Physically informed neural networks for surrogate models of gravitational wave signal

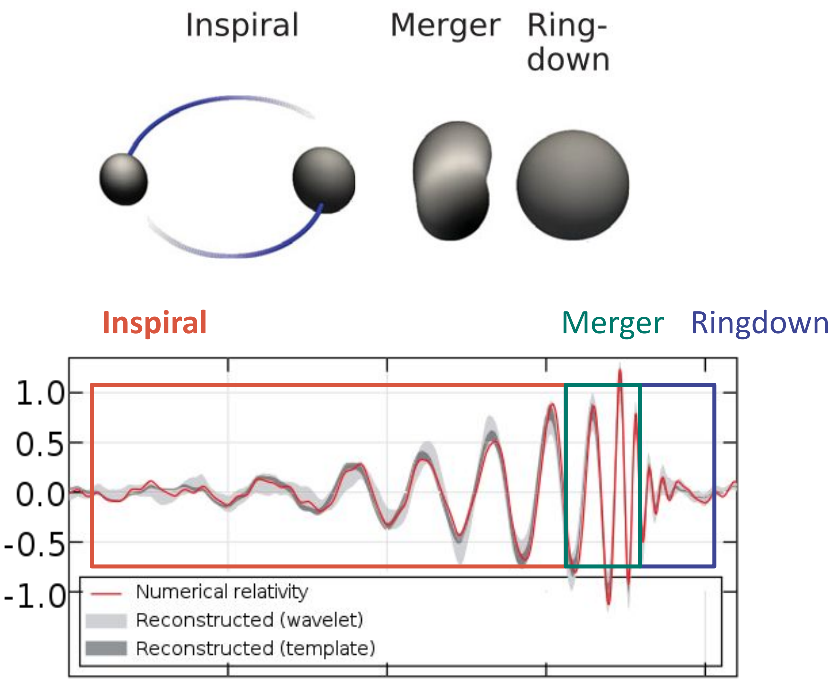
ANR Ricochet - Lille - 13/06/2024

Thomas Barreira, Pierre Palud
Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France

Table of contents

Disclaimer: work in progress

Gravitational wave models and uses



3 approaches in modelling :

- Numerical relativity
- Phenomenological
- **Effective One Body (EOB)**

Numerical solvers: slow for parameter inference

→ Use of PINNs

For now: model the BH dynamics

- major part for the GW generation
- simple models. Precession and consequences on polarization will come next

EOB : center of mass frame

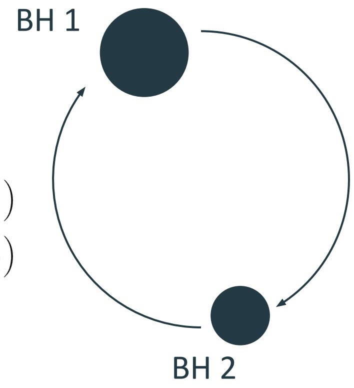
Inspiral phase

m_1

m_2

(q_1, p_1)

(q_2, p_2)



Trajectory of an effective body in the center mass frame

$$M = m_1 + m_2$$

$$\mu = m_1 m_2 / M$$

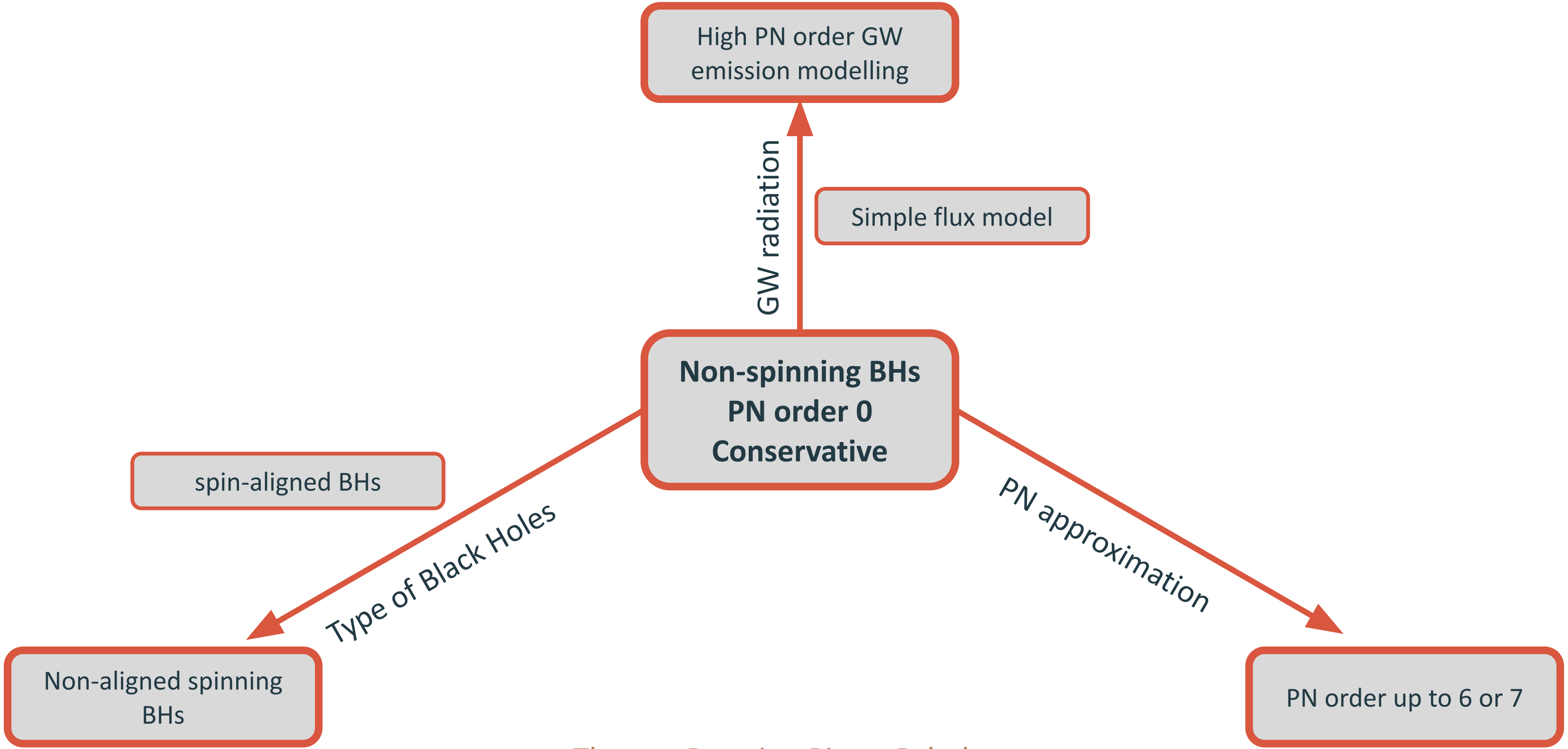
$$(r, \varphi, p_r, p_\varphi)$$

Equivalent metric leads to a Hamiltonian

$$g_{\mu\nu} \rightarrow H(q, p, t)$$

Thomas Barreira, Pierre Palud

Hamiltonian definition: levels of complexity



First simple Hamiltonian

Hamiltonian for **non-spinning BHs**

$$H = \sqrt{\left(1 - \frac{2GM}{r}\right)\left(\mu^2 + \left(1 - \frac{2GM}{r}\right)p_r^2 + \frac{p_\varphi^2}{r^2}\right)},$$

total mass and reduced mass

with $M = m_1 + m_2$ and $\mu = m_1 m_2 / M$

radius

$$\dot{r} = \frac{\partial H}{\partial p_r}$$

phase

$$\dot{\varphi} = \frac{\partial H}{\partial p_\varphi}$$

conjugate moment of radius

$$\dot{p_r} = -\frac{\partial H}{\partial r}$$

conjugate moment of phase

$$\dot{p_\varphi} = -\frac{\partial H}{\partial \varphi}$$

First simple Hamiltonian (with adimensionalisation)

Hamiltonian for **non-spinning BHs**

$$\tilde{H} = \sqrt{\left(1-\frac{1}{\tilde{r}}\right)\left(1+\left(1-\frac{1}{\tilde{r}}\right)\tilde{p_r}^2+\frac{\tilde{p_\varphi}^2}{\tilde{r}^2}\right)}$$

radius

$$\dot{\tilde{r}} = \frac{\partial \tilde{H}}{\partial \tilde{p_r}}$$

phase

$$\dot{\tilde{\varphi}} = \frac{\partial \tilde{H}}{\partial \tilde{p_\varphi}}$$

conjugate moment of radius

$$\dot{\tilde{p_r}} = -\frac{\partial \tilde{H}}{\partial \tilde{r}}$$

conjugate moment of phase

$$\dot{\tilde{p_\varphi}} = -\frac{\partial \tilde{H}}{\partial \tilde{\varphi}}$$

Using

$$\tilde{r} = \frac{r}{2M}$$

$$\tilde{t} = \frac{t}{2M}$$

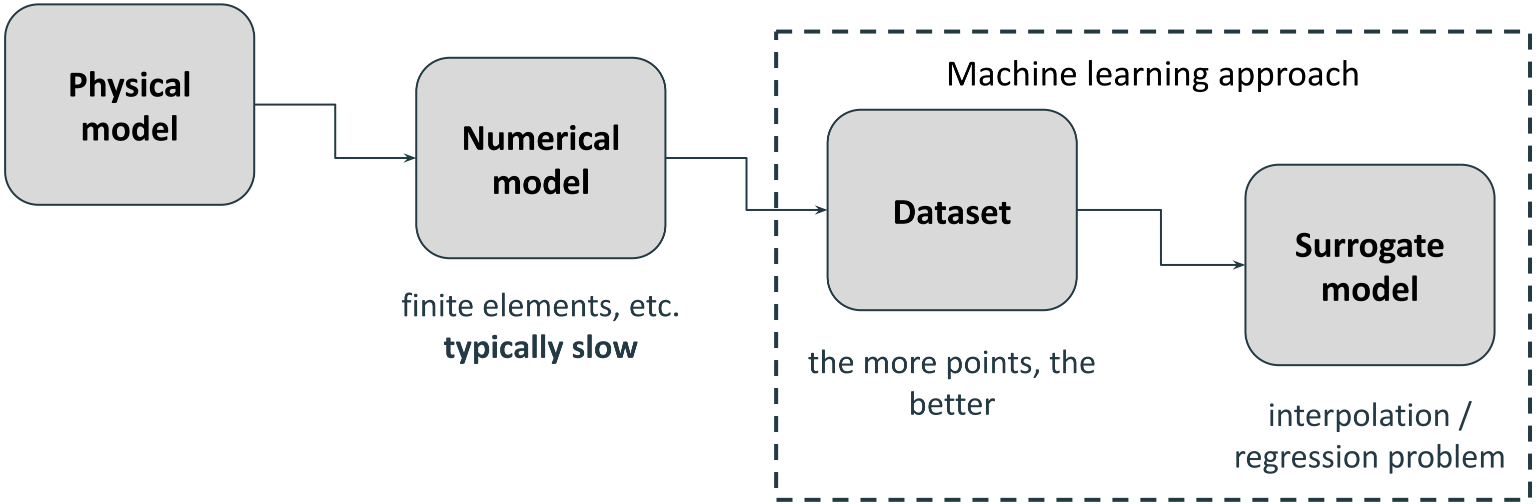
$$\tilde{\varphi} = \varphi$$

$$\tilde{p_r} = \frac{p_r}{\mu}$$

$$\tilde{H} = \frac{H}{\mu}$$

$$\tilde{p_\varphi} = \frac{p_\varphi}{2M\mu}$$

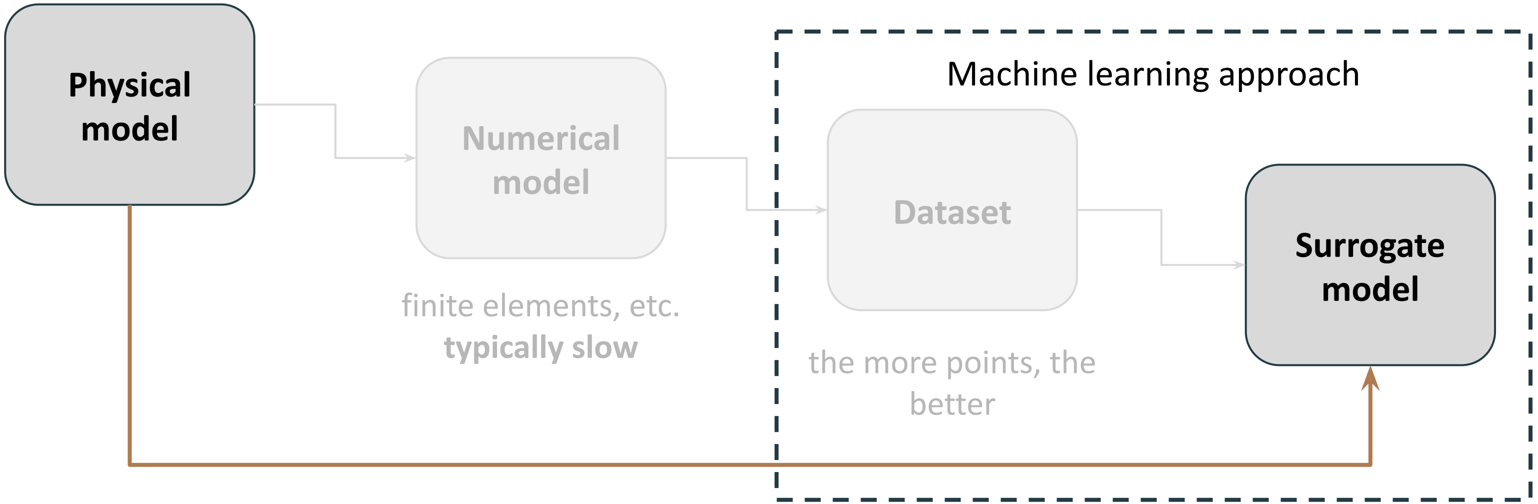
Context: numerical models and surrogates



For efficient surrogate models:
exploitation of physics knowledge

e.g., Schmidt, Cano, Palud

Context: Physics informed neural networks



Proposition with PINNs: derive a surrogate model directly from the physical model

For efficient surrogate models: **exploitation of physics knowledge**

e.g., Schmidt, Cano, Palud

Physics-informed neural networks

initial conditions

time ← ↑ → PDE parameters

} defined on cartesian product of intervals

Goal: find $f_\Psi : t, v, \lambda \mapsto f_\Psi(t, v, \lambda)$ parametrized by a vector Ψ s.t.

$$\begin{aligned} (PDE) \quad & E_\lambda (f_\Psi) = 0 \\ (IC) \quad & \forall v, \lambda,\ D_v [f_\Psi](0, v, \lambda) = 0 \end{aligned}$$

with E_λ and D_v known differential operators

Example: damped harmonic oscillator

$$\begin{aligned} (PDE) \quad & & \ddot{f}_\Psi + \lambda_1 \dot{f}_\Psi + \lambda_2 f_\Psi = 0 \\ (IC) \quad & \forall v, \lambda, & f_\Psi(0, v, \lambda) = v_1 \\ & & \dot{f}_\Psi(0, v, \lambda) = v_2 \end{aligned}$$

Physics-informed neural networks

How? Convert equations into loss function terms evaluated on finite datasets

Generate two datasets: $\mathcal{D}_{\text{PDE}} = \left(t^{(i)}, v^{(i)}, \lambda^{(i)}\right)_{i=1}^{N_{\text{PDE}}}$ and $\mathcal{D}_{\text{IC}} = \left(0, v^{(i)}, \lambda^{(i)}\right)_{i=1}^{N_{\text{IC}}}$

Loss term on PDE

$$\mathcal{L}_{\text{PDE}}(\mathbf{\Psi}; \mathcal{D}_{\text{PDE}}) = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left\| E_{\lambda^{(i)}} [f_{\mathbf{\Psi}}] \left(t^{(i)}, v^{(i)}, \lambda^{(i)}\right) \right\|^2$$

Loss term on IC

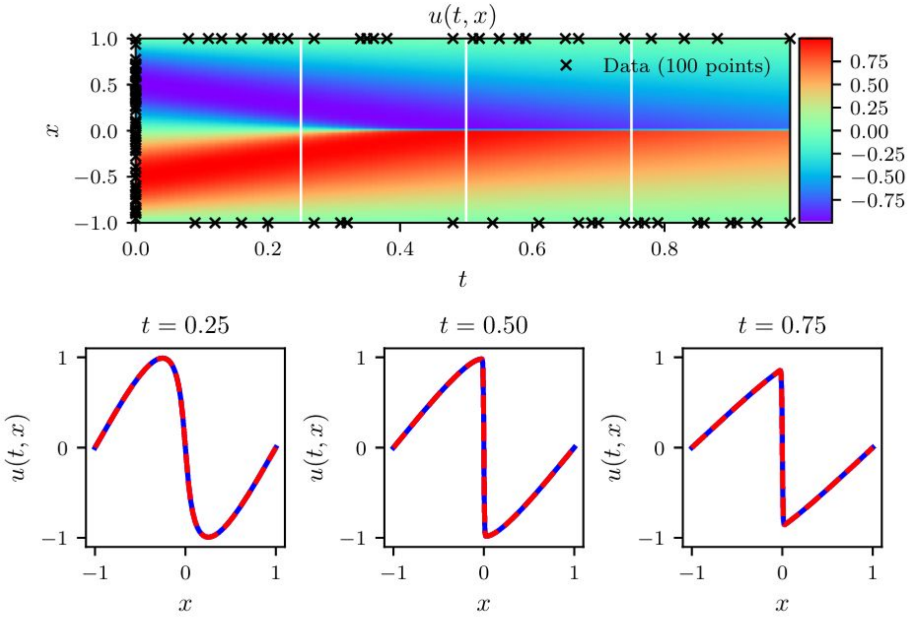
$$\mathcal{L}_{\text{IC}}(\mathbf{\Psi}; \mathcal{D}_{\text{IC}}) = \frac{1}{N_{\text{IC}}} \sum_{i=1}^{N_{\text{IC}}} \left\| D_{v^{(i)}} [f_{\mathbf{\Psi}}] \left(0, v^{(i)}, \lambda^{(i)}\right) \right\|^2$$

Global Loss and optimisation problem

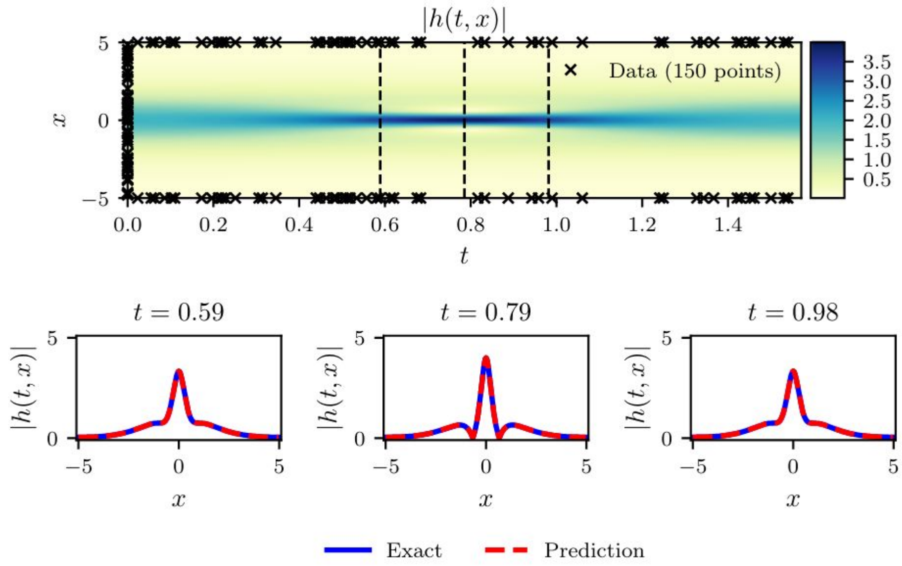
$$\widehat{\mathbf{\Psi}} \in \arg\min_{\mathbf{\Psi}} \ \mathbb{E}_{\mathcal{D}_{\text{PDE}}, \mathcal{D}_{\text{IC}}} \left[w_{\text{PDE}} \mathcal{L}_{\text{PDE}}(\mathbf{\Psi}; \mathcal{D}_{\text{PDE}}) + w_{\text{IC}} \mathcal{L}_{\text{IC}}(\mathbf{\Psi}; \mathcal{D}_{\text{IC}}) \right]$$

Why neural networks? rich class of functions + automatic differentiation

Physics-informed neural networks



PINN applied to a hydrodynamics PDE (Burger's eq.).
Figure from Raissi et al. (2017).



PINN applied to The Schrödinger equation.
Figure from Raissi et al. (2017).

Easier problem than introduced. Introduce a spatial variable,
but **solves for fixed initial conditions and eq. parameters**

Physics-informed neural networks

Architecture

- Fourier layers (Random, sine activation, etc.)
- Random Weight Factorization
- Output structuring
- ResNet / DenseNet / standard feedforward

Training

- Causal loss
- Loss balancing
- Curriculum learning
- Self-adapted
- Pre-training / use of solver data

Others

- Equation adimensionalisation

Physics-informed neural networks

Architecture

- Fourier layers (Random, sine activation, etc.)
- Random Weight Factorization
- Output structuring
- ResNet / DenseNet / standard feedforward

Training

- Causal loss
- Loss balancing
- Curriculum learning
- Self-adapted
- Pre-training / use of solver data

Others

- Equation adimensionalisation

Causal loss and Fourier features

Fourier layers (Random, sine activation, etc.)

How to efficiently learn oscillations?

for at least first layer: use a sine activation function

questions (each addressed in the literature) :

- initialize weights at random, with which distribution?
- train weights or not

Causal loss

How to enforce first learning for low t, close to IC ?

Rewrite loss PDE with time-dependent weights

$$\mathcal{L}_{\mathrm{PDE}}(\boldsymbol{\Psi}; \mathcal{D}_{\mathrm{PDE}}) = \frac{1}{N_{\mathrm{PDE}}} \sum_{i=1}^{N_{\mathrm{PDE}}} w_i \left\| E_{\lambda^{(i)}} [f_{\boldsymbol{\Psi}}] \left(t^{(i)}, v^{(i)}, \lambda^{(i)}\right) \right\|^2$$

with

$$w_i = \exp\left[-\epsilon \sum_{t^{(j)} < t^{(i)}} \left\| E_{\lambda^{(j)}} [f_{\boldsymbol{\Psi}}] \left(t^{(j)}, v^{(j)}, \lambda^{(j)}\right) \right\|^2 \right]$$

Thomas Barreira, Pierre Palud

Application: damped harmonic oscillator

$$\ddot{y} + \beta \dot{y} + ky = 0$$
$$y(0) = y_0, \dot{y}(0) = \dot{y}_0$$

Neural network 0 :

Input time only

Neural network 1 :

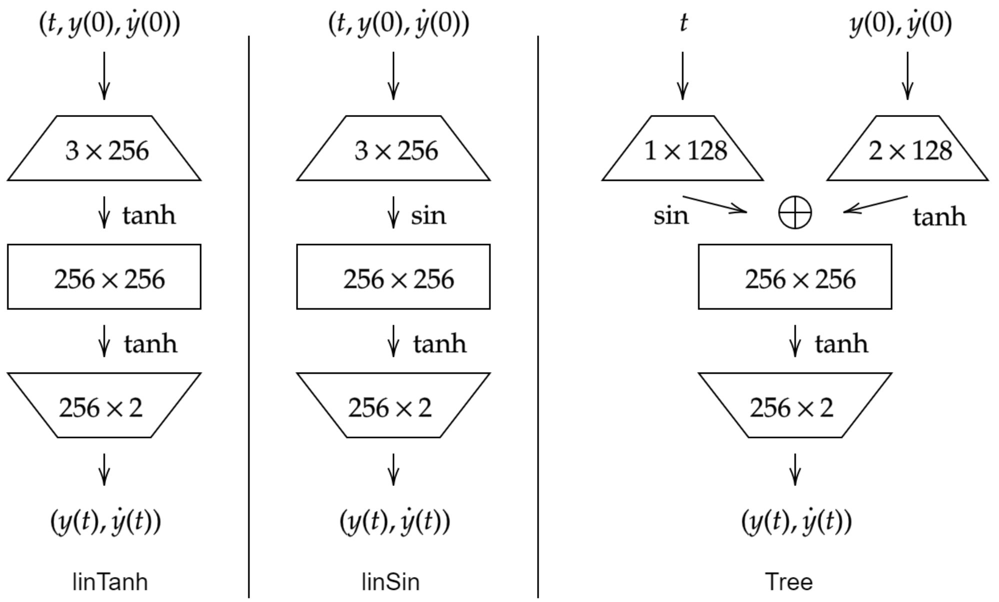
Fixed parameters (k = 16, beta = 0,1,4)
Input initial conditions (from -1 to 1)

Neural network 2 :

Input parameters (k = 10 to 16, beta = 0 to 4)
Fixed initial conditions (y(0) = 1, yp(0) = 0)

Application: damped harmonic oscillator

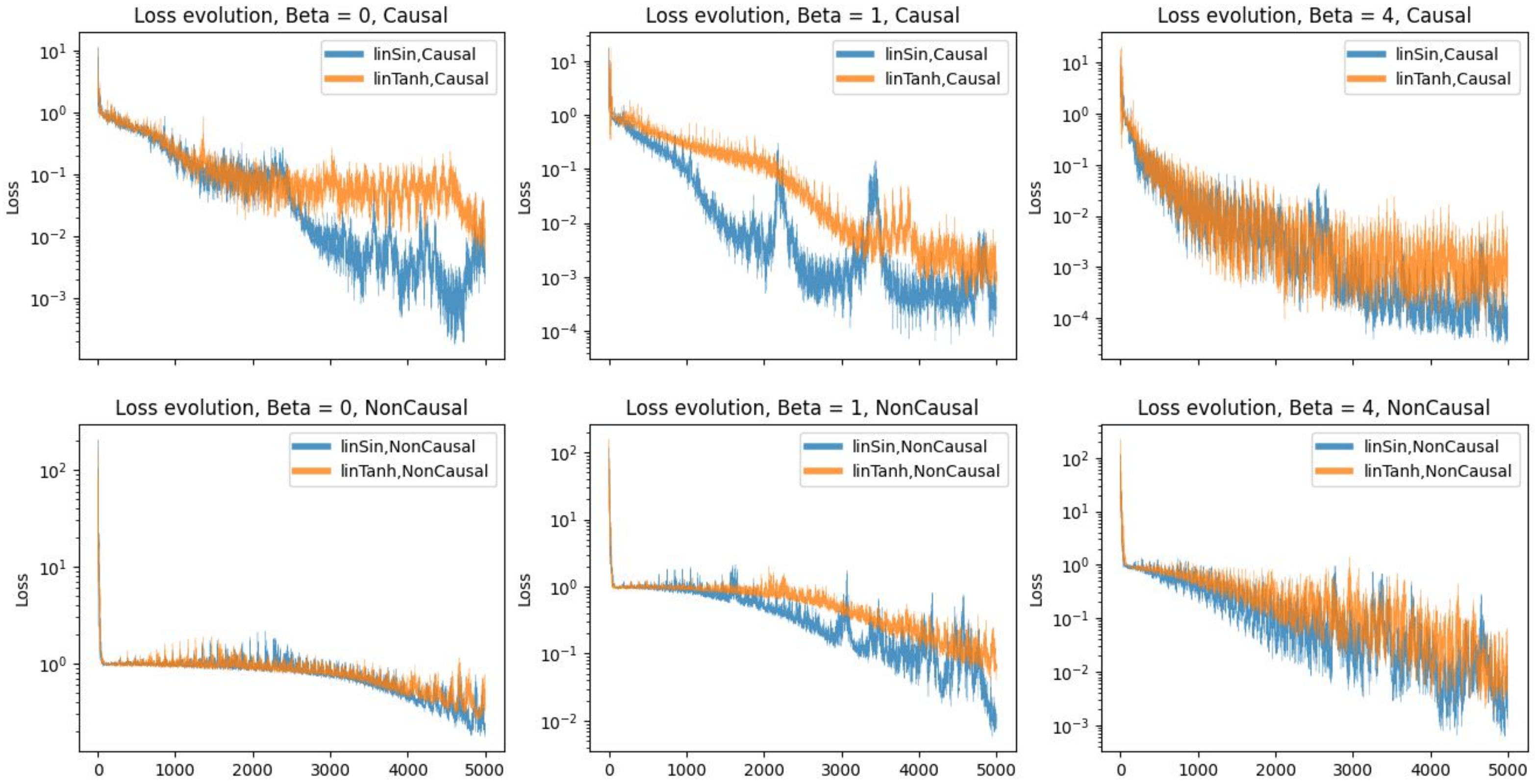
3 architectures



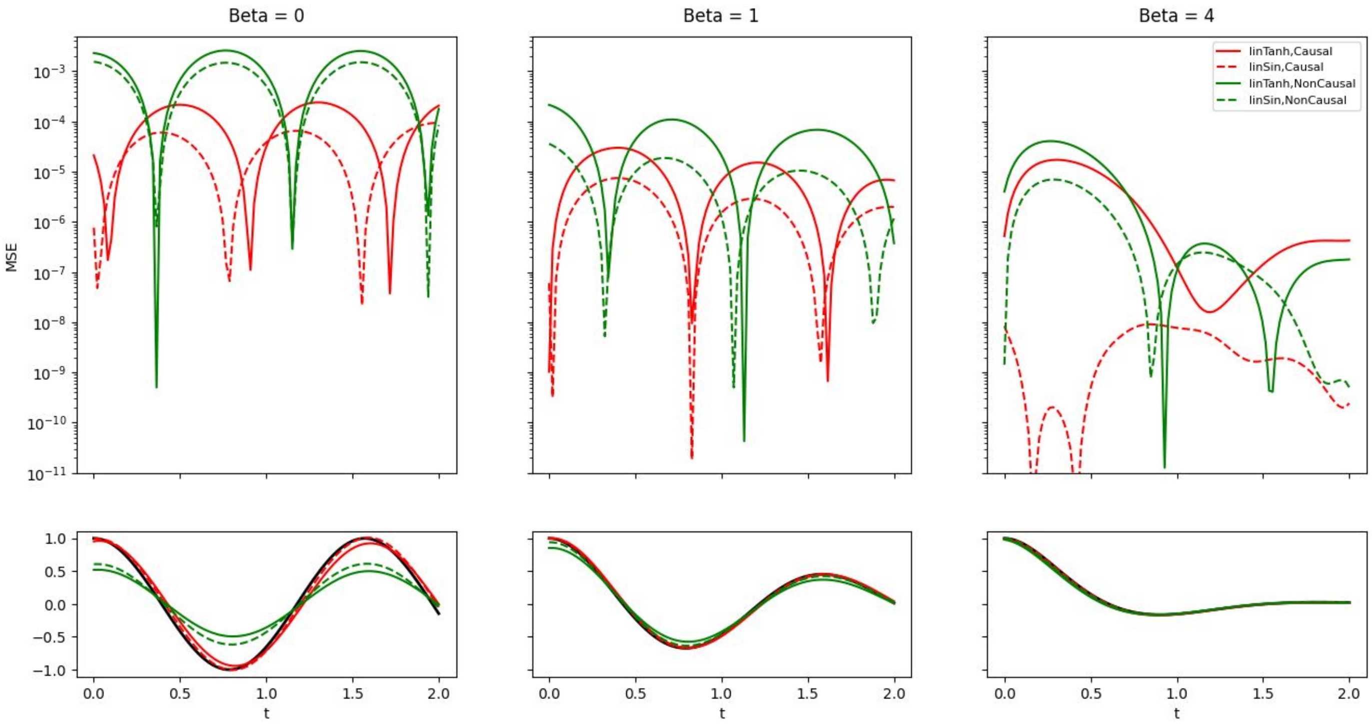
2 learning strategies

causal loss VS non-causal loss

Application 0 : input t only, Losses



Application 0 : input t only, squared error



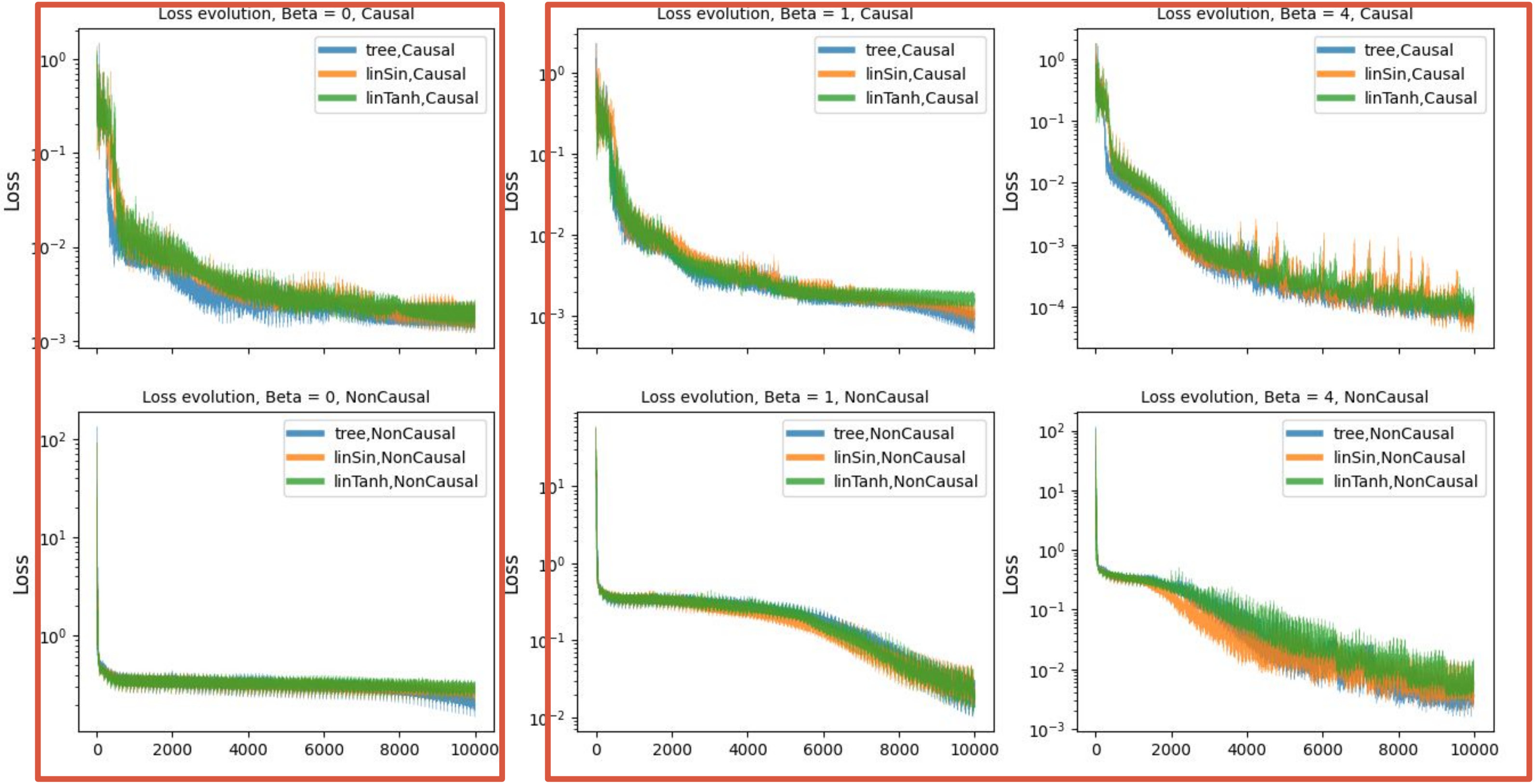
Beta small = harder

Causal >> NonCausal

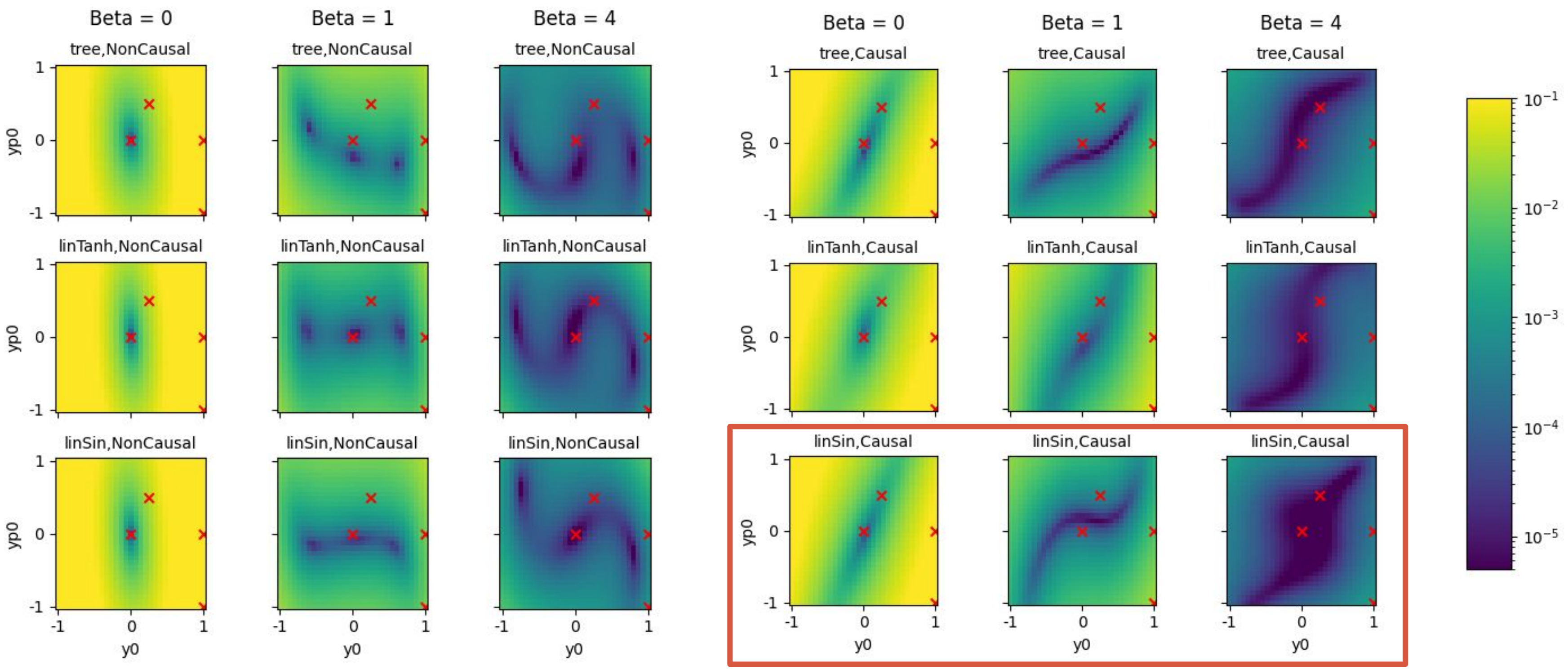
using Causal: critical for small beta.

LinSin > LinTanh

Application 1 : input (t,ic), Losses



Application 1 : input (t,ic), MSE(ic)



Application 1 : input (t,ic), MSE(t)

averaged over ic

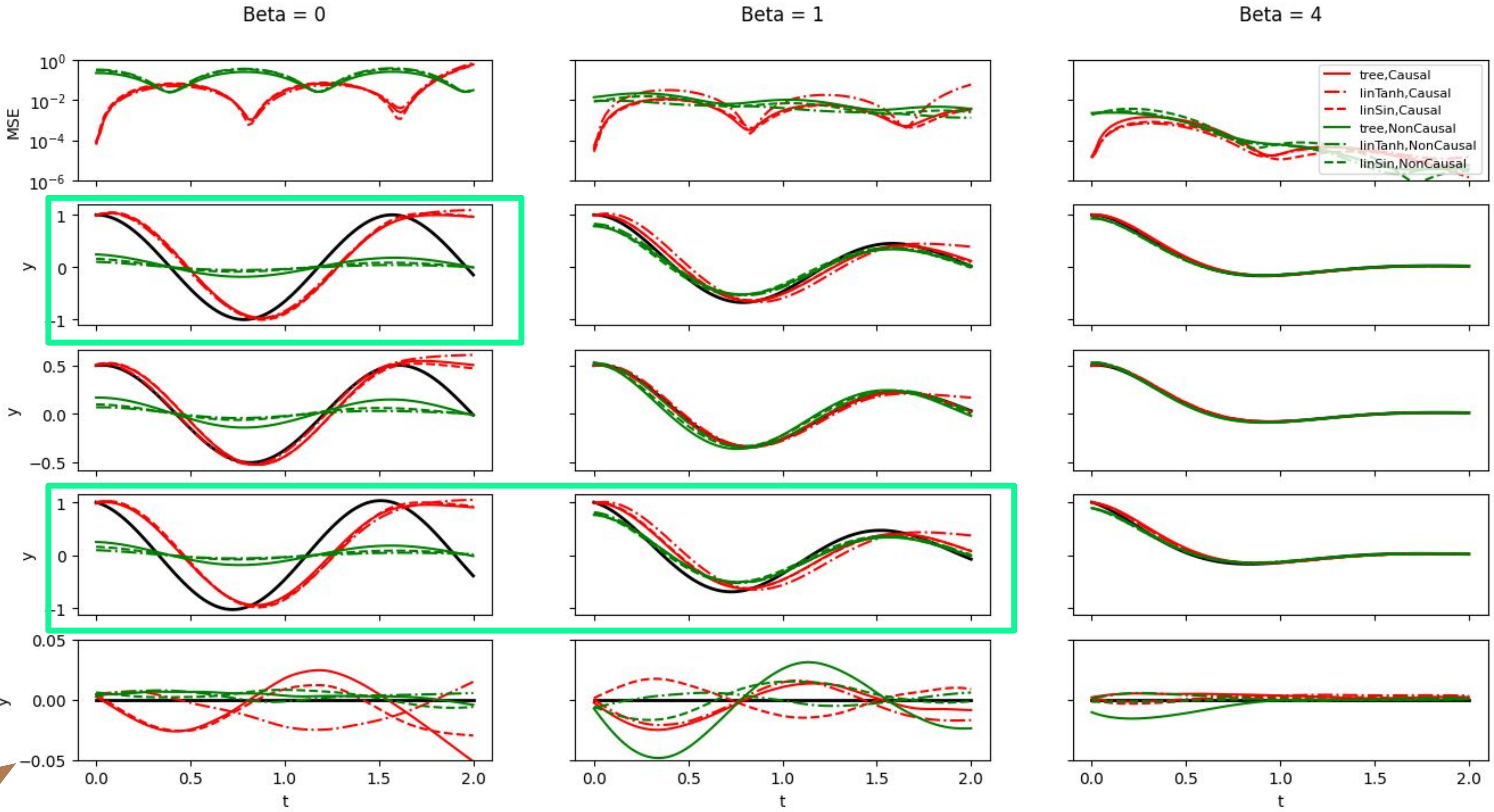
illustration on some specific ic

(1, 0)

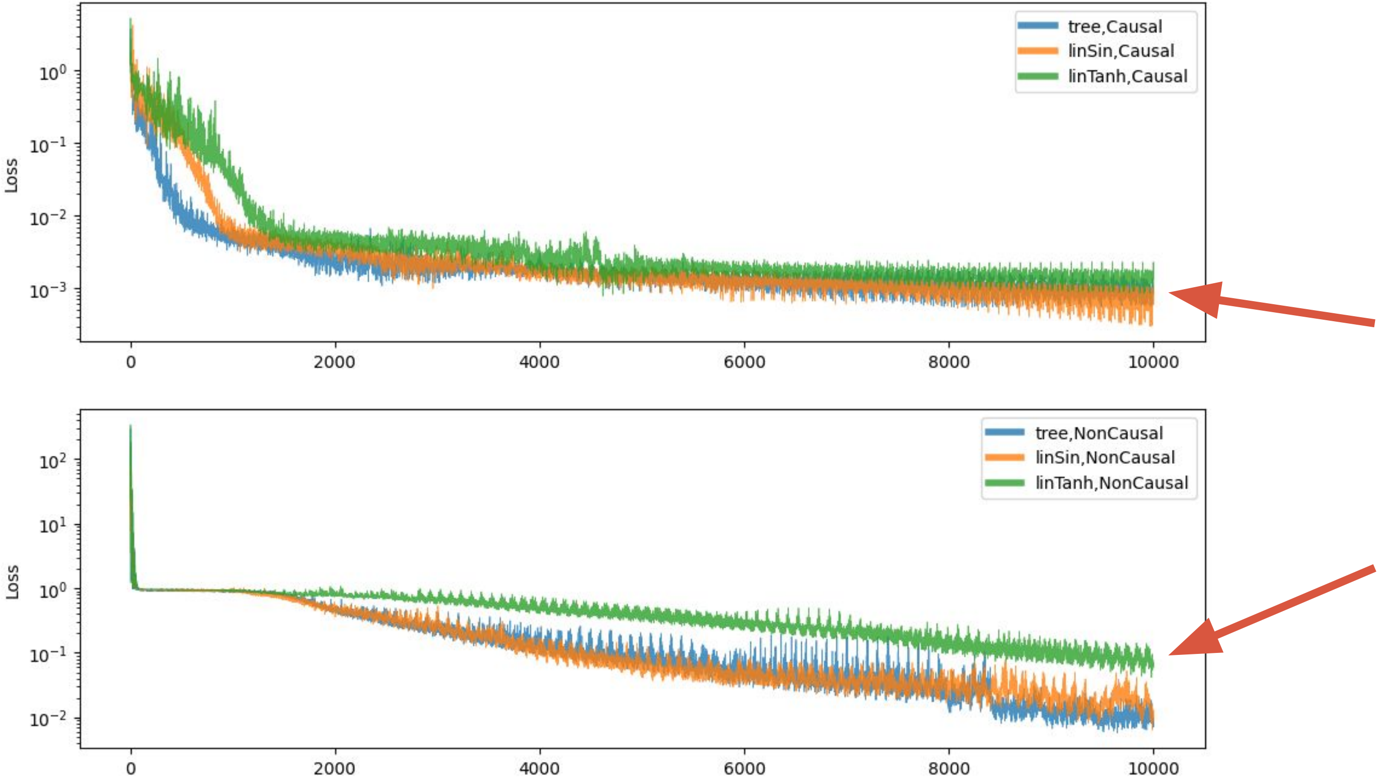
(0.5, 0.25)

(1, -1)

(0, 0)

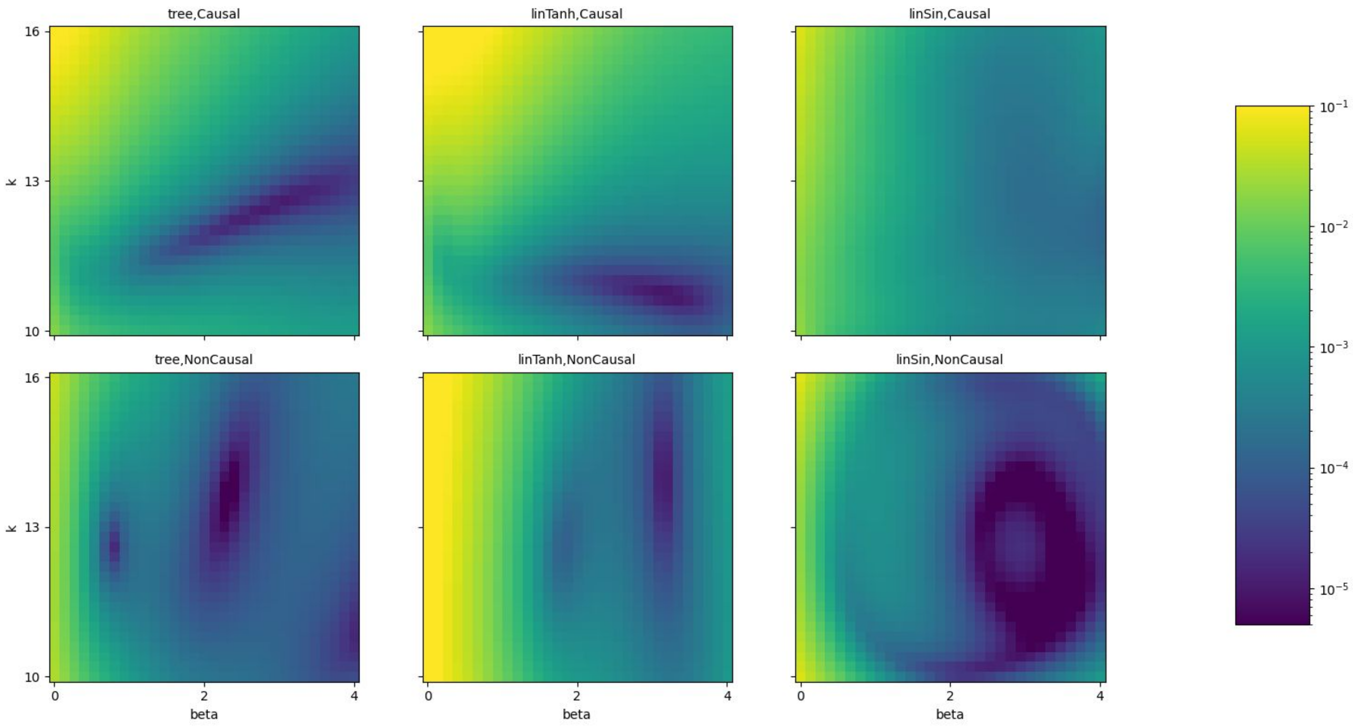


Application 2 : input (t, param) Losses



linTanh less efficient

Application 2 : input (t, param) MSE(param)



Beta small = harder

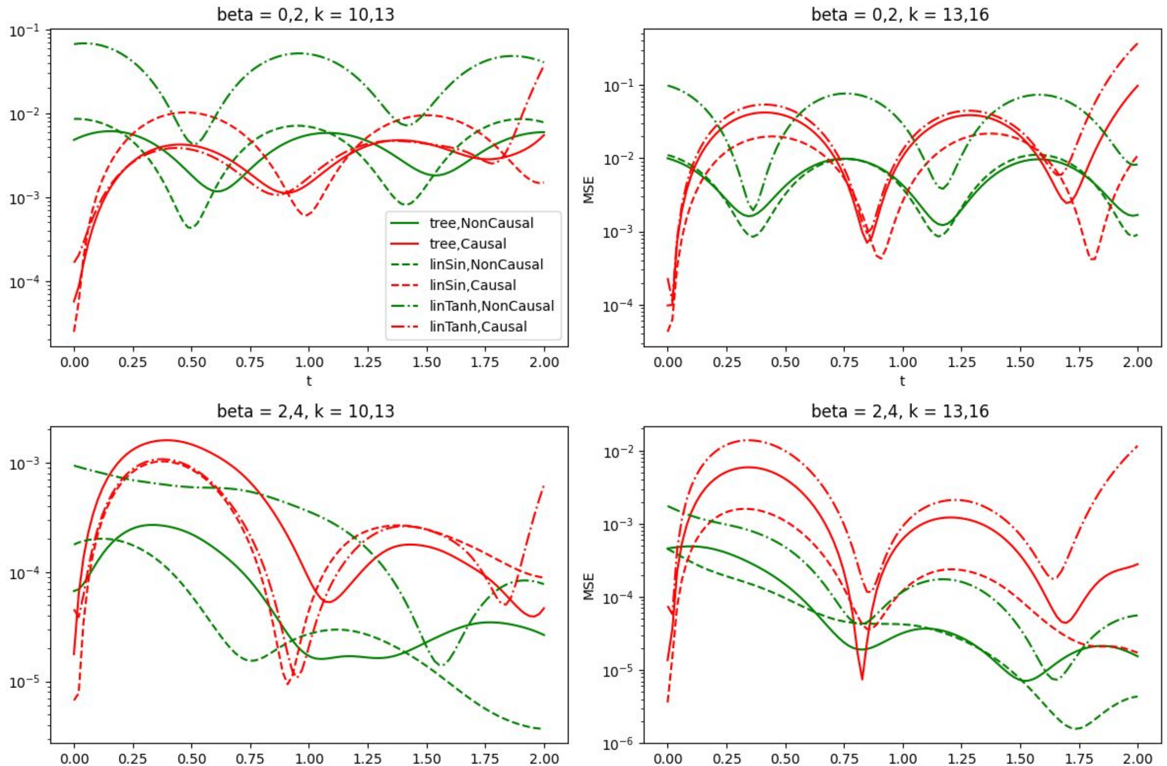
k big = harder only for Causal

NonCausal independent of k

Causal not better than NonCausal !

LinTanh worse

Application 2 : input (t, param) MSE(t)



Conclusions

Overall regarding PINNs:

- In SciML community: big hype, but **application not straightforward**. Lots of handcrafting and tests.
- But PINNs struggle to capture **high frequency oscillations**.
- Currently: multiple papers with several methods, hard to keep up with literature

In our project for GW generation:

- difficulties with simple cases
- requires further study of PINNs, future steps on Hamiltonian are ready.

Thank you for your attention