



Universität
Bremen

Center for Industrial
Mathematics (ZeTeM)

Faculty 03

Mathematics / Computer science

Introduction to TorchPhysics

Parameter studies and inverse problems

Nick Heilenkötter, Janek Gödeke, Tom
Freudenberg
Heidelberg, 08.11.2023

Parameter Studies and Parameter Identification

Realization with PINN and TorchPhysics

Parameter-dependent wave equation:

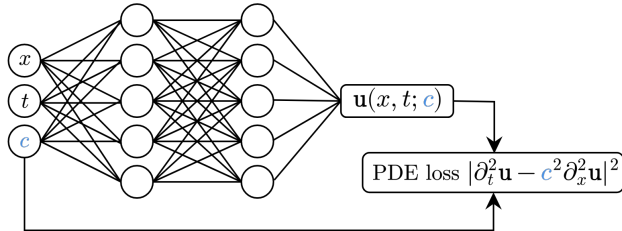
$$\begin{cases} \partial_t^2 u = c \partial_x^2 u, & \text{in } I_x \times I_t, \\ u = 0 & \text{in } \partial I_x \times I_t, \\ \partial_t u(\cdot, 0) = 0 & \text{in } I_x, \\ u(\cdot, 0) = \sin(x) & \text{in } I_x, \end{cases}$$

many industrial applications involve:

- **Parameter studies:** Solving the same PDE for many different choices of c
- **Parameter identification:** Finding the c that leads to given solution data $\{\hat{u}_i\}$

Parameter Studies with PINNs

Solving the same PDE for many different choices of c

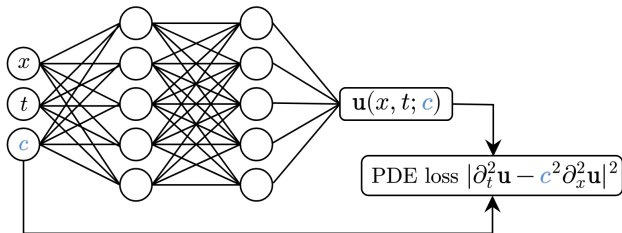


Method:

- Include parameter(s) as additional input(s) to the PINN
- Training: Sample parameter range together with function domain

Parameter Studies with PINNs

Solving the same PDE for many different choices of c



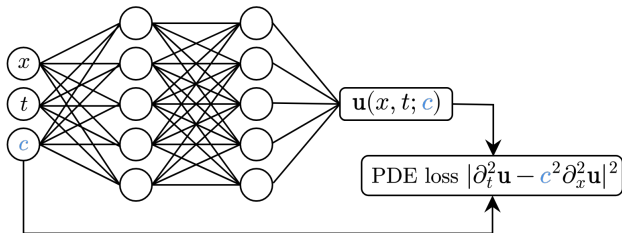
Method:

- Include parameter(s) as additional input(s) to the PINN
- Training: Sample parameter range together with function domain

```
1 model = tp.models.FCN(input_space=X*T*C, output_space=U)
2
3 def pde_residual(u, t, x, c):
4     return ...
```

Parameter Studies with PINNs

Solving the same PDE for many different choices of c

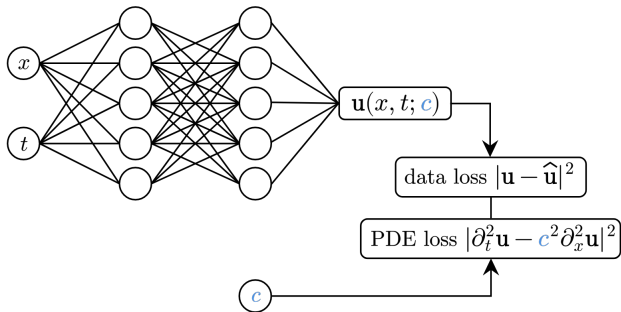


Result:

- Inference of solution for new parameter by a forward pass to the trained network
- Very little additional effort in evaluation of the network
- Increased amount of training points necessary

Parameter Identification with PINNs

Finding the \mathbf{c} that leads to given solution data $\{\hat{u}_i\}$

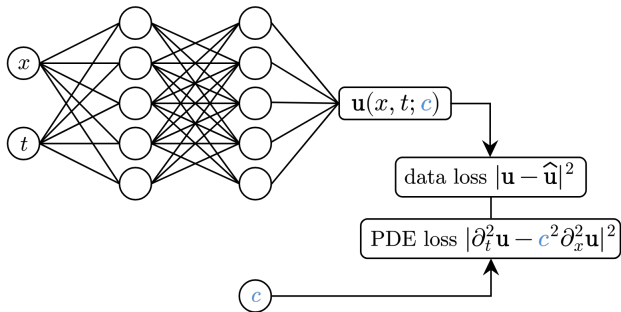


Method:

- Include parameter(s) as learnable parameter(s)
- Training: Incorporate data loss in training
 \rightsquigarrow Goal: Find a solution that fits data and solves PDE for the optimized parameter

Parameter Identification with PINNs

Finding the \mathbf{c} that leads to given solution data $\{\hat{u}_i\}$



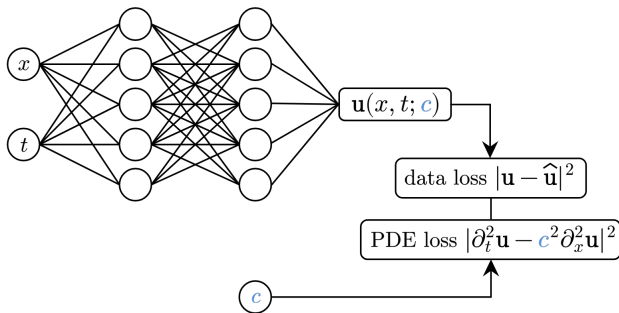
Method:

- Include parameter(s) as learnable parameter(s)
- Training: Incorporate data loss in training
 \rightsquigarrow Goal: Find a solution that fits data and solves PDE for the optimized parameter

```
1 Param_c = tp.models.Parameter(init=1.0, space=D)
2 pde_condition = tp.conditions.PINNCondition(... ,
3                                             parameter=param_D)
```

Parameter Identification with PINNs

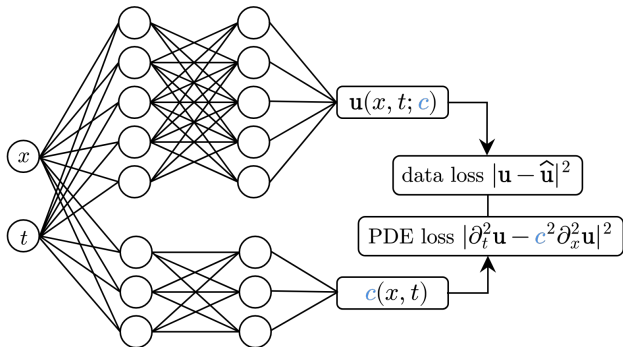
Finding the c that leads to given solution data $\{\hat{u}_i\}$



Result:

- Solution is learned in parallel to the parameter
- Effort: full NN training for one parameter

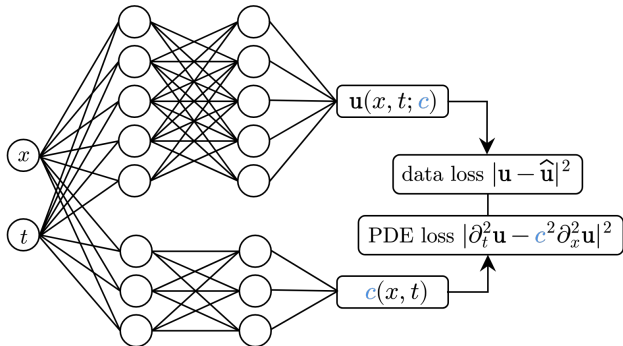
Function-Valued Parameter Identification



Method:

- Parameterize parameter(s) as NN
- Training: incorporate data loss in training

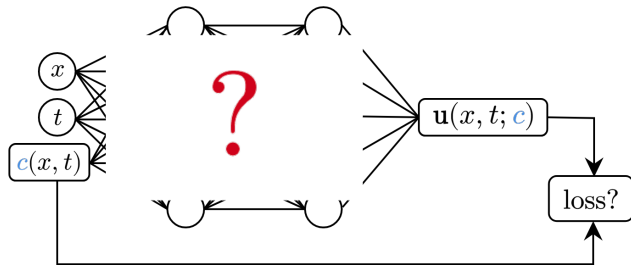
Function-Valued Parameter Identification



Result:

- Solution learned in parallel to the parameter function
- New parameter requires retraining

Function-Valued Parameter Studies



How to use a function-valued
parameter as an input to the NN?

~> Later: **DeepONets**

Exercises

$$\begin{cases} \partial_t^2 u = c \partial_x^2 u, & \text{in } I_x \times I_t, \\ u = 0 & \text{in } \partial I_x \times I_t, \\ \partial_t u(\cdot, 0) = 0 & \text{in } I_x, \\ u(\cdot, 0) = \sin(x) & \text{in } I_x, \end{cases}$$

where $I_x = [0, 2\pi]$ and $I_t = [0, 5]$.

Parameter Study (Example_6.ipynb)

- Learn the solution for multiple c

Inverse Problem (Example_7.ipynb)

- Determine c for given data (u_i, x_i, t_i)

