nets [Gao, 2019]
  - Multiresolution message passing [Fortunato, 2022]




*[Fortunato, 2022]*

# Future Work

- Uncertainty quantification
  - Bayesian frameworks [Kingma, 2013]
- Architecture improvements
  - Graph U-nets [Gao, 2019]
  - Multiresolution message passing [Fortunato, 2022]
  - Frequency-based encoding [Li, 2020]




*[Li, 2020]*

# Future Work

- AR experience: VR headset + haptic gloves



# Code

- All datasets and pre-trained nets available on Github
  - `https://github.com/quercushernandez`

**StructurePreservingNN** Public

Code for the paper "Structure-preserving neural networks" published in Journal of Computational Physics (JCP).

Python ☆ 15 ⑂ 3

**ThermodynamicsGNN** Public

Code for the paper "Thermodynamics-informed graph neural networks" published in IEEE Transactions on Artificial Intelligence (TAI).

Python ☆ 65 ⑂ 14

**DeepLearningMOR** Public

Code for the paper "Deep learning of thermodynamics-aware reduced-order models from data" published in Computer Methods in Applied Mechanics and Engineering (CMAME).

Python ☆ 12 ⑂ 2

# Acknowledgements


- This work has been partially funded by the **Spanish Ministry of Science and Innovation**, AEI /10,13039/501100011033, through Grant number PID2020-113463RB-C31.
- The authors also acknowledge the support of **ESI Group** through the project UZ-2019-0060.
- This material is based upon work supported in part by the **Army Research Laboratory** and the **Army Research Office** under contract/grant number W911NF2210271.

# Structure-preserving machine learning for dissipative systems: methods and applications

**SciML Workshop CWI**

**December 8th**


**Quercus Hernández**

**quercushernandez@gmail.com**

@QuercusEtAl

github.com/quercushernandez

github.quercushernandez.io

Universidad Zaragoza