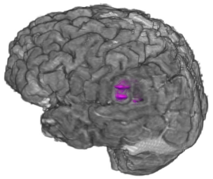
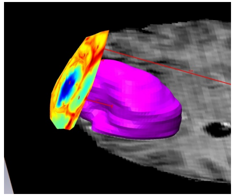
HELMHOLTZAI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

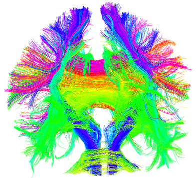
AI for Advanced Photon Sciences. A surrogate modelling perspective.

06/17/21

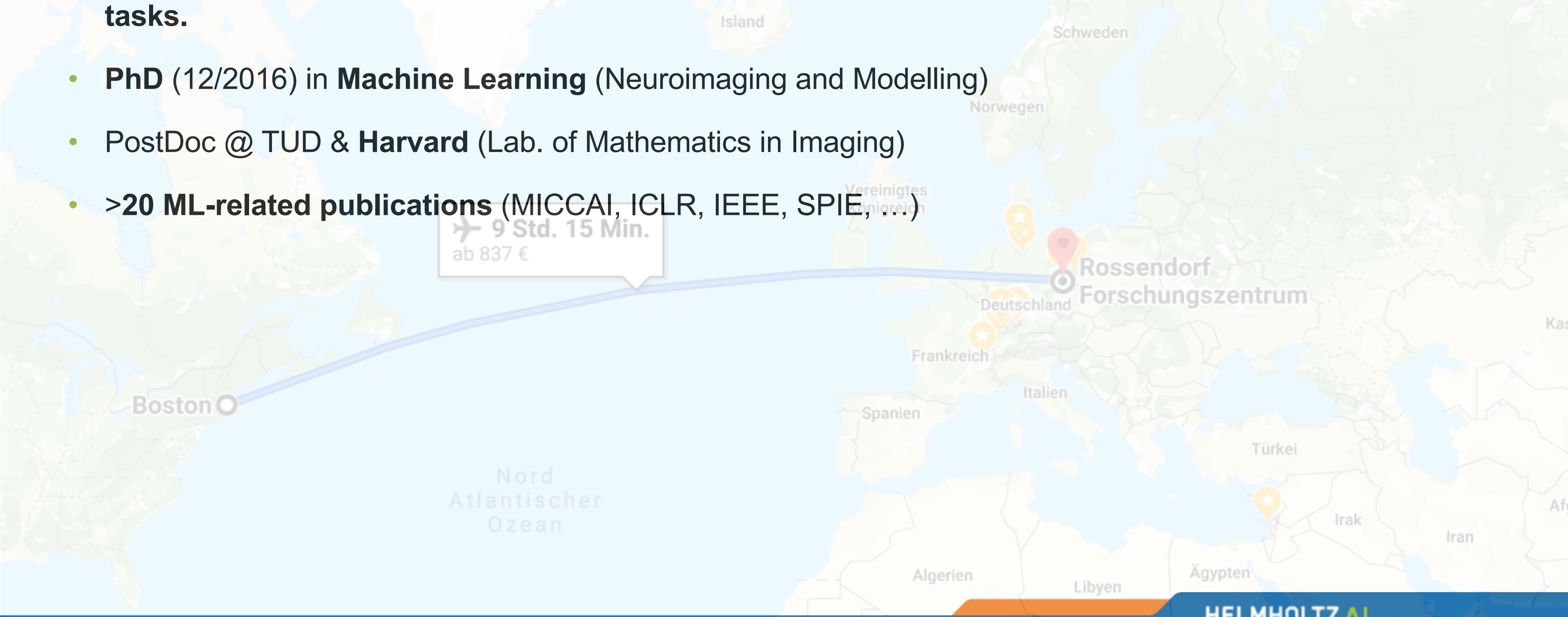
Nico Hoffmann et al.
n.hoffmann@hzdr.de
Helmholtz-Zentrum Dresden-Rossendorf



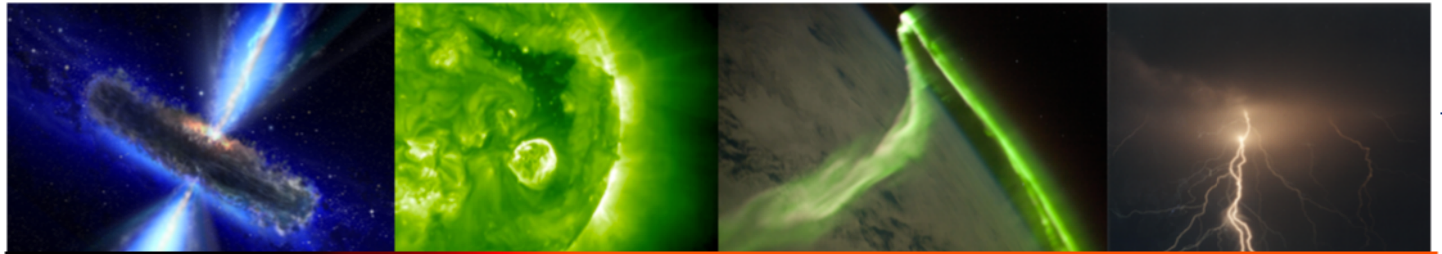


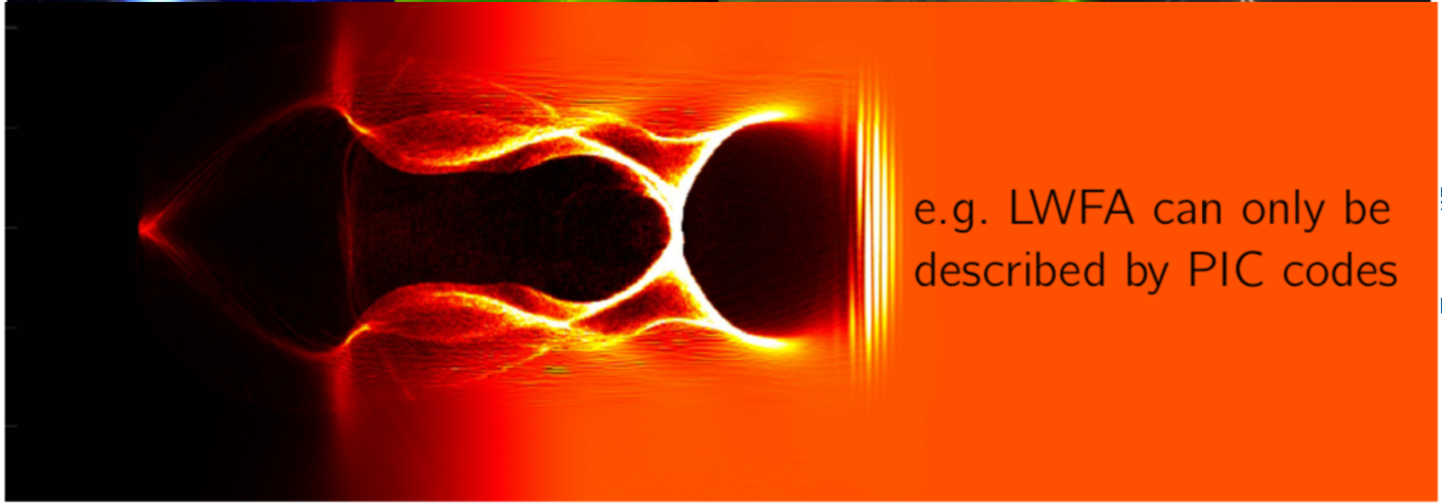


- **Application of machine learning in imaging for solving inverse problems and reconstruction tasks.**
- **PhD** (12/2016) in **Machine Learning** (Neuroimaging and Modelling)
- PostDoc @ TUD & **Harvard** (Lab. of Mathematics in Imaging)
- **>20 ML-related publications** (MICCAI, ICLR, IEEE, SPIE, ...)



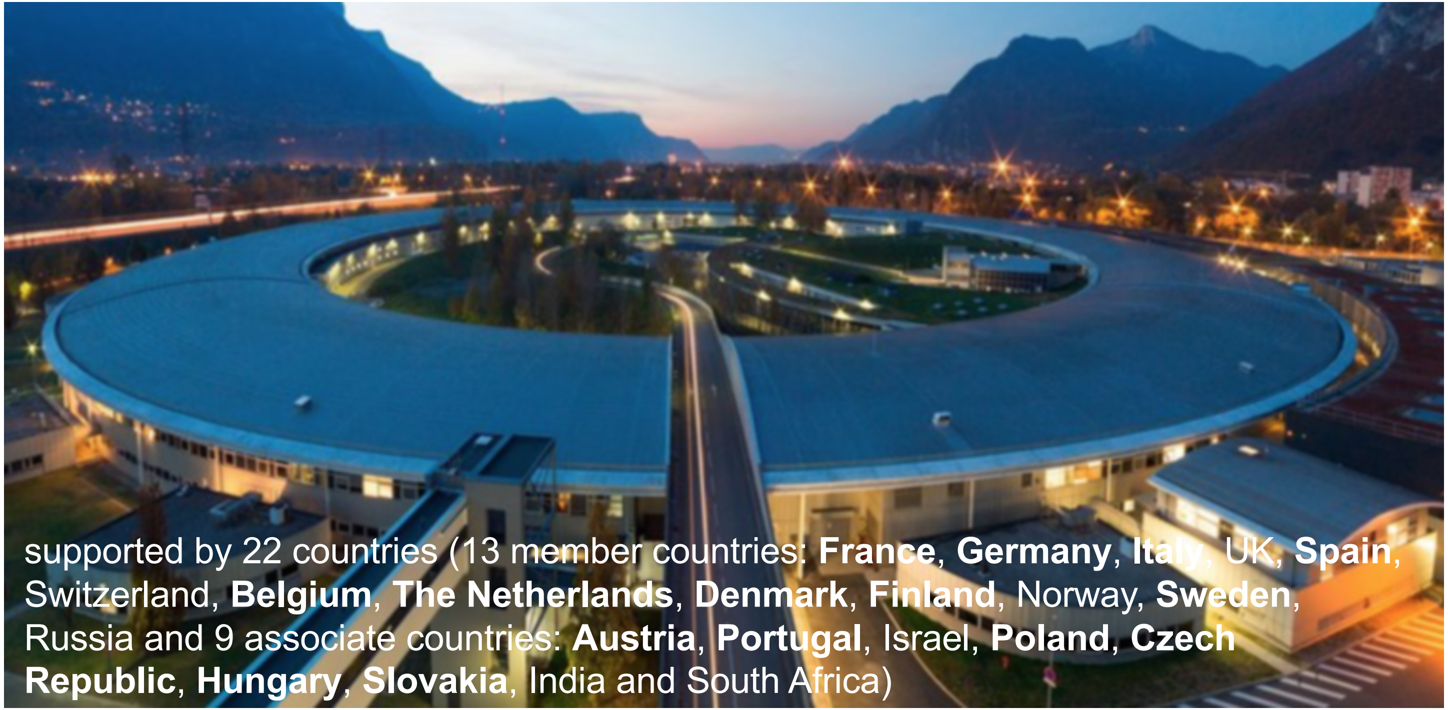
IS ELECTRO-MAGNETIC PLASMA PHYSICS IMPORTANT?





EUROPEAN SYNCHROTRON RADIATION FACILITY, GRENOBLE, FRANCE

One of our X-ray and Neutron "light" sources..

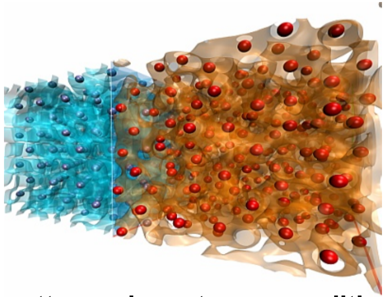


Recent upgrade: 177M €

Advanced comprehension of Laser-Plasma Accelerators


Source: HZDR.de

Modern cancer therapy


Source: Jan Vorberger

matter under extreme condition

ML-driven fast feedback systems for non-equilibrium processes requires

- Theory-guided Neural Networks that integrates all knowledge about the system => DT
- full knowledge of the beamline including potential perturbations (e.g. non-planar wavefronts, point spread function) => DT
- reliable ML techniques (uncertainty quantification, outlier detection)
- resolving ambiguity by joint reconstruction of orthogonal slices through the object

Reconstruction of non-linear & non-equilibrium processes ...

1) Differentiable Simulations

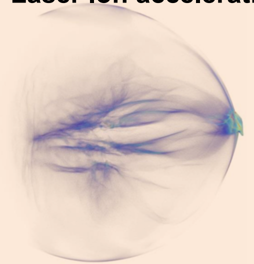
2) Multi-modal Data

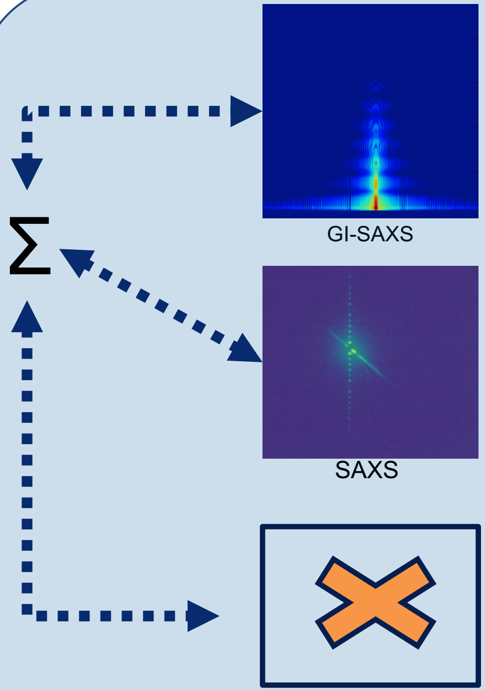
System description

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

Reduced Order Model, Surrogate Model

e.g.
Laser Ion acceleration





beam diagnostics,
Phase Contrast
Imaging, ...

(Willmann et al.), (Bethke et al.) @
Simulation with Deep Learning at ICLR’21

.. requires comprehensive Digital Twins.

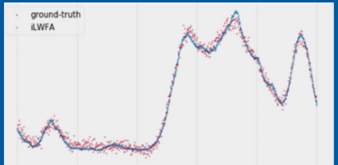
Building blocks of a Digital Twin

Amount of data

Surrogate Models

Invertible Neural Networks

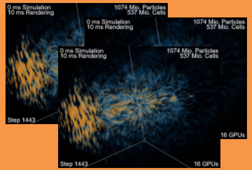
System parameters



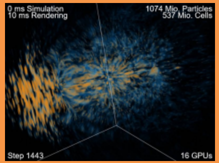
Diagnostics

Reduced Order Models

Data-driven methods



Full simulation runs

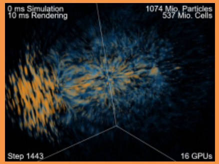


time evolution

Neural Solver

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

theoretical model



time evolution

Flexibility & Computational Cost

(Invertible) Surrogate Model

Task

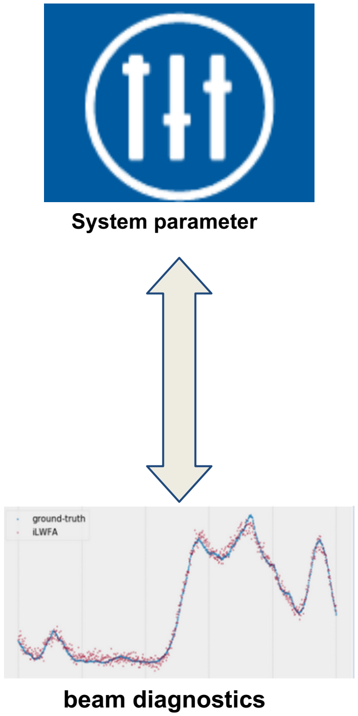
- Finding best system parameter (laser energy or stability)
- requires multiple simulation runs
- computationally expensive

Idea

- Surrogate model learns relationship from parameters to diagnostics
- fast inference (ms range) enables fast parameter scans
- Invertible surrogates models solve ambiguous inverse problem
- uncertainty quantification

Disadvantages

- needs high amount of data
- Black box model



Ambiguous inverse problems

For many applications, especially complex systems, the **forward process loses information** rendering the **inverse process ill-posed**. That means the inverse process is uncertain, i.e. multiple variables x can result in the same measurement y.

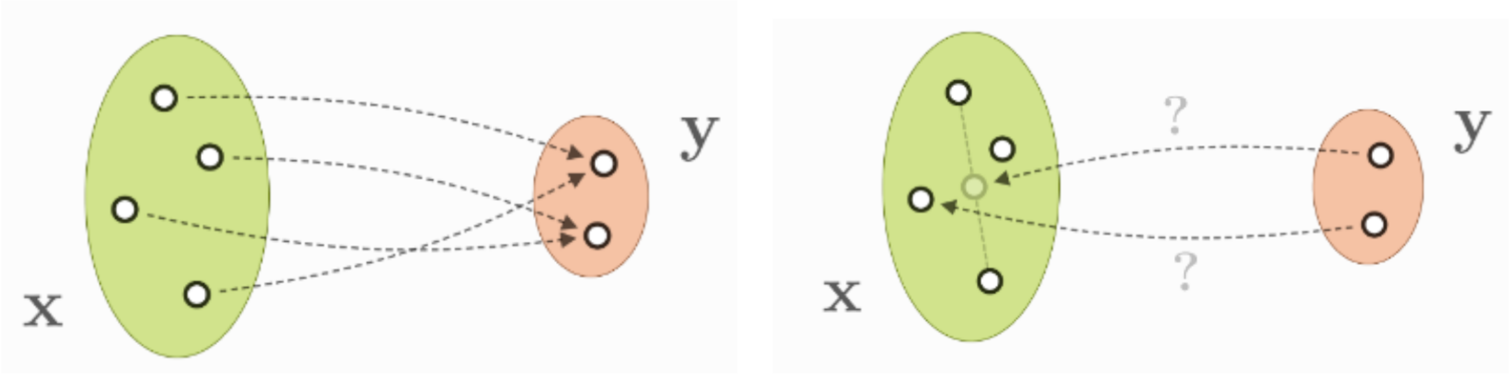


Figure: The intrinsic dimension of observation y is typically lower than independent variables x resulting in an ambigous inverse problem.

Normalizing flows

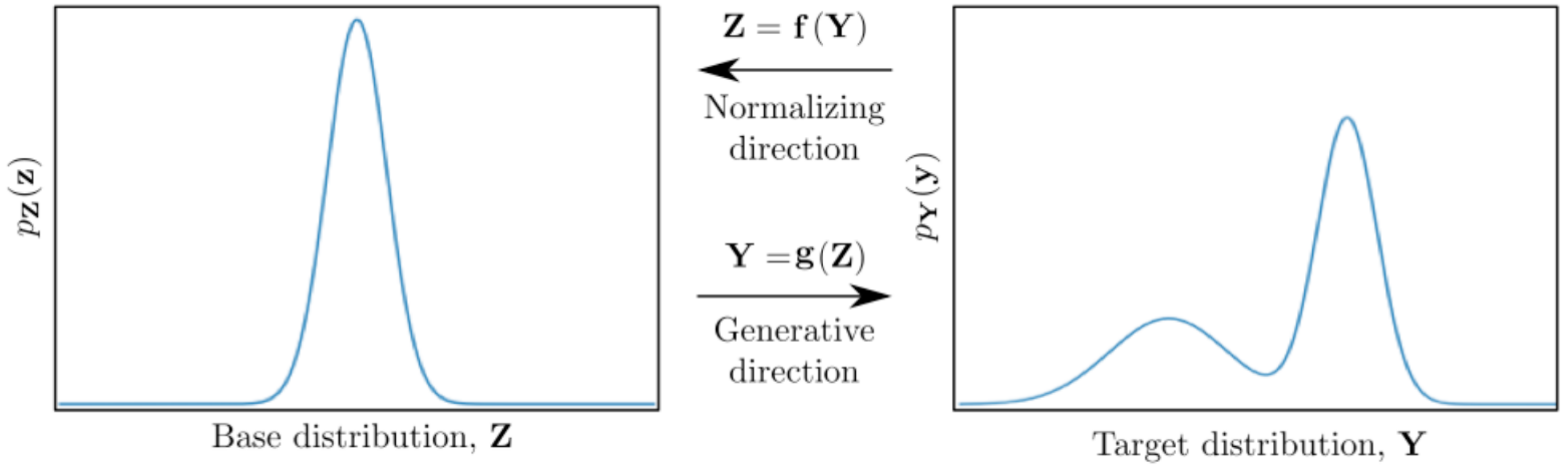
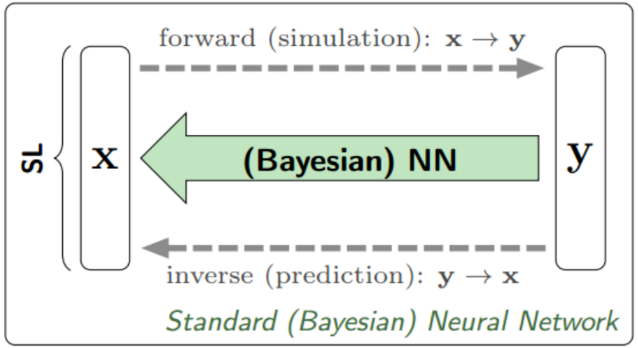
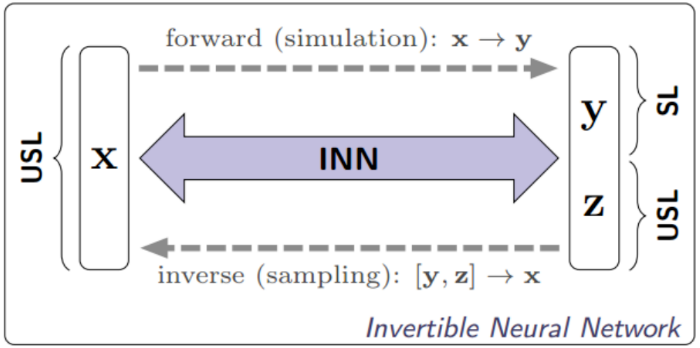


Figure: Mapping from normal distribution π(Z) to target distribution π(Y) via unconditioned normalizing flow. Image source: [Kobyzev et al., 2019]

Invertible Neural Networks

Iff. latent space $z \in R^K$ captures the information not contained in measurement $y \in R^M$, then the former non-bijective mappings becomes bijective via $\boldsymbol{INN(y, z) = (x)}$:





(see Peter's talk)

Figure: Abstract comparison of standard approach (left) and INN (right)

Laser-Plasma acceleration

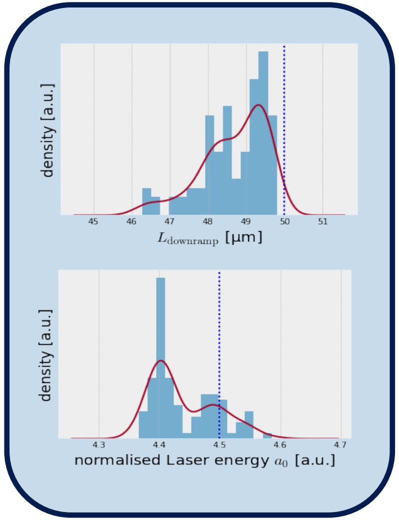
Surrogate Models

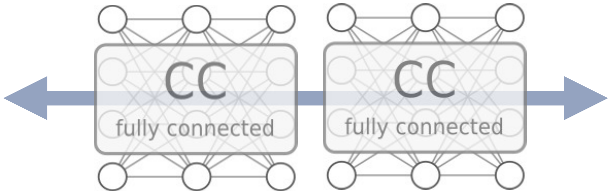
HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

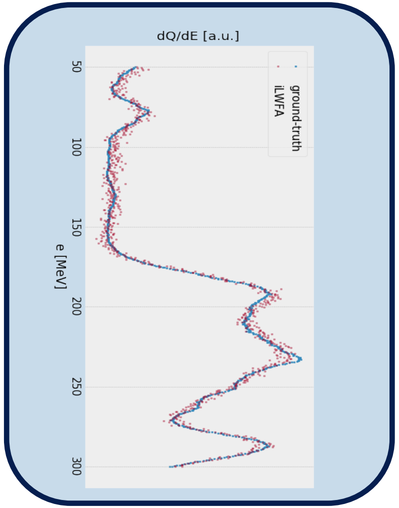
Invertible Surrogate Models jointly approximate simulation and reconstruction.
Implemented by **ML4IP framework** of Helmholtz AI@HZDR**.** Beta testers are welcome!

Way faster than Bayesian computation.

Fast parameter scans!







<u>Benefits</u>

Recover ambiguous mapping

Uncertainty quantification

Building blocks of a Digital Twin

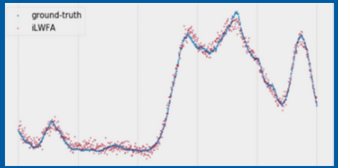
Amount of data

Flexibility & Computational Cost

Surrogate Models

Invertible Surrogate Models

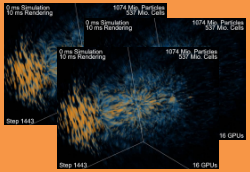
System parameters



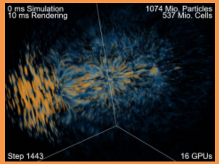
Diagnostics

Reduced Order Models

Data-driven methods



Full simulation runs

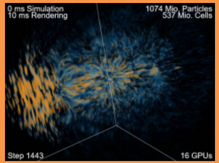


time evolution

Neural Solver

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

theoretical model



time evolution

Projection Based Reduced Order Model

Task

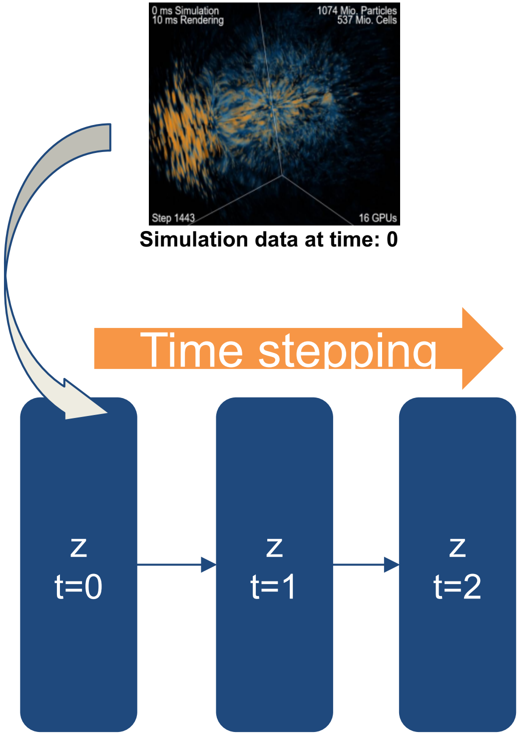
- Simulations need high performance tailored code
- Runs on large HPC systems
- Create high amount of data
- Not every scientist or laboratory has high computational resources

Solution & Benefits

- learns an mapping in a reduced domain
- Forward simulation is performed in reduced domain
- Generalization to other simulation parameter
- High memory compression & speed up
- Runs on a laptop → democratization

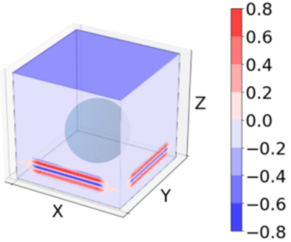
Disadvantages

- Simulation resolution depends on training data
- Quality loss through reduction

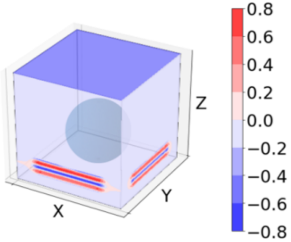


Shadowgraphy Reduced Order Model

Learning Reduced Order Representation (data reduction by x 7000) Willmann et al. (2021)

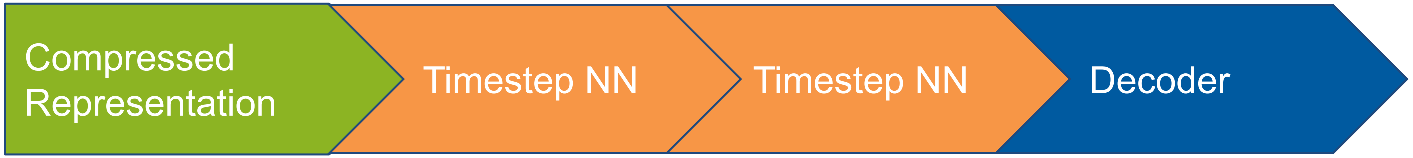


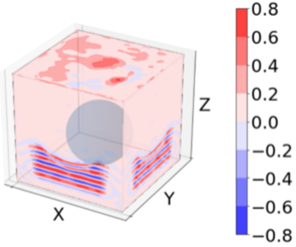
$t = 1.0$



$t = 1.0$

Time evolution in reduced order representation (800 x faster)





$a)\ r = 3.25\mu m, n = 1.65, t = 3.0$

Plasma Wave Imaging

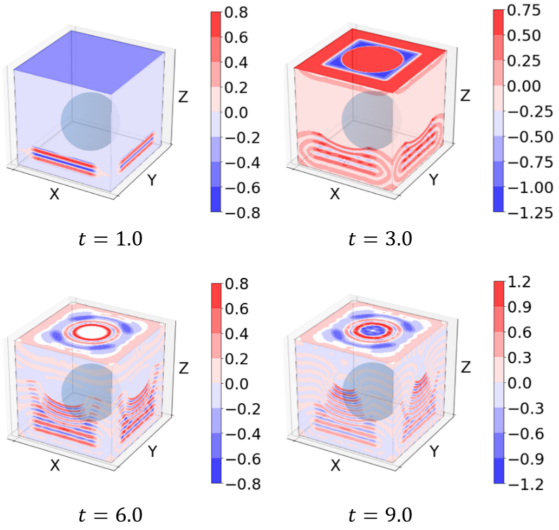
Physical model: propagation of 3D electrical field given by Maxwell's equations through a cell with a sphere(defined by radius and refractive index) in the middle of it

Aim: to reconstruct approximation of the field propagation such that radius and refractive index of the sphere are varying between the known values

Idea: a reduced order model

•size of the original domain is reduced – input arrays are projected to the smaller space

•approximation of solution is calculated in the smaller space



HELMHOLTZ AI

Plasma Wave Imaging

- a convolutional autoencoder for reduction of space dimensionality,

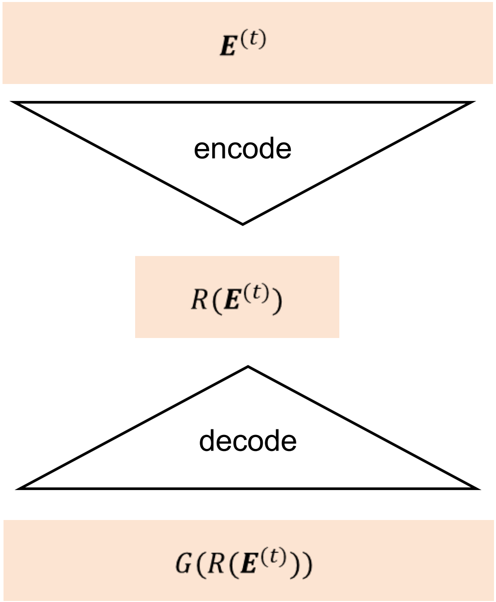
encoder $R: \mathbb{R}^{h \times w \times d} \to \mathbb{R}^{l}$,

h, w, d – height, width and depth of the original volume $\boldsymbol{E}^{(t)}$ at time point t, l – dimensionality of the reduced space,

decoder $G: \mathbb{R}^{l} \to \mathbb{R}^{h \times w \times d}$

Objective function is given as a supervised reconstruction error:

$$\mathcal{L}_{R,G}(\boldsymbol{E}^{(t)}) = ||\boldsymbol{E}^{(t)} - G(R(\boldsymbol{E}^{(t)}))||_1$$



Scheme of the model

Plasma Wave Imaging

- a projection approximator for computation of a solution in the reduced space: $F: \mathbb{R}^{k+1} \to \mathbb{R}^{l}$,

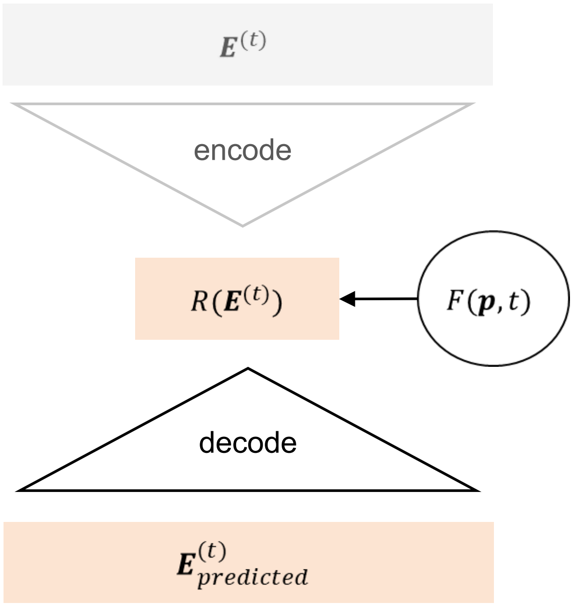
k – number of parameters and an additional parameter is point in time

Objective function is a supervised approximation error:

$$\mathcal{L}_F\left(\boldsymbol{E}^{(t)}, R, \boldsymbol{p}, t\right) = ||R\left(\boldsymbol{E}^{(t)}\right) - F(\boldsymbol{p}, t)||_2$$

for a pretrained encoder R.

Approximation of solution at time point t and parameters $\boldsymbol{p} \in \mathbb{R}^k$ is derived as $\boldsymbol{E}^{(t)}_{predicted} = G(F(\boldsymbol{p}, t))$



Scheme of the model

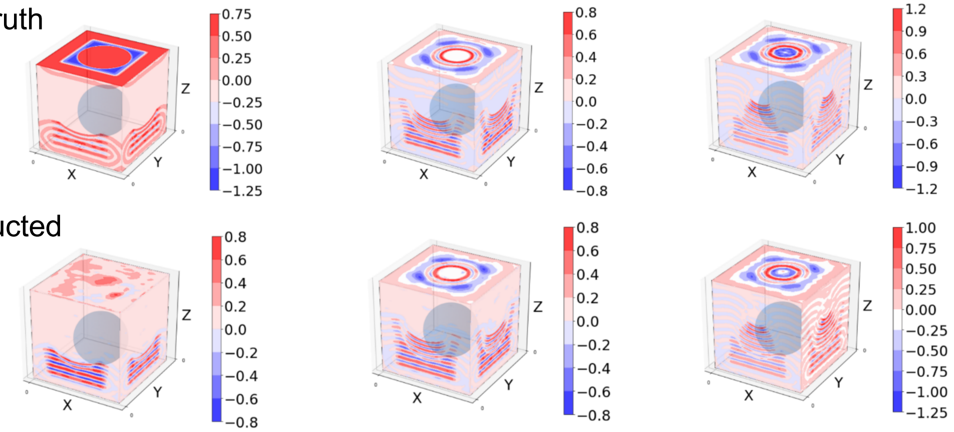
Plasma Wave Imaging

Examples of reconstruction

- Reconstructions are computed for parameters of sphere: radius r and refractive index n that are varying between values used for training

ground-truth

reconstructed



$r = 3.25\mu m, n = 1.65,$ $t = 3.0$

$r = 3.25\mu m, n = 1.65,$ $t = 6.0$

$r = 3.25\mu m, n = 1.65,$ $t = 9.0$

Average reconstruction error(L_1) over a full simulation: 0.01813

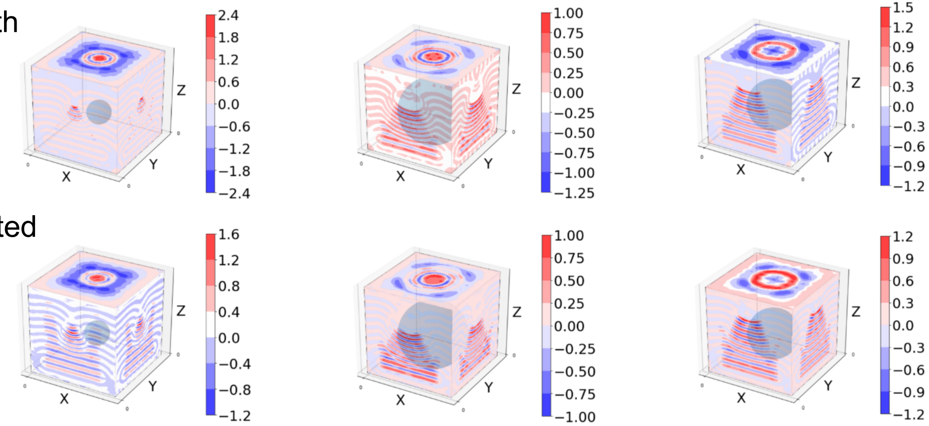
Plasma Wave Imaging

Examples of reconstruction

- Reconstructions are computed for parameters of sphere: radius r and refractive index n that are varying between values used for training

ground-truth

reconstructed



$r = 2.25\mu m, n = 1.65,$ $t = 9.0$ | $r = 3.75\mu m, n = 1.65,$ $t = 9.0$ | $r = 3.25\mu m, n = 1.25,$ $t = 9.0$

Average reconstruction error(L_1) over a full simulation:

0.01639	0.01823	0.01595

HELMHOLTZ AI

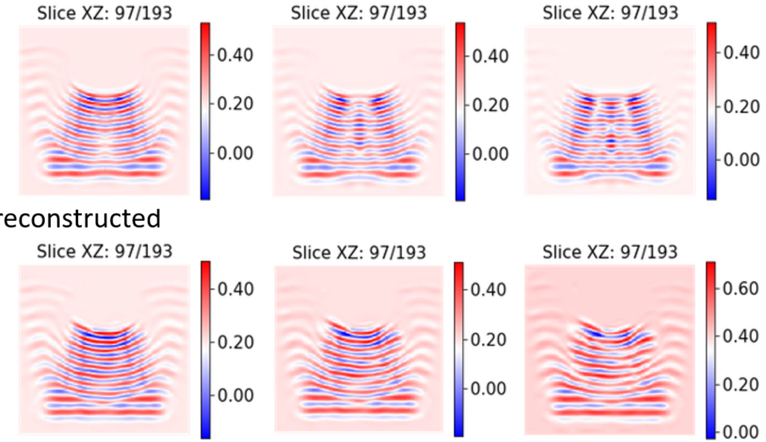
ROM for 3D Wave Equation

Extrapolation

Radius of Sphere

original

reconstructed



$r = 4.25\mu m$ +25%

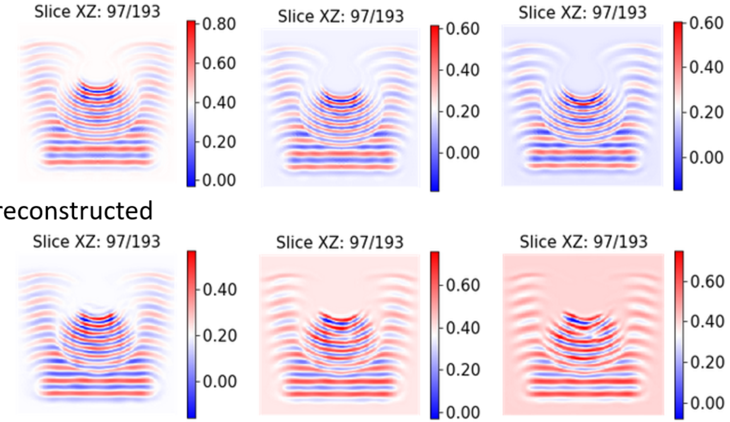
$r = 4.50\mu m$ +50%

$r = 4.75\mu m$ +75%

Refractive Index

original

reconstructed



$n = 1.75$ +50%

$n = 1.90$ +200%

$n = 2.00$ +300%

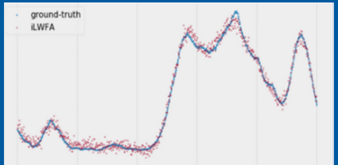
Building blocks of a Digital Twin

Amount of Data

Surrogate Models

Invertible Surrogate Models

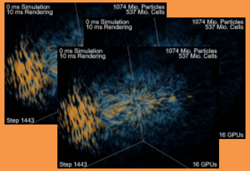
System parameters



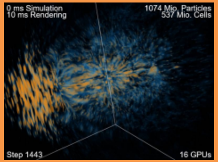
Diagnostics

Reduced Order Models

Data-driven methods



Full simulation runs

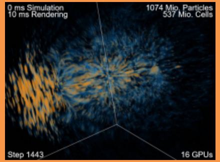


time evolution

Neural Solver

$$
\begin{aligned}
\nabla \cdot \mathbf{D} &= \rho \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{H} &= \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}
\end{aligned}
$$

theoretical model



time evolution

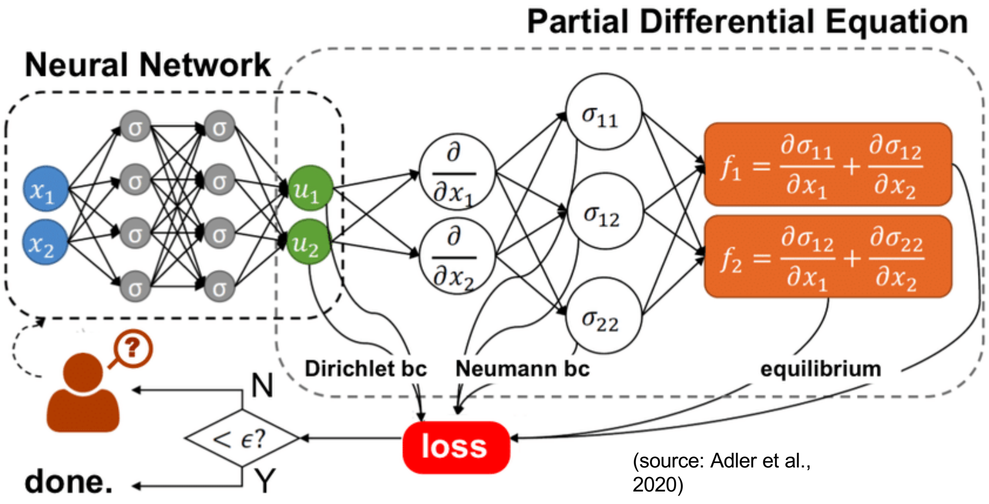
Flexibility & Computational Cost

Laser-Plasma acceleration

Approximate Simulations

Recover time evolution of phase space/field data by Physics-informed Neural Networks promise

- **exascale speed-up**, guidance by governing equations
- Helmholtz AI framework **NeuralSolvers** for 2d/3d Wave-✅, Heat- ✅ & Maxwell's equation (soon).



(source: Adler et al., 2020)

Toy problem 1: High-energy laser beams

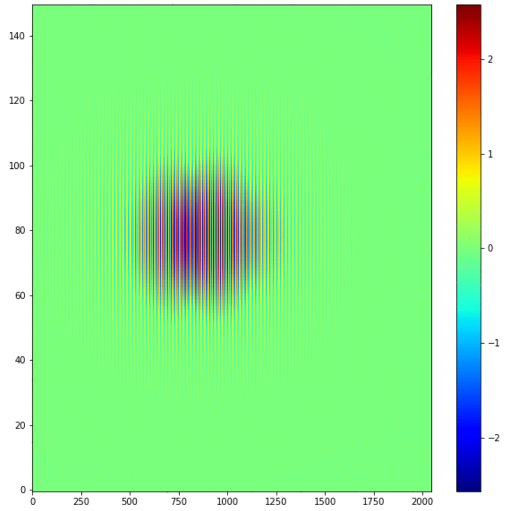
- Laser-beam propagates through vacuum
- 3D Helmholtz equation:

$$\left(\nabla^2 - c^{-1}\frac{\delta^2}{\delta t^2}\right)\vec{E} = 0$$

boundary conditions:

$$\vec{E}(x_b, y_b, z_b, t) = \vec{E}(-x_b, -y_b, -z_b, t)$$

Not being discussed today ☺



Toy problem 2: Quantum harmonic oscillator (QHO)

- elemental properties of condensed matter can be described by QHO
- time-evolution of 2D Schrödinger equation

$$i\psi_t - \frac{1}{2}(-\nabla^2 + V)\psi = 0$$

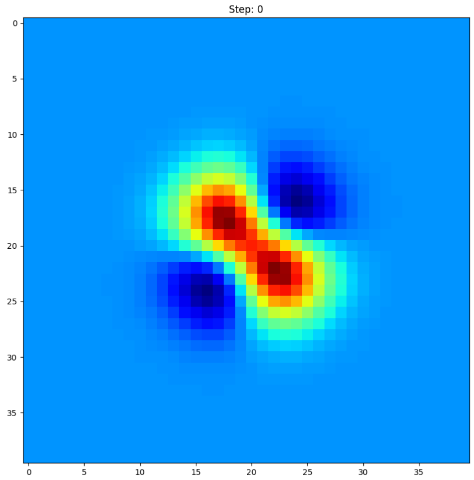
$$V(x,y) = x^2 + y^2$$

boundary conditions:

$\psi(x,y,0) =$ stationary solution

$\psi(x_b, y_b, t) = 0$

$\int \psi(\dots, t) = 1$



Example: Neural Solver for 2D QHO

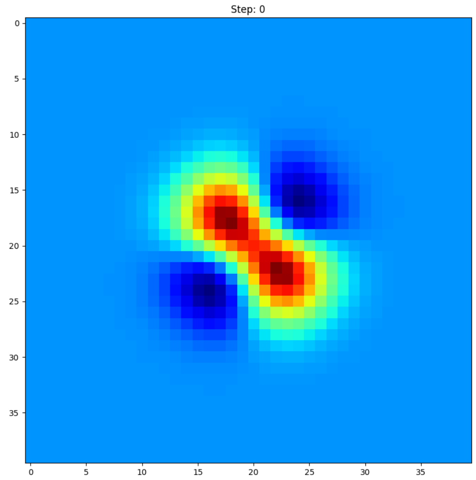
- 2D Schrödinger equation

$$i\psi_t - \frac{1}{2}(-\nabla^2 + V)\psi = 0$$

$$V(x,y) = x^2 + y^2$$

- NN approximates solution of our PDE: $\mathbf{nn(x,y,t) \approx \psi\ (x,y,t)}$
  - Multi-Layer Perceptron
  - tanh, some layers, …

- PDE solving translated into optimisation problem and parameters of neural network optimized accordingly



Neural solvers

- Training of our NN yields solution of our PDE: $\mathbf{nn(x, y, t) = \psi\ (x, y, t)}$
  - Multi-Layer Perceptron
  - ReLU, some layers, …

- loss $L\ = \alpha L_0\ +\ L_b + L_f$
  - L_0 => initial state t=0: $L_0 = \sum \|nn(\dots, t_0)\ - \psi(\dots, t_0)\|_2^2$
  - L_b => boundary conditions: $L_b = \left(1 - \iint_{x,y\,\in\,Tb} |\psi|\ dx\ dy\right)^2$
  - L_f => pde loss $L_f =\ \dots$

PDE loss

- temporal evolution of 2D QHO:

$$i\psi_t = \frac{1}{2}(-\nabla^2 + V)\psi \text{ -> } i\psi_t - \frac{1}{2}(-\nabla^2 + V)\psi = 0$$

- setting temporal evolution to zero yields **PDE loss** L_f :

$$\mathrm{L_f} = \sum_{(x,y,t)} f_u(x,y,t)^2 + f_v(x,y,t)^2$$

$$f_u(x,y,t) = \frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial^2 v}{\partial x^2} + \frac{1}{2}\frac{\partial^2 v}{\partial y^2} - \frac{1}{2}x^2 u - \frac{1}{2}y^2 u$$

$$\mathrm{u = re}(\psi), v = im(\psi), f_v = \ (similar)$$

- network predicts ψ, partial derivatives computed via automatic differentiation

$$\psi_t = \frac{\partial nn(x,y,t)}{\partial t}$$

↔

```
psi = net.forward(x,y,t)
psi_t =
torch.autograd.grad(psi,t)
```

Sketch of the approach

Compute Domain

$x \in [l_x, u_x]$

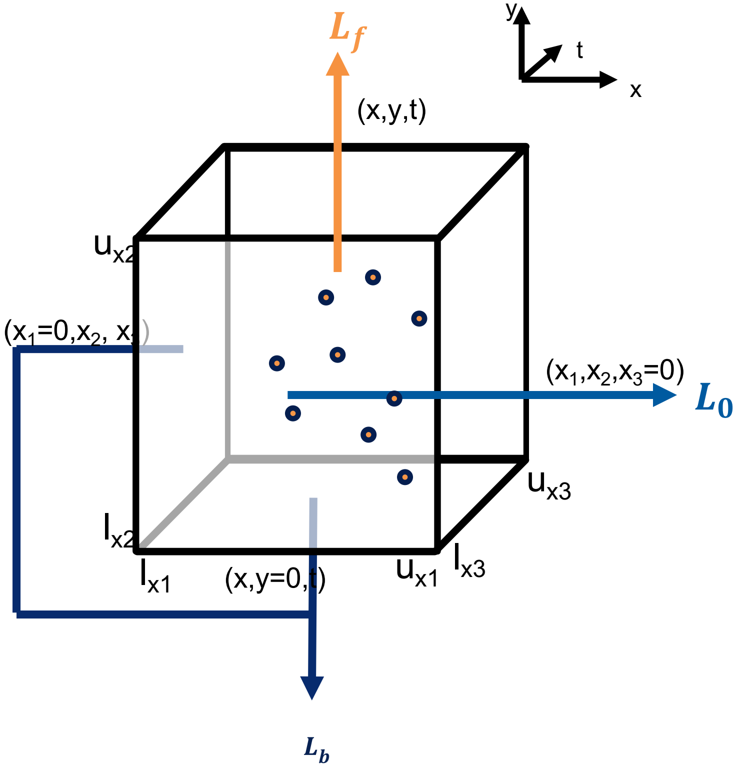
$y \in [l_y, u_y]$

$t \in [l_t, u_t]$

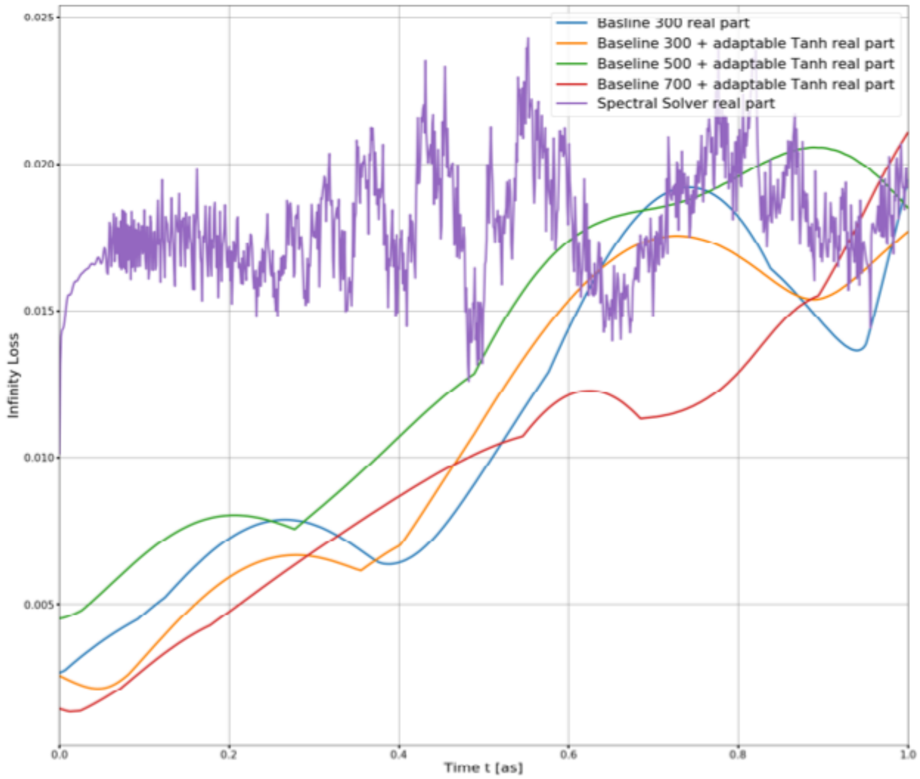
Loss

$$L = \alpha L_0 + L_b + L_f$$

- points (x, y, t=0) sampled for interpolation of initial state ψ_0 (L_0)
- PDE loss L_f evaluated at randomly sampled co-location points
- boundary conditions enforced at corresponding points: $\psi(x = 0, y, t)$ or $\psi(x, y = 0, t)$ => L_b



Comparison of PINN vs Spectral Methods for solving 2D parabolic PDE

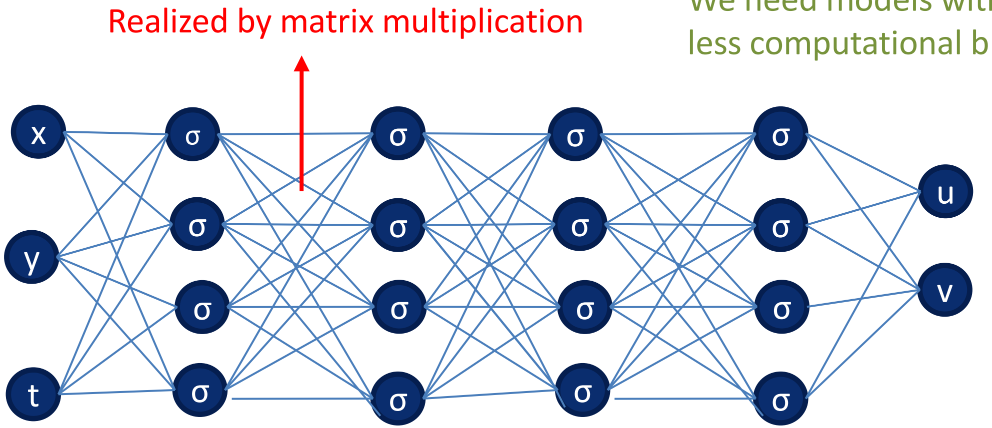


Bottleneck of Physics-informed neural networks

- **Accuracy** of the model highly depends on the capacity (number of parameters)

→ Increasing capacity = computational blow-up

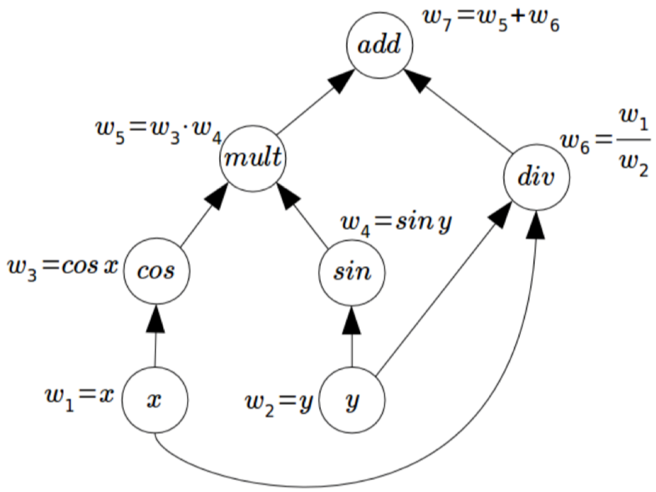
- Large models make Autodifferention **memory and time intensive**



We need models with high capacity but with less computational blow up

Automatic Differentiation

- Computing derivatives:
  - Finite differences, symbolic computation
  - Automatic differentation yields exact gradients efficiently
- Complex numeric computations can be decomposed into elementary operations (+,-,*,/,exp, log, sin, cos, etc.)
- These derivatives are already known.



Computational graph of
f(x,y) = cos(x)*sin(y) + x/y

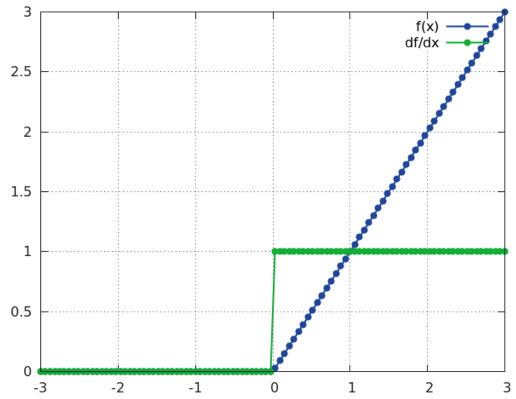
Automatic Differentiation

- Forward as well as backward operation of each node easy to implement

```
class ReLU(torch.autograd.Function):

    def forward(ctx, input):
        ctx.save_for_backward(input)
        return input.clamp(min=0)

    def backward(ctx, grad_output):
        # Compute gradient wrt. input
        input, = ctx.saved_tensors
        grad_input = grad_output.clone()
        grad_input[input < 0] = 0
        return grad_input
```



Conditional Computing

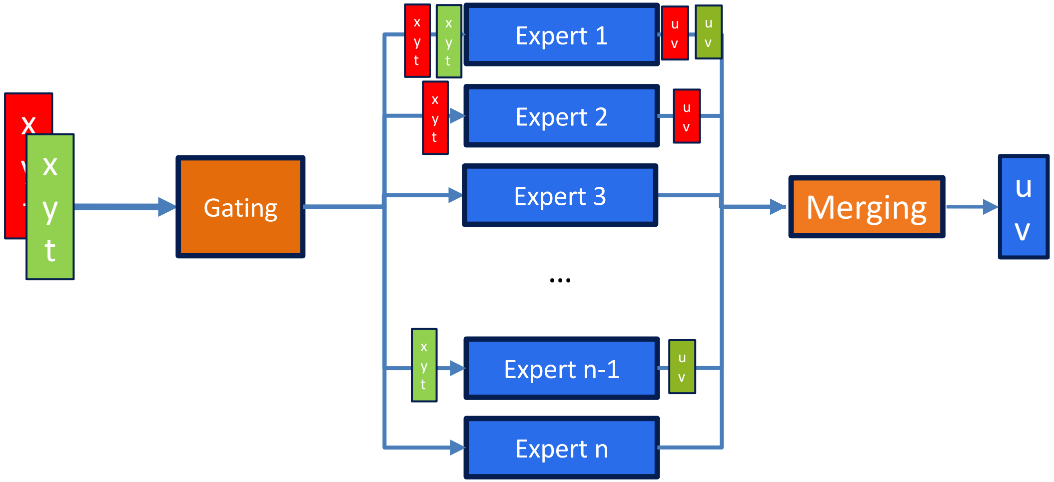
"Conditional Computation refers to a class of algorithms in which each input sample uses a different part of the model, such that on average the compute, latency or power (depending on our objective) is reduced. " Bengio et. al [10]

- Conditional Condition in terms of PINNs leads to **domain decomposition**
- Domain decomposition is commonly used tool to accelerate simulations
- Mixture of expert is a popular conditional computation approach

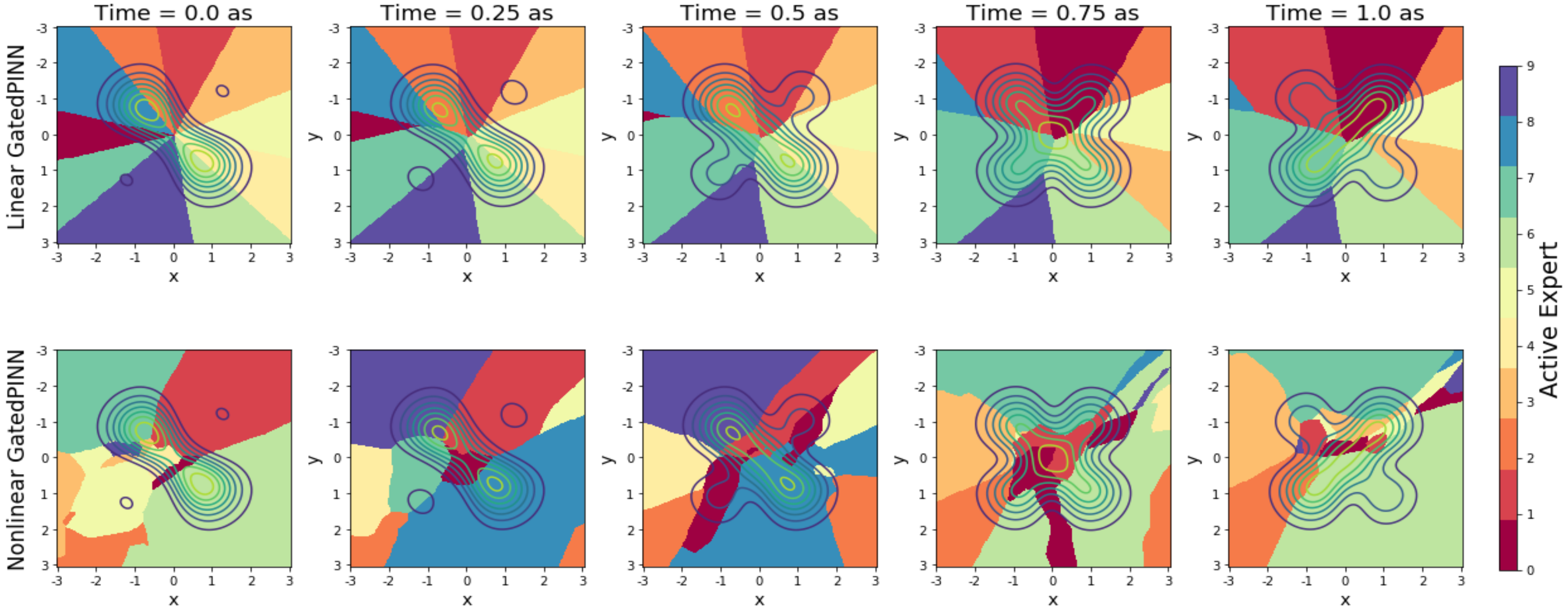
GatedPINN = Mixture-of-Expert + Physics-informed neural network

GatedPINN

- Meng et. al [2] showed that domain decomposition increasing the quality of PINNS
- Mixture of Experts Models provides an adaptive domain decomposition through their gating mechanism



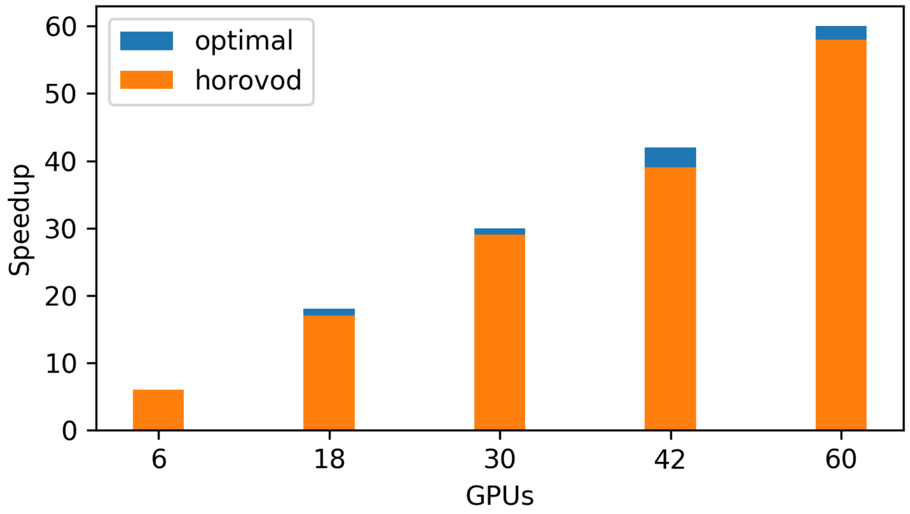
Domain Decomposition



Scalability

- In order to handle large amount of datapoints a parallelization is necessary
- Data-parallelism framework Horovod is used
- Speedup scales and power draw scales with the number of used GPUs

→ Horovod is an excellent choice for the distributed training of physics-informed neural networks



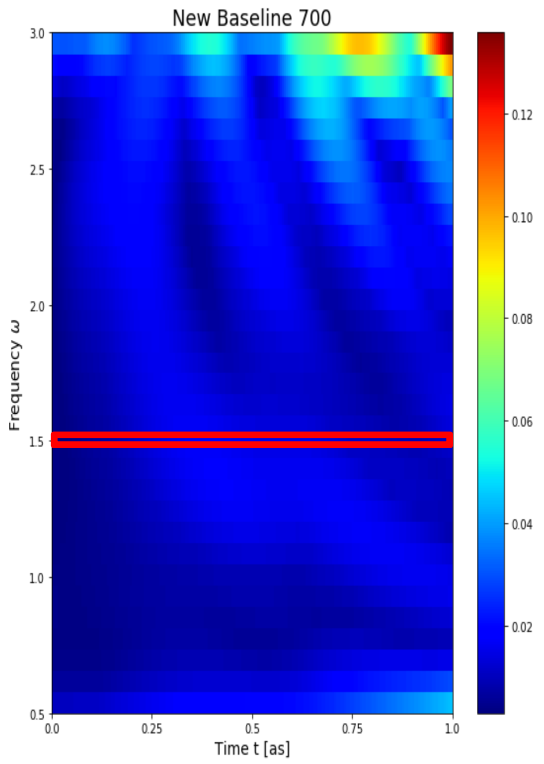
Interpolation in Solution Space

https://github.com/ComputationalRadiationPhysics/NeuralSolvers

- Extending the PINN input by the frequency of the quantum harmonic oscillator

- Reduced runtime of parameter scans
  - Spectral Solver: 40 Minutes /32 frequencies
  - Paramaterized PINN: 6 Minutes / 32 frequencies
  - 6x Speedup
  - Jumping to later time points are possible with PINN
  - Avoid restarts of the solver

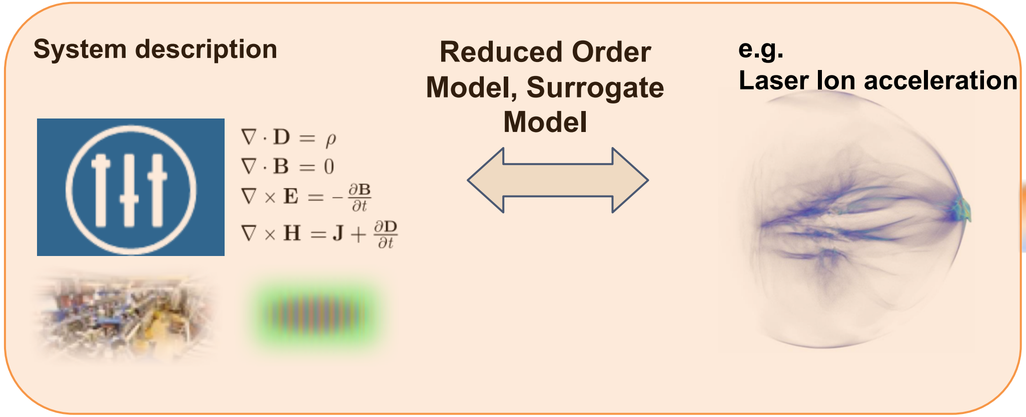
- Reduced memory footprint
  - Spectral Solver: 20 gb for 32 frequencies
  - PINN: 48 mb for complete domain + interpolation



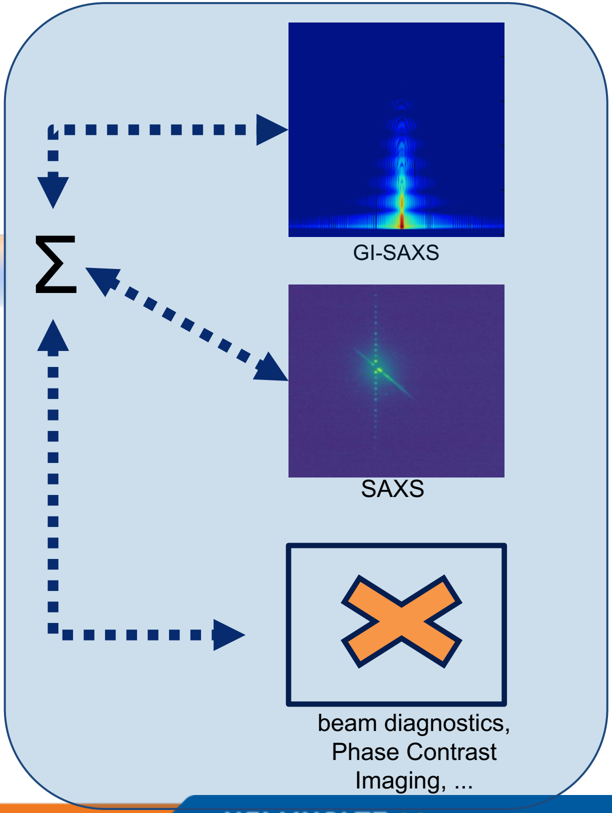
Reconstruction of non-linear & non-equilibrium processes ...

1) Differentiable Simulations

2) Multi-modal Data



$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$



(Willmann et al.), (Bethke et al.) @
Simulation with Deep Learning at ICLR'21

.. requires comprehensive Digital Twins.

Thanks for your attention!

and thanks to my Machine Learning team at **HZDR**! ☺

HI Jena
Helmholtz-Institut Jena

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

DLR

UFZ

UKE HAMBURG

HARVARD MEDICAL SCHOOL




HIRING:
PostDoc:
Inverse
Imaging
Problems





n.hoffmann@hzdr.de