Improving data-driven model predictions using physics in the CERN accelerator complex

F. M. Velotti, V. Kain, F. Huhn, A. Lu

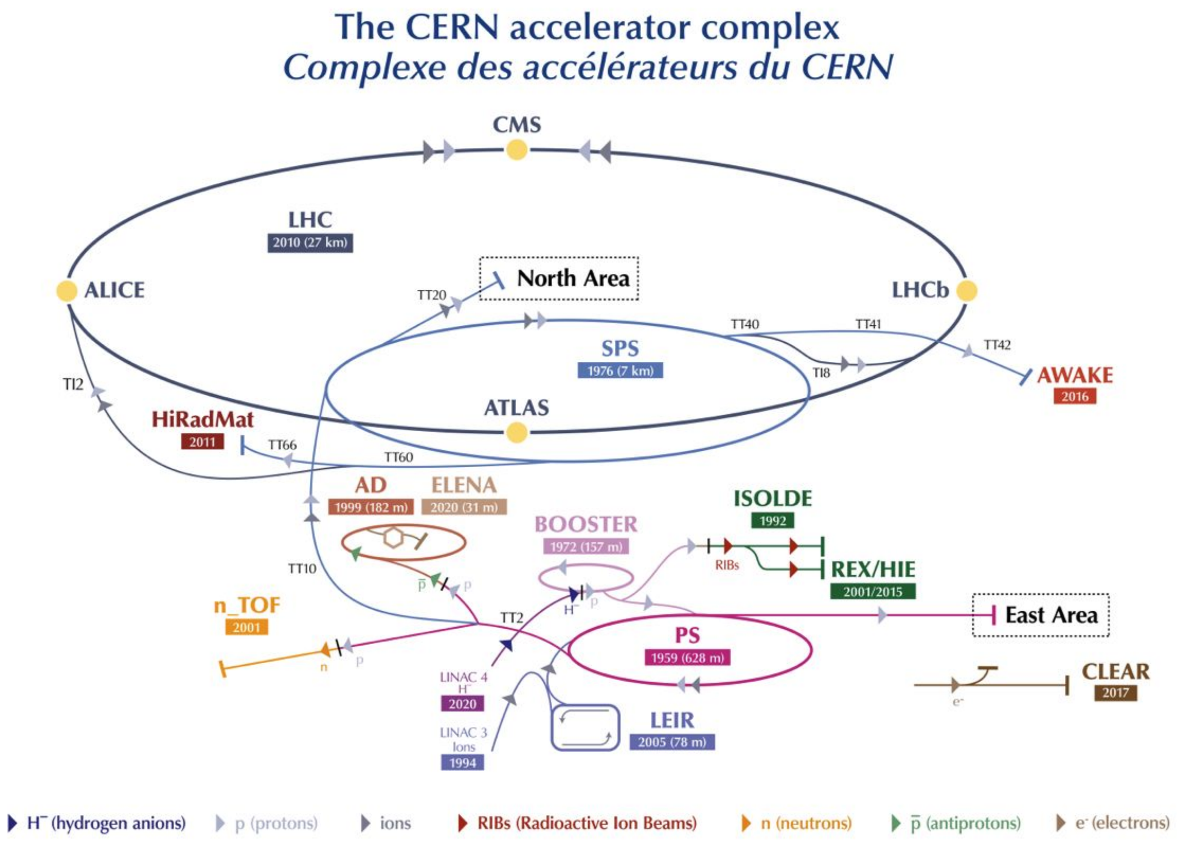
Thanks to M. Shenk, M. Barnes, B. Goddard, K. Li
And to the ML community forum

Outline

- ➔ Introduction and motivation
- ➔ Physics-informed data driven model being explored
- ➔ Some applications
- ➔ Models in operation
- ➔ Conclusions

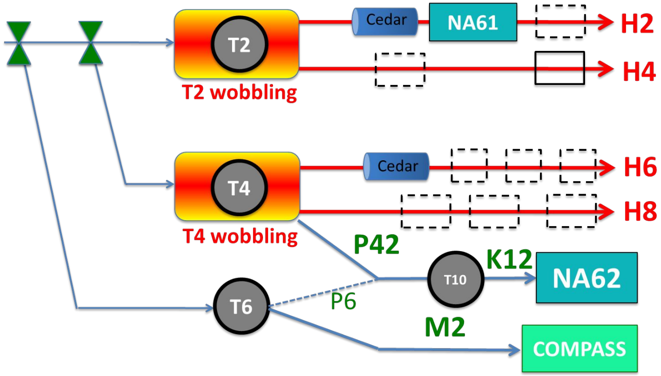
Introduction

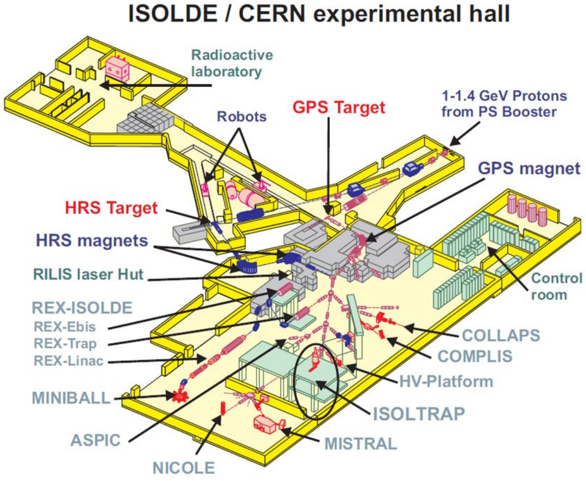
The CERN accelerator chain



LHC and other experiments

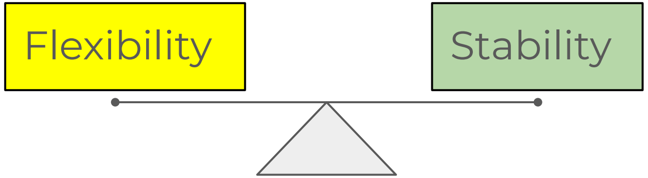
- ➔ The SPS North experimental Area hosts very interesting and demanding fixed target experiments: COMPASS, NA62…
  - **Slow extraction is used to deliver constant proton and heavy ion flux ⇒ 3rd integer slow extraction**
- ➔ ISOLDE takes the largest number of protons accelerated at CERN
- ➔ The PS serves directly several experimental facilities, like EAST area and nToF, but also indirectly via AD/ELENA: ASACUSA, ATRAP, GBAR…
- ➔ **LHC towards HL-LHC ⇒ high integrated Lumi!**

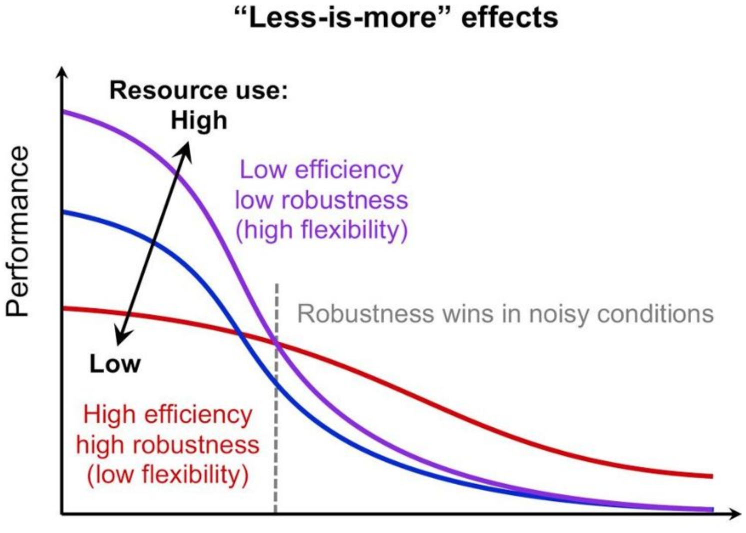




Motivation

- Multi-purpose machines need to efficiently share the time among users (experiments) and guarantee stable and reproducible conditions
    - Classically this is a trade-off to decide upon
- A possible way to break the balance is to <u>**“predict” the effect of changes**</u> in a very entangled and complicated system as an accelerator chain





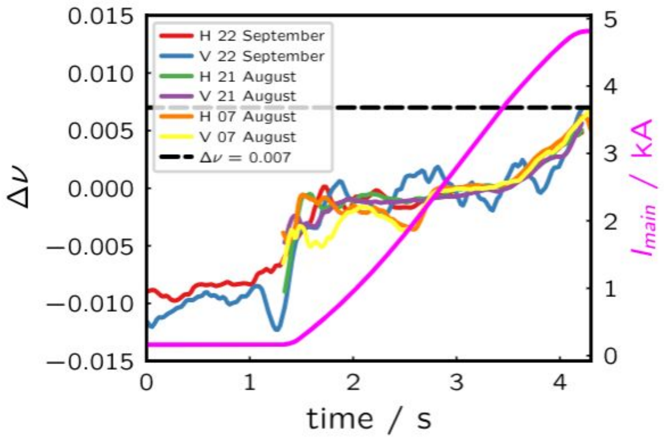
[source]

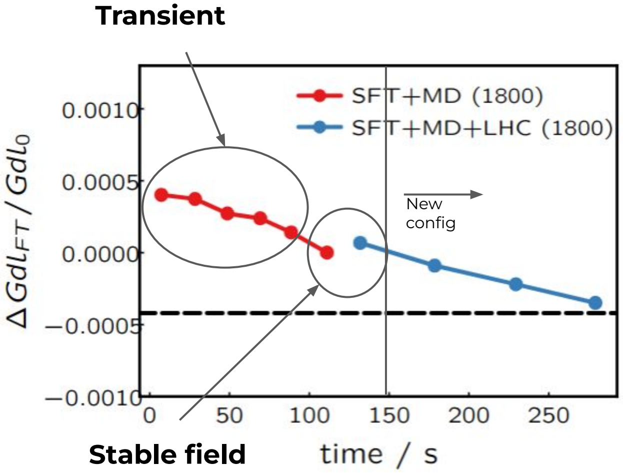
SPS slow extraction reproducibility

➔ Hysteresis on the main SPS quadrupoles responsible for extracted beam quality degradation [1]

- Beam based measurements highlighted tune variation
- Magnetic measurements on spare quadrupole showed field variation compatible with beam observations

Tune variation in the cycle after a configuration change



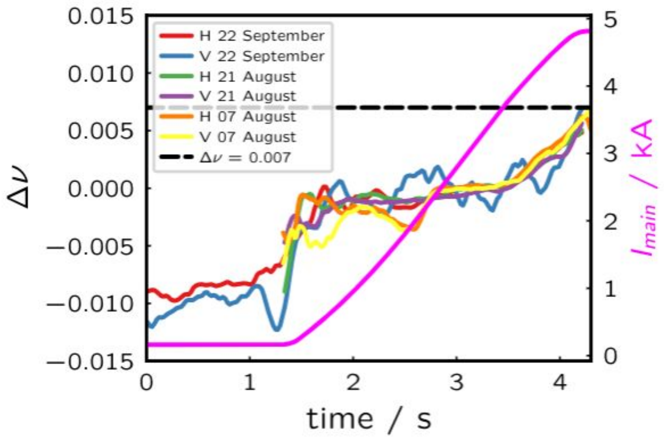


SPS slow extraction reproducibility

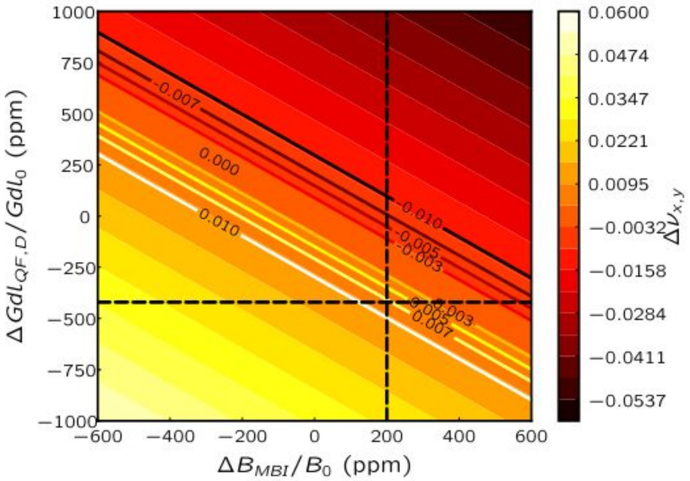
➔ Hysteresis on the main SPS quadrupoles responsible for extracted beam quality degradation [1]

- Beam based measurements highlighted tune variation
- Magnetic measurements on spare quadrupole showed field variation compatible with beam observations

Tune variation in the cycle after a configuration change

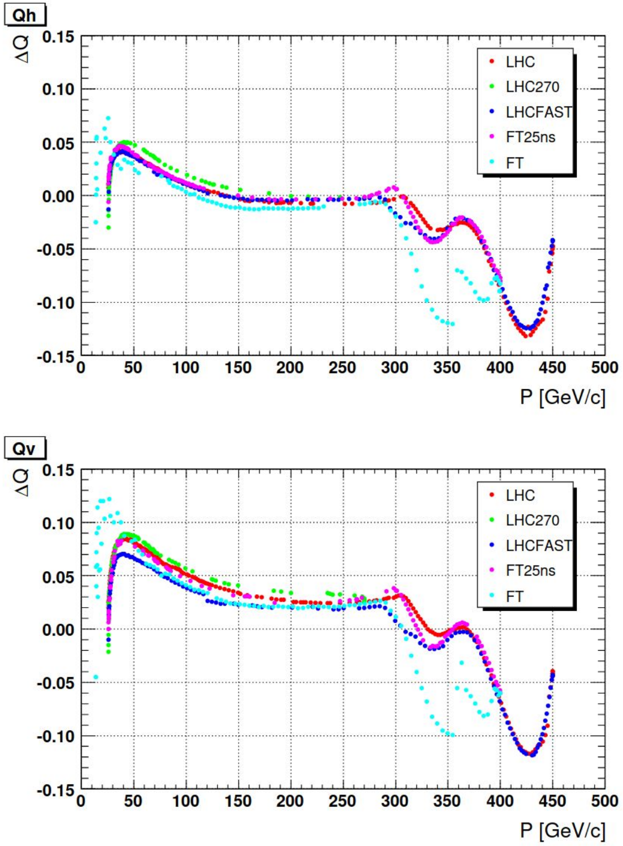


MADX simulations from quad and dipole measurements



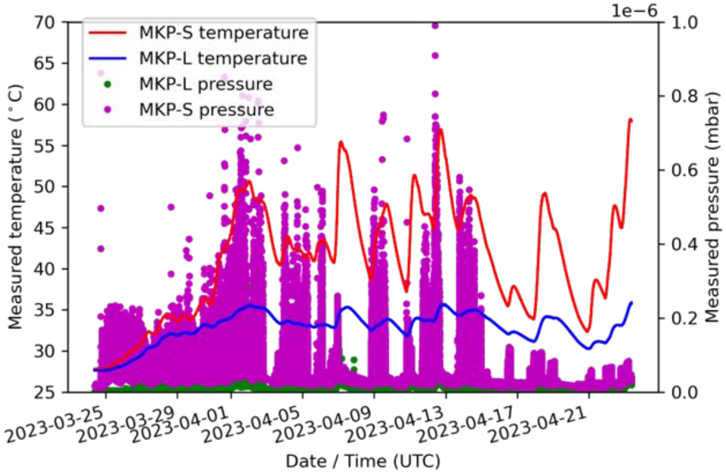
Chromaticity and tune settings

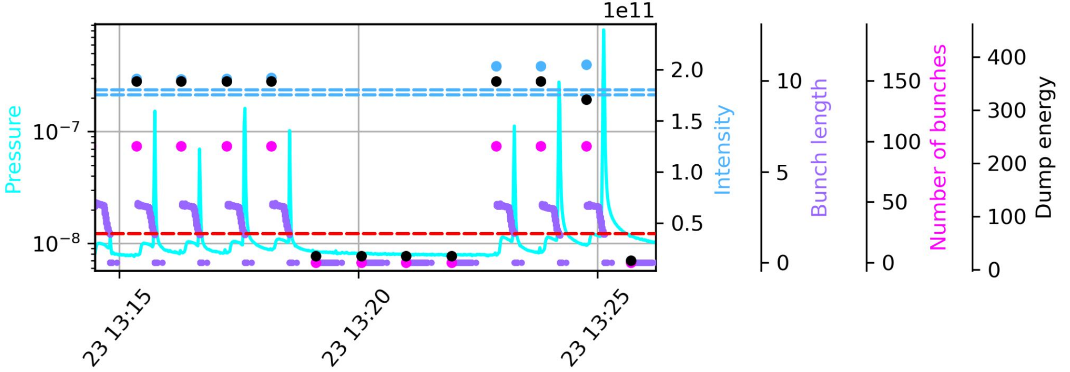
- Multi-cycled machines need to adapt to different beam requirements hence different parameters
- This translate into the need to be able to quickly change from one set of settings to others
    - Like tune, chromaticity
- On paper, this could be very simple but in reality we have eddy-currents, non-linearity and non-ideality of magnets and power supplies
- How can we produce a model that given some target beam parameters returns settings needed for the accelerator magnets?



High intensity limitations in the SPS

- ➔ High intensity particle beams heat up accelerator components
- ➔ Other effects, still linked to HI, lead to vacuum pressure rise
- ➔ Kickers are usually the most sensitive:
  - Hold high voltage
  - Yoke directly in vacuum and exposed to beam usually with no shielding
- ➔ In the SPS, the MKP (injection) and the MKDH (dump) are the most reactive to high intensity beams





What are we looking for and what we have

- ➔ Correct spill structure by predicting machine magnetic behaviour
  - Very accurately predict effect on the beam of available machine settings => easy to change users on the fly and maintain performance
- ➔ Predict beam induced heating, vacuum behaviour given beam parameters and status of our systems from beam observations => better scheduling and more efficient operation

What are we looking for and what we have

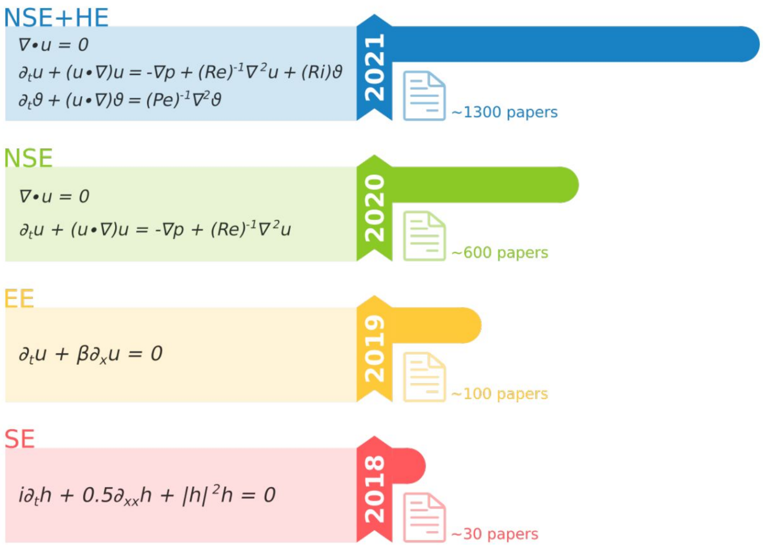
- ➔ Correct spill structure by predicting machine magnetic behaviour
  - Very accurately predict effect on the beam of available machine settings => easy to change users on the fly and maintain performance
- ➔ Predict beam induced heating, vacuum behaviour given beam parameters and status of our systems from beam observations => better scheduling and more efficient operation

- ➔ The available dataset we have are not enormous
  - Complicated NN easy to overfit
  - Physics models available (in many cases) but too slow or not very accurate
- ➔ Working towards exploiting physics knowledge to regularise, build features, improve NN performance and be able to “extrapolate” to future or unknown quantities

Physics informed data-driven models being explored

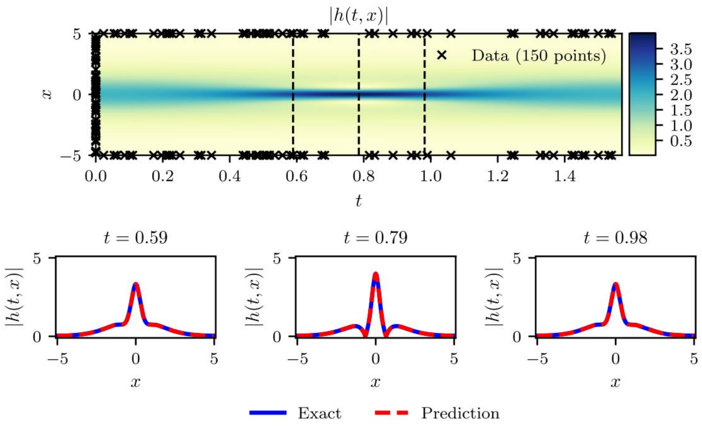
Physics Informed Neural Networks

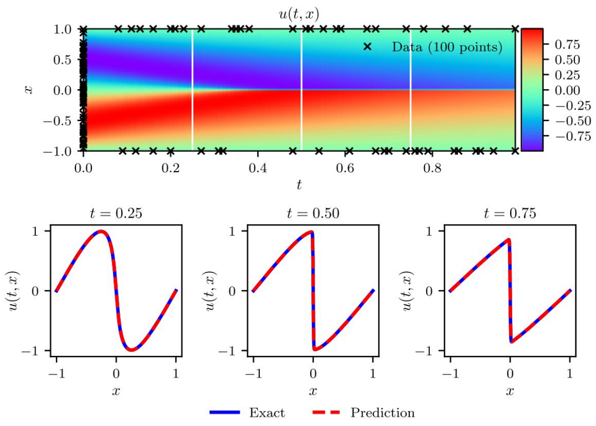
- ➔ Embedding physics knowledge in NN is becoming very common
- ➔ Very complete summary of applications [2] (figure taken from [2]) and the general field of physics informed ML [2.1]
- ➔ We were looking for a way to extend temperature prediction to very long time periods and to predict ferrite temperature...



Physics Informed Neural Networks

- ➔ First proposed to solve nonlinear PDE [3] (all plots from [3])
- ➔ Basically using boundary and initial conditions values, NN can interpolate the whole system dynamics “knowing” the PDE that describe the system
  - ◆ At the same time though, one can just use a physics loss term…it doesn’t have to be a PDE system

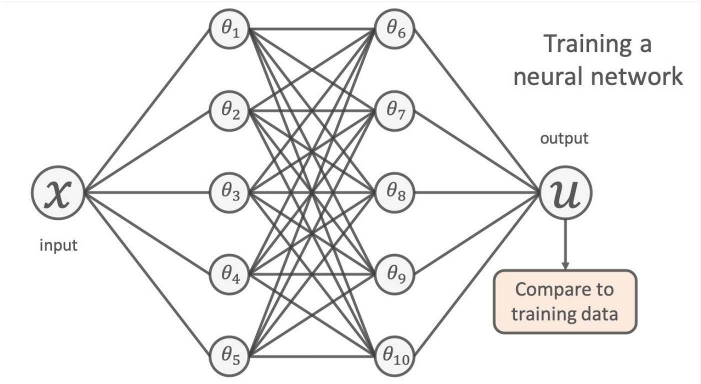




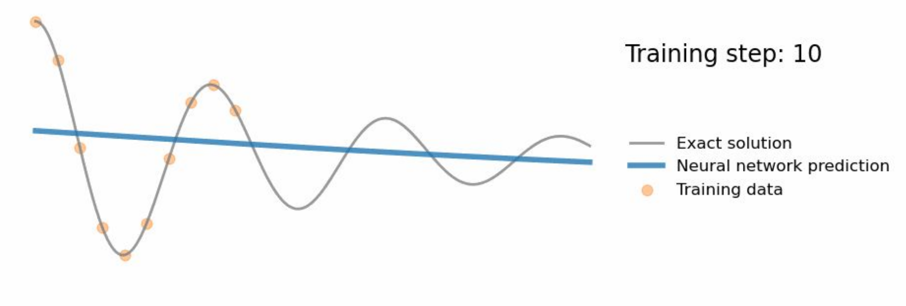
Physics Informed Neural Networks

➔ DNN cannot extrapolate beyond the training domain…which is exactly what we would expect from interpolation function

min(Loss) => Loss = Mean(data - prediction)^2



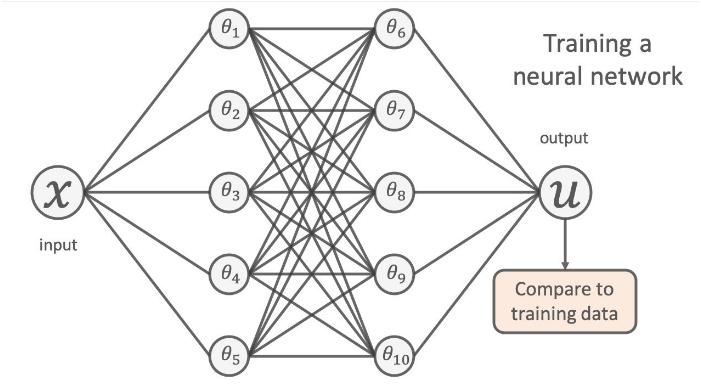
Source: [4]



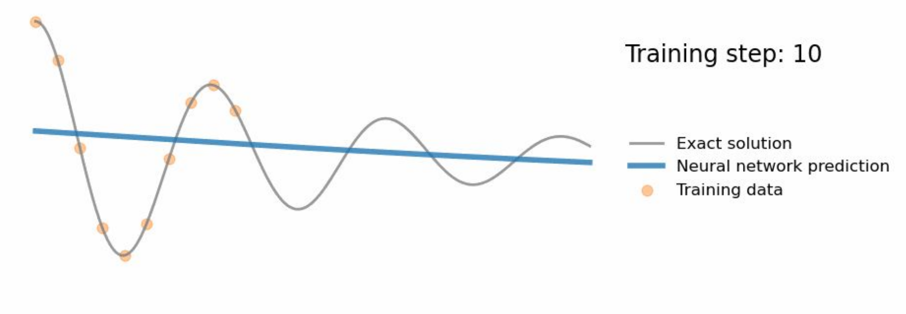
Physics Informed Neural Networks

➔ DNN cannot extrapolate beyond the training domain…which is exactly what we would expect from interpolation function

$$\mathcal{L} = \sum_{i}^{N} (u(x_i) - \hat{u}(x_i, \theta))^2$$



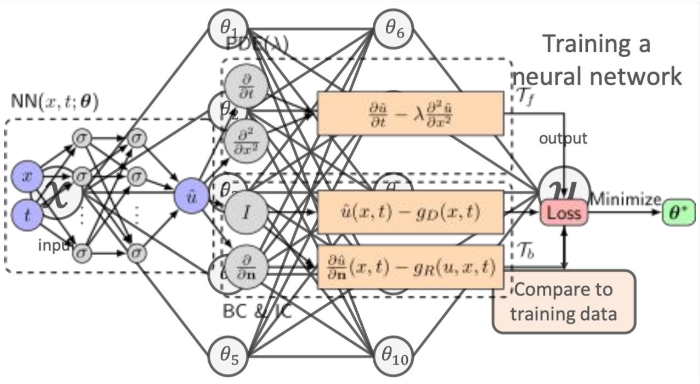
Source: [4]



Physics Informed Neural Networks

➔ DNN cannot extrapolate beyond the training domain...which is exactly what we would expect from interpolation function

$$\mathcal{L} = \sum_{i}^{N}(u(x_i) - \hat{u}(x_i, \theta))^2$$



Source: [4]

➔ Go beyond data domain => more information needed:

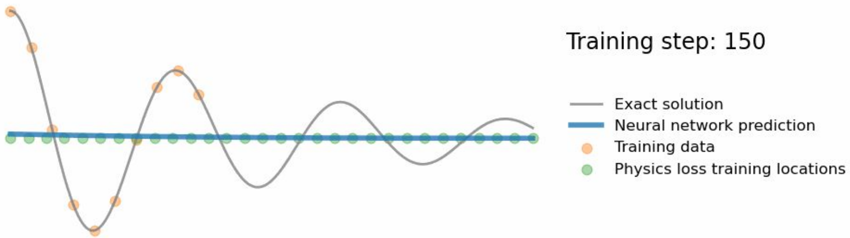
min(Loss) => Loss = Mean(data - prediction)^2
+ Additional_info(prediction)

$$\mathcal{L}_1 = 1/N\sum_{i}^{N}(u(x_i) - \hat{u}(x_i, \theta))^2$$

$$\mathcal{L}_2 = 1/M\sum_{j}^{M}(\frac{\partial^2 \hat{u}}{\partial x^2} - \frac{\partial \hat{u}}{\partial t})^2$$

$$\mathcal{L}_3 = \hat{u}(x, t=0) - f(x)$$

$$\mathcal{L}_4 = \hat{u}(x=0, t) - u_0$$



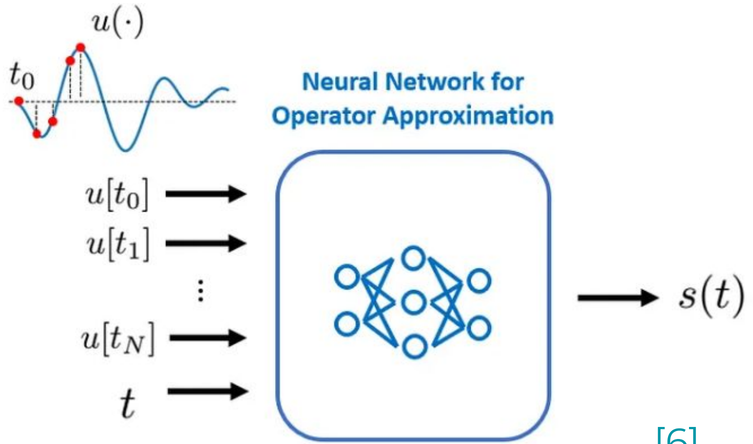
$$\mathcal{L}_{tot} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2 + \gamma\mathcal{L}_3 + \eta\mathcal{L}_4$$

Physics Informed DeepONets [5]

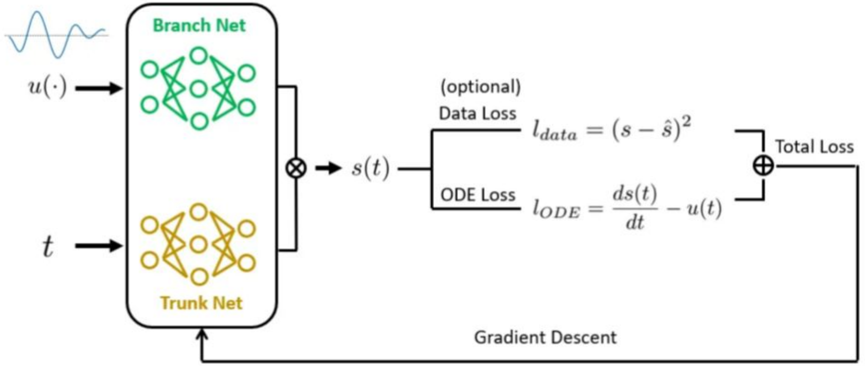
- ➔ Usually we have to deal with forced/controlled systems
- ➔ We are not learning a simple function anymore but an operator:

$$G(u(t),t) = s(u(t))$$

- ➔ In many cases, long memory/inertia
- ➔ Need to include “all” past history

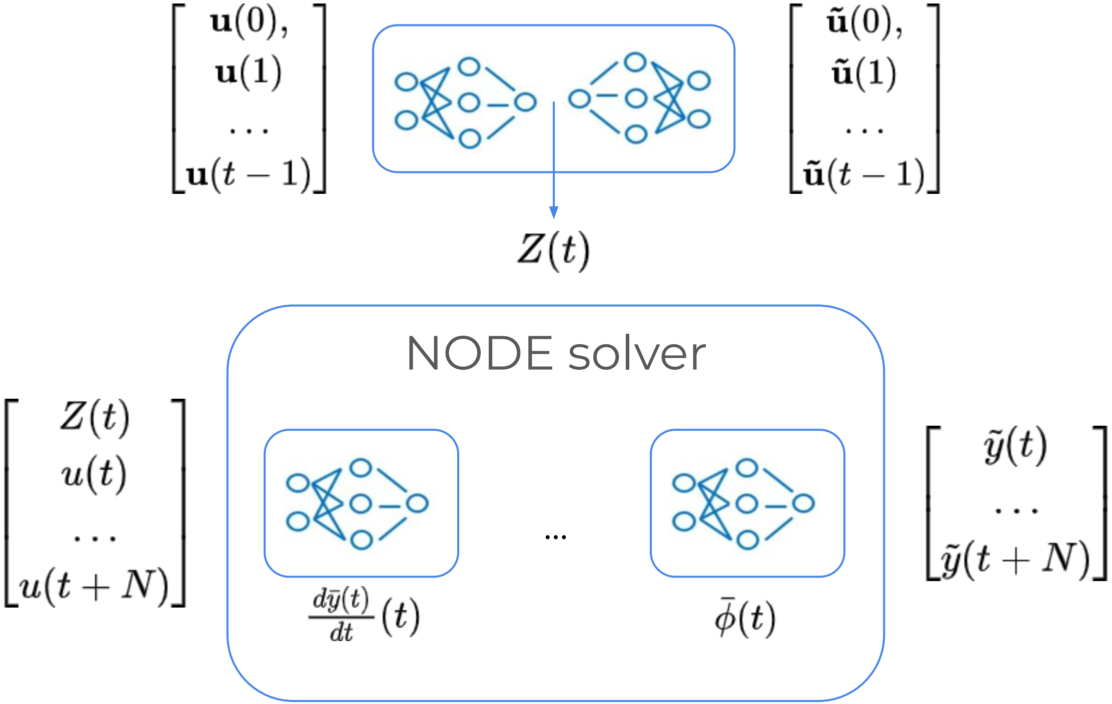


[6]



Neural ODE [7]

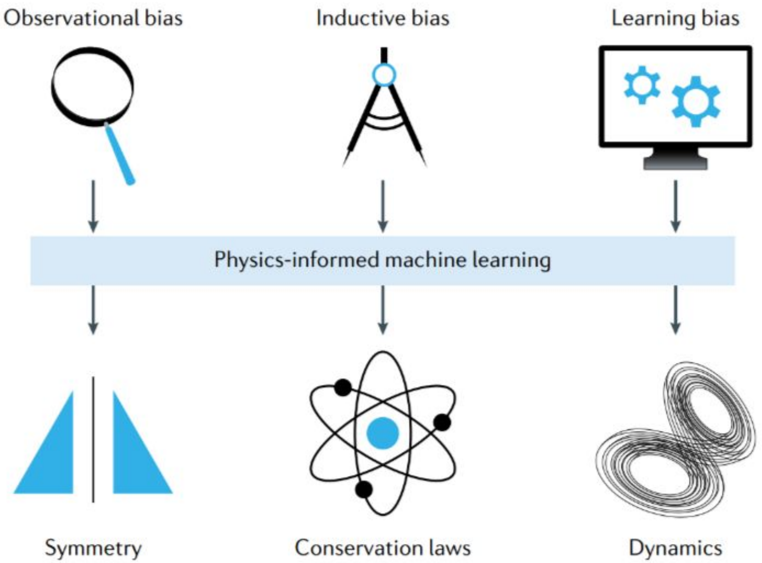
- ➔ Another possible way is to use the known problem statement as PDE and use a Neural solver (classic ODE solvers but using NN as functions)
  - ◆ Applications already in industry [8]
- ➔ Similar principle of Deep ONet but more suitable for real applications



Physics in ML models

➔ The pattern is always the same:
  - Make the model conceptually similar to the underlying physics
  - Add a term to the loss function to satisfy physics constraints

➔ We are basically adding additional information via physics laws and not directly data



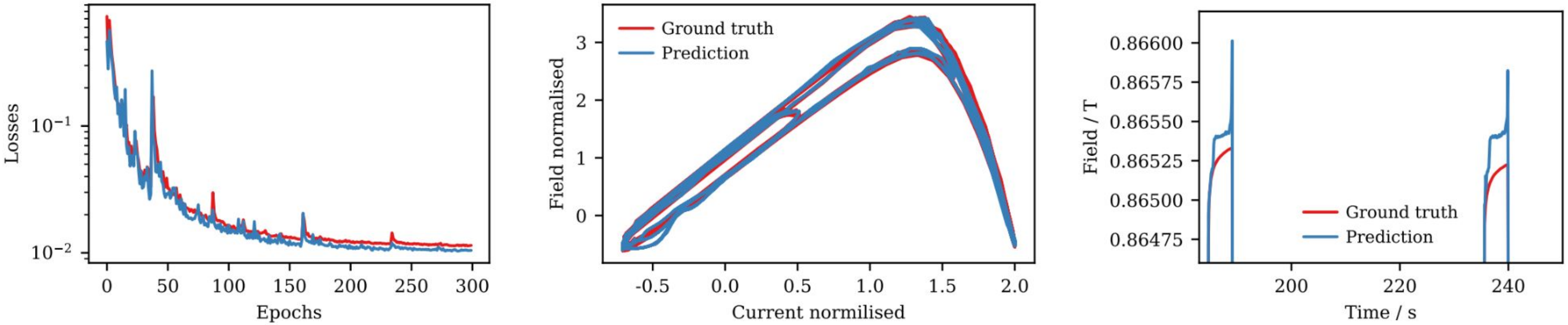
[2.1]

Some applications

Quadrupoles hysteresis prediction

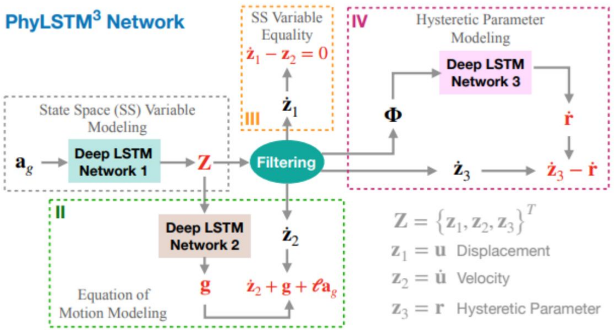
➔ First attempt using simple LSTM (as done for kicker temperature prediction)

➔ Very poor results! Dataset available not large enough and complicated dynamics



Hysteresis modelling

- ➔ Hysteresis is rather common in physics and many other fields (chemistry, biology, economics…)
- ➔ Modelling is rather challenging: main models Preisach and Bouc-Wen
- ➔ In [9], PINN applied to hysteresis modelling of behaviour of structures under seismic excitation
  - ◆ This was our inspiration => very similar problem but different system
- ➔ Here is the model used in [9]:



PhyLSTM for SPS quadrupole hysteresis

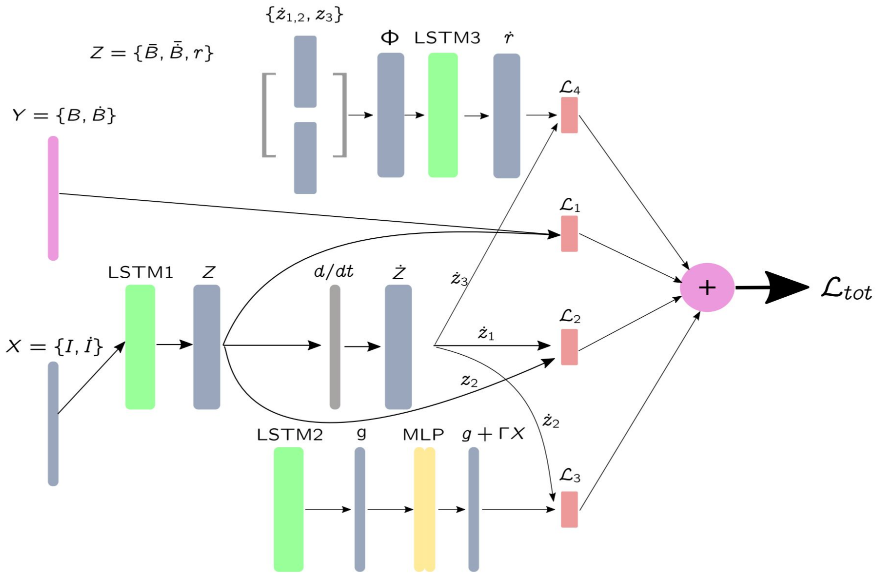
➔ A generic hysteretic model can be written as [10]:

$$a\ddot{y}(t) + b(y, \dot{y}) + r(y, \dot{y}, y(\tau)) = \Gamma x(t) \qquad \ddot{y} + g = \Gamma x$$

➔ Using input x = {I, dI/dt} and output y = {B, dB/dt}, we wrote our model and loss:

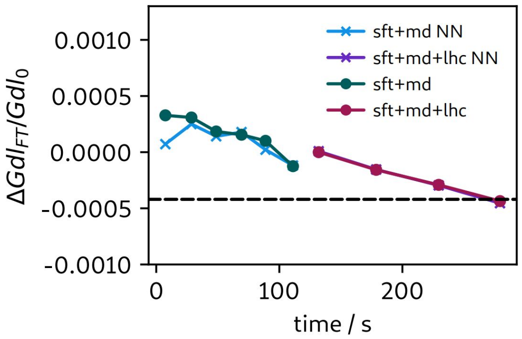
$$\mathcal{L}_1 = MSE(z_1(\theta_1) - y_1) + MSE(z_2(\theta_1) - y_2)$$
$$\mathcal{L}_2 = MSE(\dot{z}_1(\theta_1) - z_2(\theta_1))$$
$$\mathcal{L}_3 = MSE(\dot{z}_2(\theta_1) + MLP(g(\theta_1, \theta_2), x_1))$$
$$\mathcal{L}_4 = MSE(\dot{r}(\theta_1, \theta_3) - \dot{z}_3(\theta_1)); \dot{r} = f(\Phi); \Phi = \{\Delta z_2, r\}$$

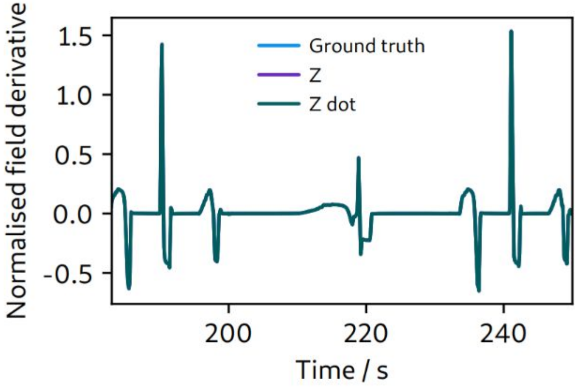
$$\mathcal{L}_{tot} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2 + \gamma\mathcal{L}_3 + \eta\mathcal{L}_4$$

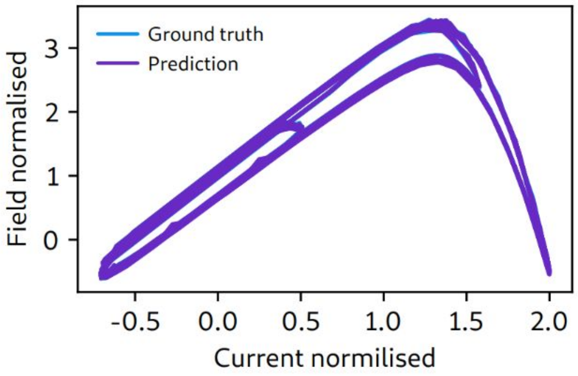


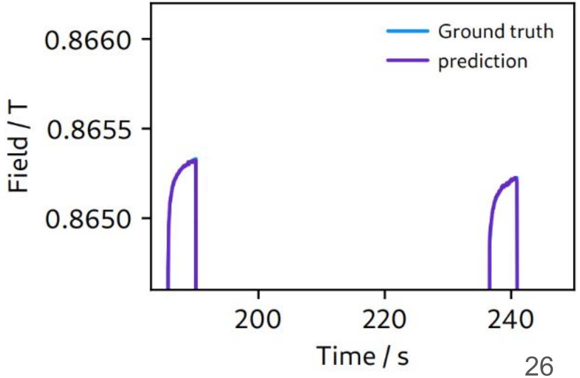
PINN for SPS quadrupole hysteresis

- ➔ Encouraging results, but very hard to train
- ➔ Evaluating pure data-driven models
- ➔ Just proof of concept: we now have Anton (PhD in CSS/DSB) actively working on this



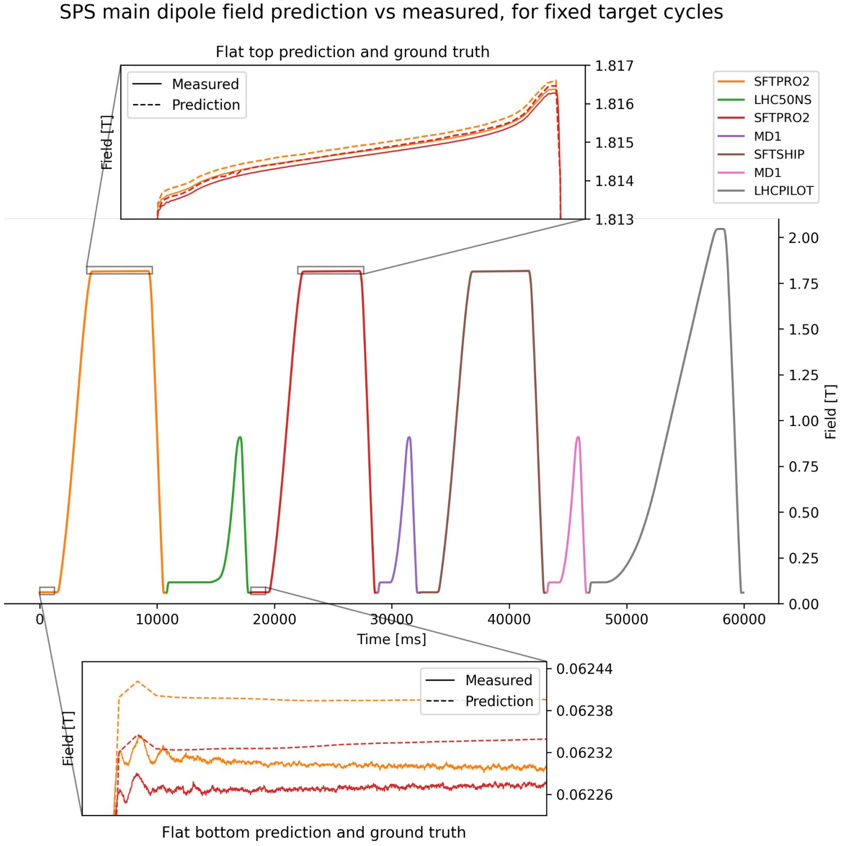






NNs for SPS main dipole hysteresis prediction

- ➔ PhyLSTM architecture trialed
  - ◆ Sub-gauss prediction accuracy very difficult to reach (~ 1e-5 T) for flat bottom
  - ◆ Hysteresis not perfectly captured even with additional data (1h varied operational cycles)
  - ◆ Bouc-Wen model used for physics loss does technically not account for rate-dependent effects (eddy currents)
- ➔ SOTA generic time series models like Temporal Fusion Transformer
  - ◆ Work better, but are
    - ● Very expensive to train
    - ● Requires vast amounts of data
    - ● Not guaranteed to generalize
- ➔ Future plans: PINNs
  - ◆ Augment existing architectures like TSMixer with physics loss
  - ◆ Choice of physics model highly important; Bouc-Wen model might not be sufficient



[A. Lu]

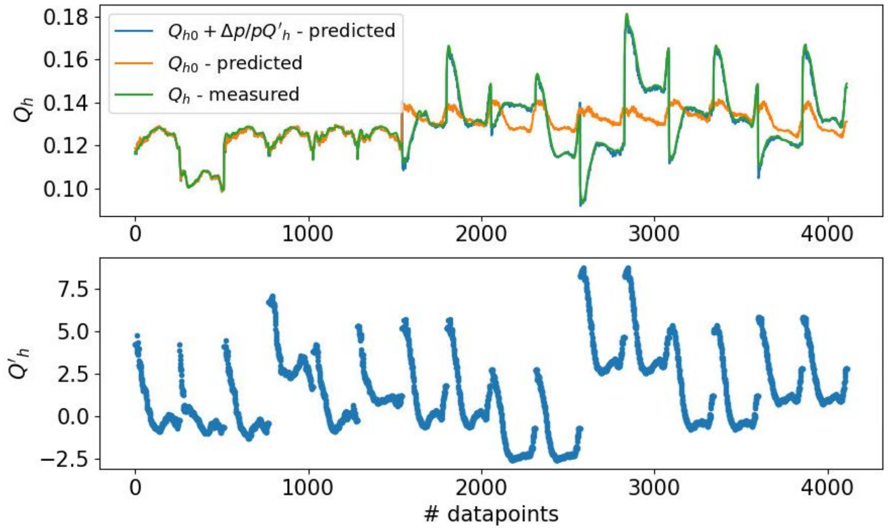
Tune and chromaticity settings

- ➔ We can measure tune and record all machine settings
  - ◆ Also save momentum offset
- ➔ Forcing (via loss function) the relationship between tune and chroma for given momentum offset => get chroma along the cycle
- ➔ We could then invert this model to be able to control tune and chroma on demand => normalizing flows?

Measured

$$F\begin{pmatrix} k_{QF} \\ k_{QD} \\ k_{SF1} \\ k_{SD1} \\ k_{SF2} \\ k_{SD2} \\ k_{S3} \\ B \\ \dot{B} \end{pmatrix} = \begin{pmatrix} Q_{\beta h} \\ Q_{\beta v} \\ Q'_h \\ Q'_v \end{pmatrix}$$

Estimated



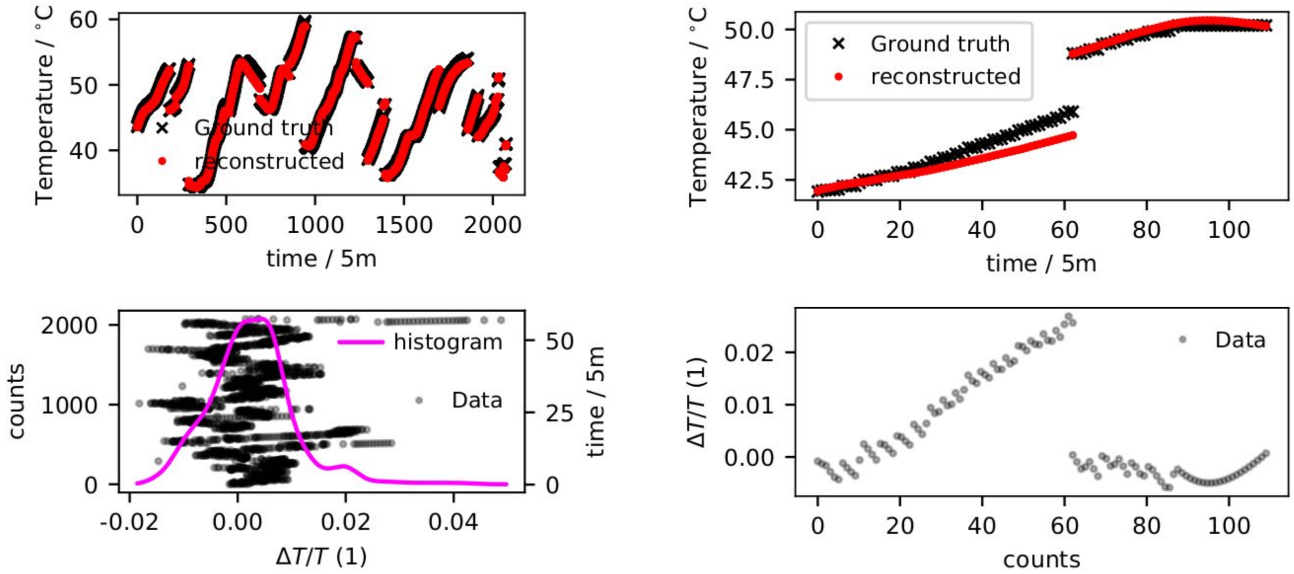
$$\mathscr{L} = \sqrt{\left[Q_{h_{true}} - (Q_{\beta h} + \frac{\Delta p}{p} Q'_h)\right]^2 + \left[Q_{v_{true}} - (Q_{\beta v} + \frac{\Delta p}{p} Q'_v)\right]^2}$$

LSTM for temperature prediction

➔ Two LSTM layers with 170 units with dropout layer with 50% probability, linear layer for the output prediction

- The loss function is calculated comparing the whole output sequence.

$$\hat{Y} = NN(X); \qquad X \in t(-40, 0]; \hat{Y} \in t[1, 30].$$



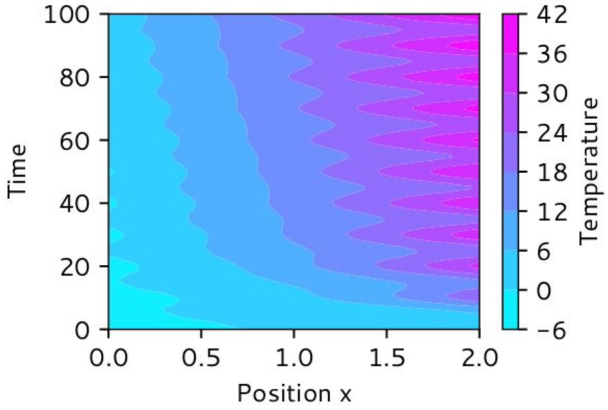
Adding physics information

- ➔ Bridge from pure data-driven model and pure physics model to PINN
- ➔ Solve heat equation with forcing term from beam-based measurements:
  - Power loss from beam induced heating

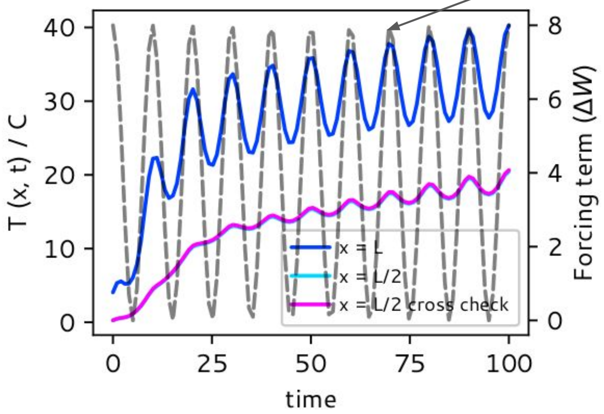
$$\Delta W = (f_0 e I_b N_b)^2 \sum_{k=-\infty}^{\infty} \left(|\Lambda(k\omega_0)|^2 \Re\left[Z_{||}(k\omega_0)\right]\right) \qquad \frac{d}{dt}T = \frac{\Delta W}{F_{cool} C_{th}}$$

  - Heat propagation inside the kicker and to temperature sensor:

$$\frac{\partial T}{\partial t} = \frac{k}{\rho C_p}\frac{\partial^2 T}{\partial x^2} + \frac{\Delta W(x,t)}{\rho C_p}$$

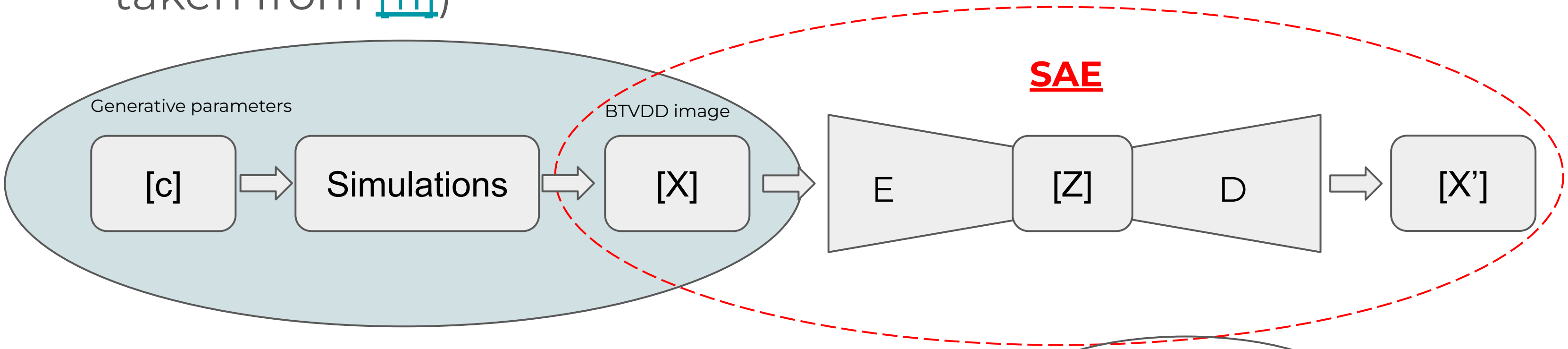


Never-seen forcing term



VAE for BTVD image reconstruction

➔ Special case of VAE => Supervised [Variational] Auto Encoder (idea taken from [11])

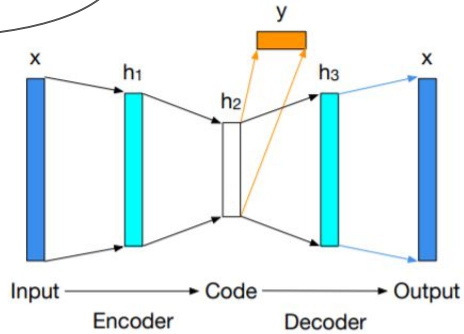


$$L_i(\theta,\phi) = -E_{z\sim q_\theta(z|x_i)}[\log_\phi(x_i|z)] + w_{KL}\,KL(q_\theta(z|x_i), p(z)) + w_g\,MSE(c, Z)$$

w_{KL} → =0

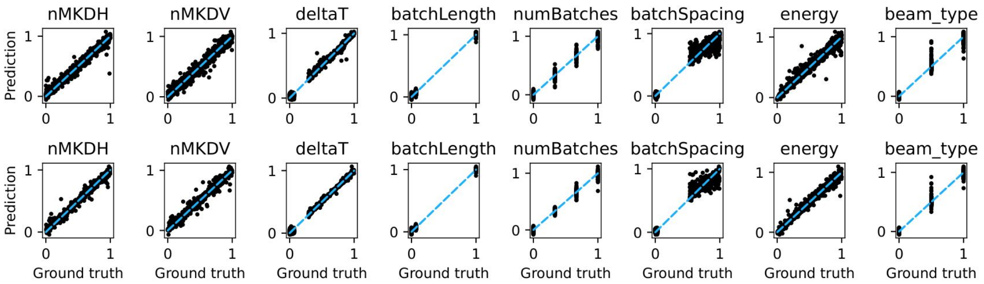
w_{KL} → !=0

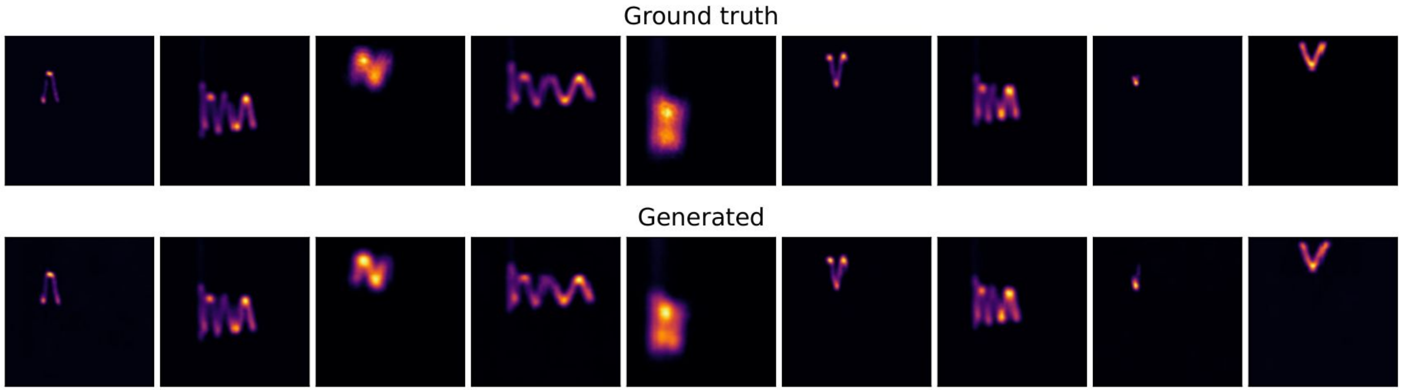
"Physics" loss



BTVDD image reconstruction in SPS

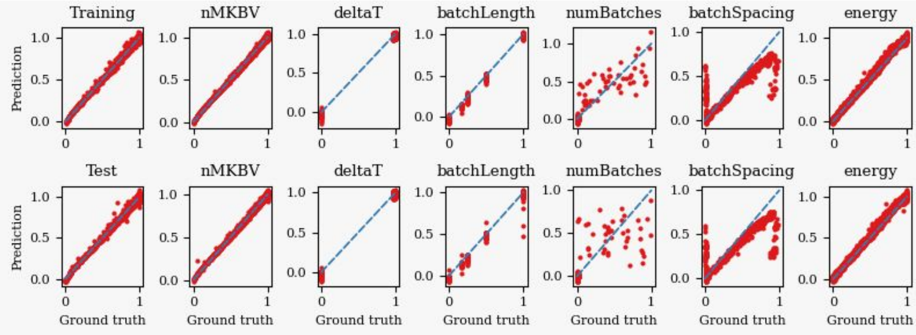
- ➔ LHC beam dump status reconstruction from beam images
- ➔ Here the most complicated part is to simulate different filling patterns
    - ◆ Number for batches very difficult for many single bunches
    - ◆ batch spacing very difficult for single bunches

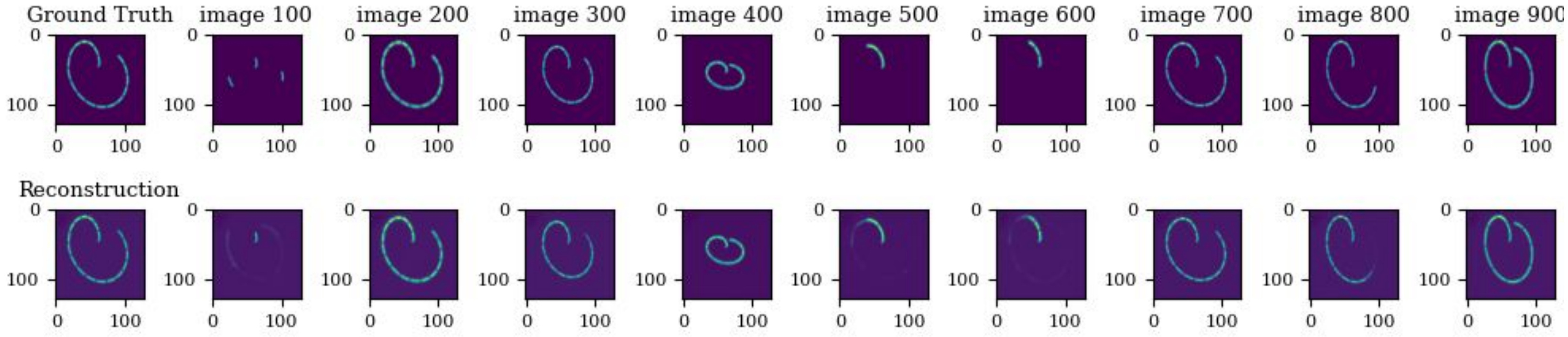




BTVDD image reconstruction in LHC

- ➔ LHC beam dump status reconstruction from beam images
- ➔ Here the most complicated part is to simulate different filling patterns
  - Number for batches very difficult for many single bunches
  - batch spacing very difficult for single bunches



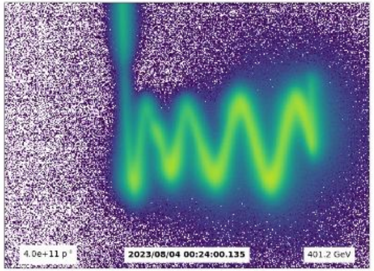


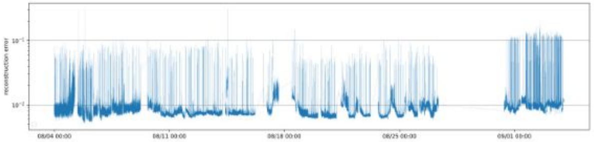
Models in operation

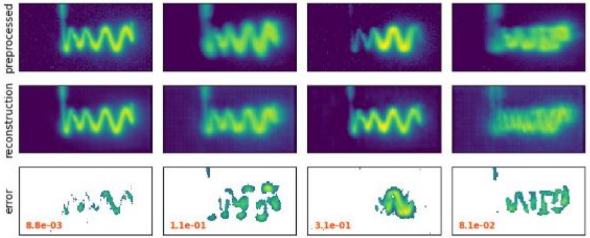
SBDS anomaly detection

Problem:

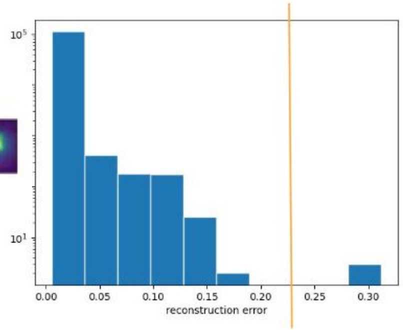
- ➔ Classify BTV images as good/anomaly
- ➔ Unlabeled dataset
- ➔ Most beam dumps are ok, i.e. dataset is biased towards good images.

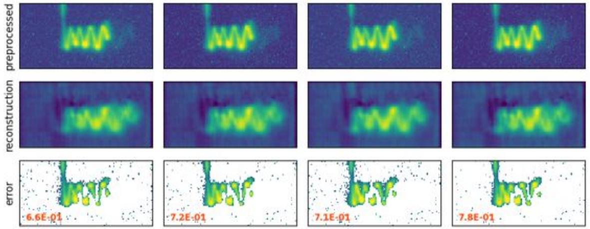






Solution: Autoencoder:





- ➔ Reconstruction error:

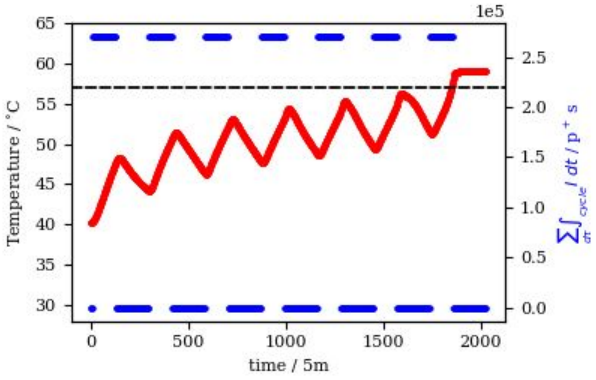
$$E = (1-\epsilon)e_w(x_{\text{good}}) + \epsilon e_w(x_{\text{bad}})$$

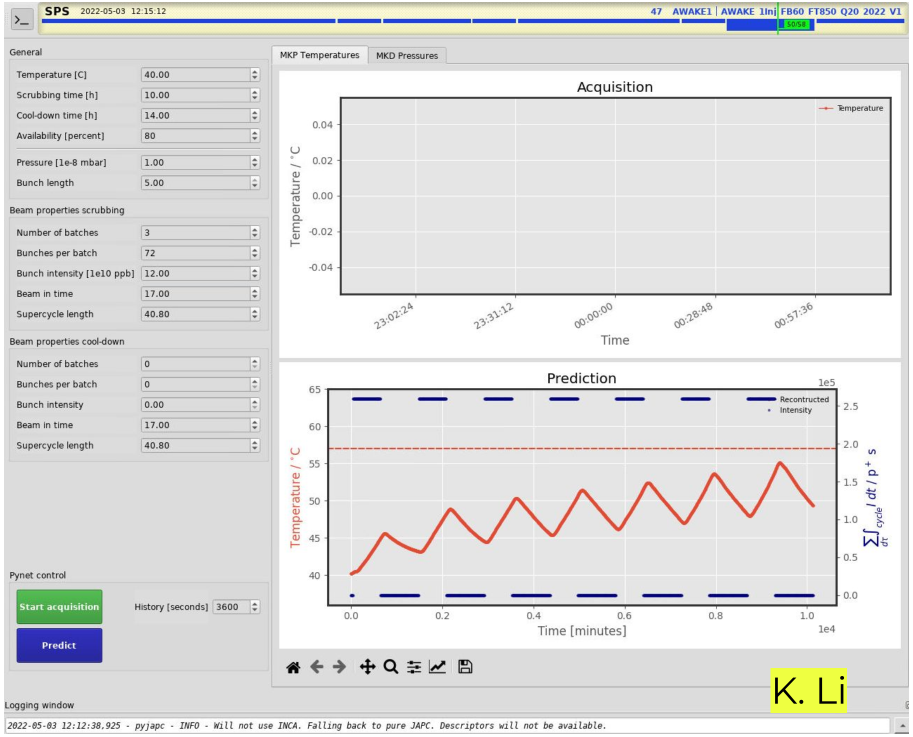
- ➔ High reconstruction error likely means an anomalous dump

[F. Hunh]

Summary and prediction

- Testing prediction on different scenarios
- Summary:
  - Model results very promising
  - Model ready and used in CCC to make estimation of time left for HI beams
  - Model not capable to extrapolate
- Need to include physics in the model...





K. Li

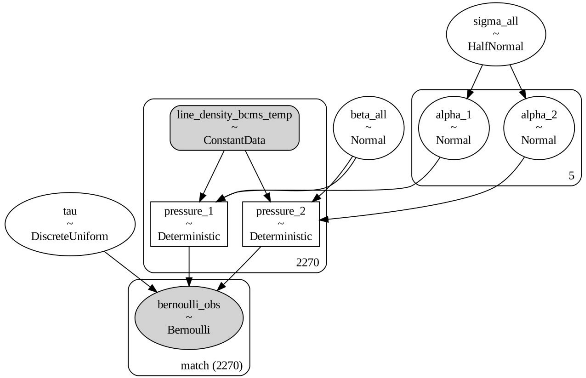
Summary and outlook

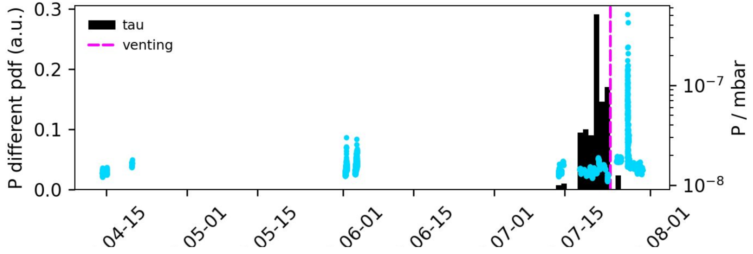
- ➔ We are working towards more automated and even more predictable machine operation
- ➔ Dealing with relatively small dataset and physics process partially known ⇒ Physics Informed machine learning
  - Rather simple to introduce physics awareness
  - Difficult to train
- ➔ First results look encouraging
  - In many cases still at PoC stage
- ➔ Model deployed only data driven so far
- ➔ Looking at other possible applications for PINN:
  - Optimisation of septa design via PINN-surrogate
  - Replacement for PDE solvers for mechanical design or design optimisation

Thanks!

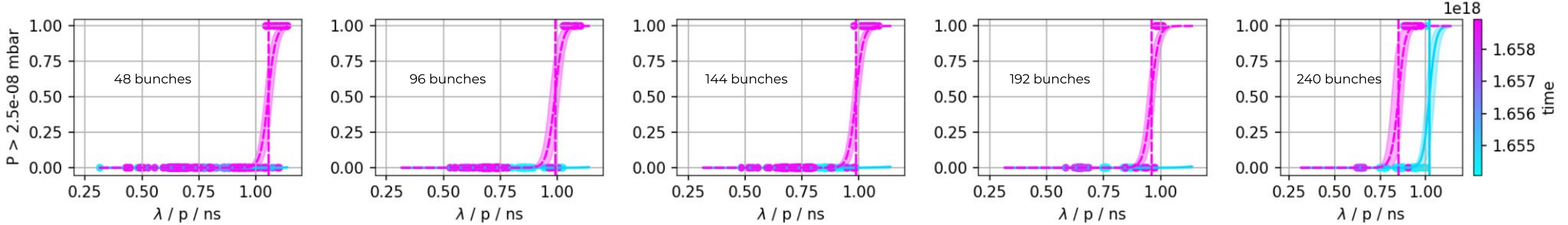
MKDH pressure prediction

- ➔ We can transform the problem to predict the probability of a vacuum spike give beam parameters
- ➔ Pure Bayesian probabilistic model: used pyMC to build a model that respect physics behind vacuum response
- ➔ Such a model can also show us if the element is showing conditioning with time





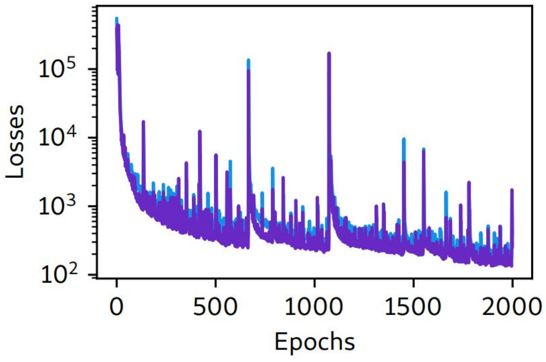
Number of batches →



PhyLSTM for SPS quadrupole hysteresis `

➔ After many attempts, we managed to train successfully one PhyLSTM for hysteresis prediction

- Not fully optimised yet
- Not enough data to make a proper general model for SPS quadrupoles
- Hyperparameters not tuned yet

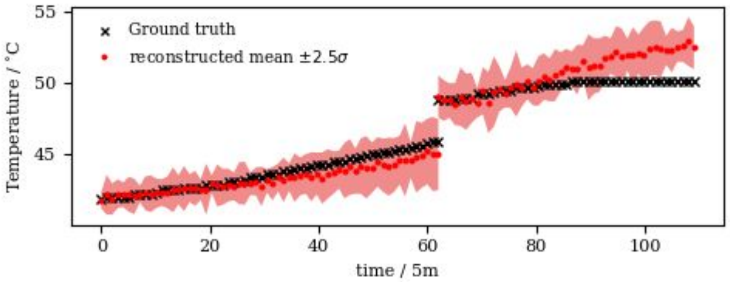


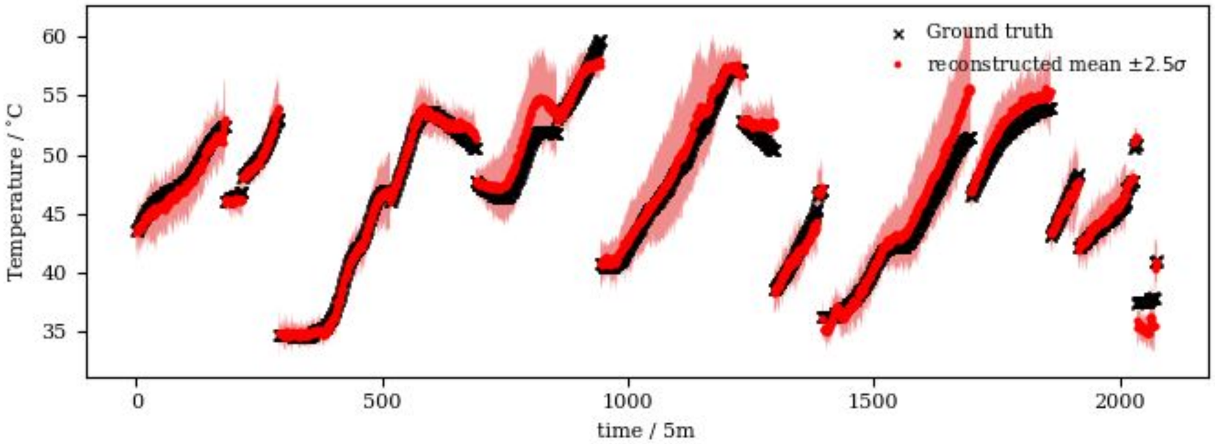
PhyLSTM[3]

```
(relu): LeakyReLU(negative-slope=0.01)
(lstm0): LSTM(1, 350, num-layers=3, batch-first=True, dropout=0.2)
(fc0): Linear(in-features=350, out-features=175, bias=True)
(fc01): Linear(in-features=175, out-features=3, bias=True)
(gradient): GradientTorch()
(lstm): LSTM(3, 350, num-layers=3, batch-first=True, dropout=0.2)
(fc1): Linear(in-features=350, out-features=175, bias=True)
(fc11): Linear(in-features=175, out-features=1, bias=True)
(lstm3): LSTM(2, 350, num-layers=3, batch-first=True, dropout=0.2)
(fc2): Linear(in-features=350, out-features=175, bias=True)
(fc21): Linear(in-features=175, out-features=1, bias=True)
(g-plus-x): Sequential(
(0): Linear(in-features=2, out-features=350, bias=True)
(1): ReLU()
(2): Linear(in-features=350, out-features=1, bias=True))
```

LSTM model for MKP: results

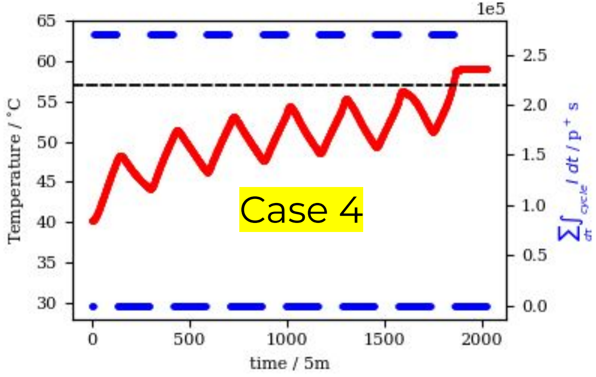
- ➔ Trained model repreduced training and validation data set almost perfectly
  - ◆ Trained on max sequence of 30 steps and capable to extend to ~100 with reasonable errors
  - ◆ Error in the order of a couple of degrees on test dataset
- ➔ Bayesian version looking also promising





Prediction for 2021 scrubbing

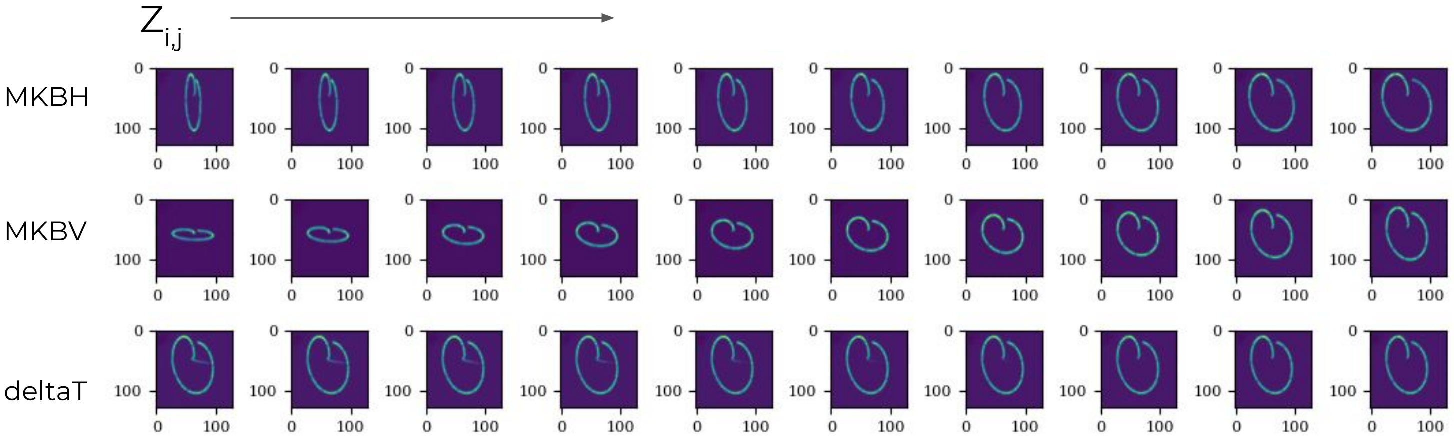
- ➔ Testing the prediction on 10/14h scrubbing, with 288x1.5e11 p at 100% availability...we should reach the **60°C in the first 2 runs of 10h**!!
- ➔ Here we really see this as the model is not capable to extrapolate...
- ➔ Both models saturates at 60°C (since no data beyond this in our training set) and cannot predict correctly cooldown after 57°C as data on that either...



Inputs	c1	c2	c3	c4
$I_{b,\ ns}$(e11)	1.5	1.5	1.5	1.5
N_b(#)	288	288	216	144
Av	1.0	1.0	1.0	1.0
b_l(s: BQM)	5e-9	5e-9	5e-9	5e-9
I_{off}(e11/cycle)	0.0	0.0	0.0	0.0
T_0(°C)	40	40	40	40
T_{bin}(min)	5	5	5	5
T_{cycle}(s)	17	17	17	17
T_{SC}(s)	40.8	40.8	40.8	40.8
T_{on}->[h]	[10] * 8	[6] * 8	[8] * 7	[10] * 7
T_{off}->[h]	[14] * 8	[18] * 8	[16] * 7	[14] * 7

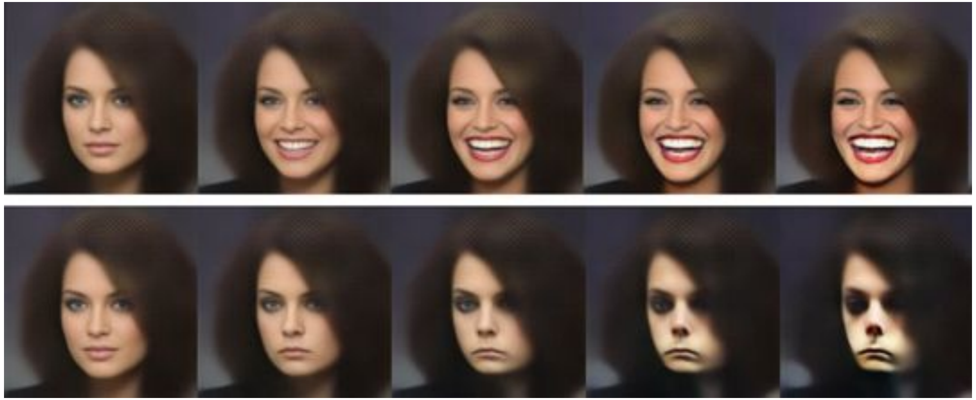
Latent space scan

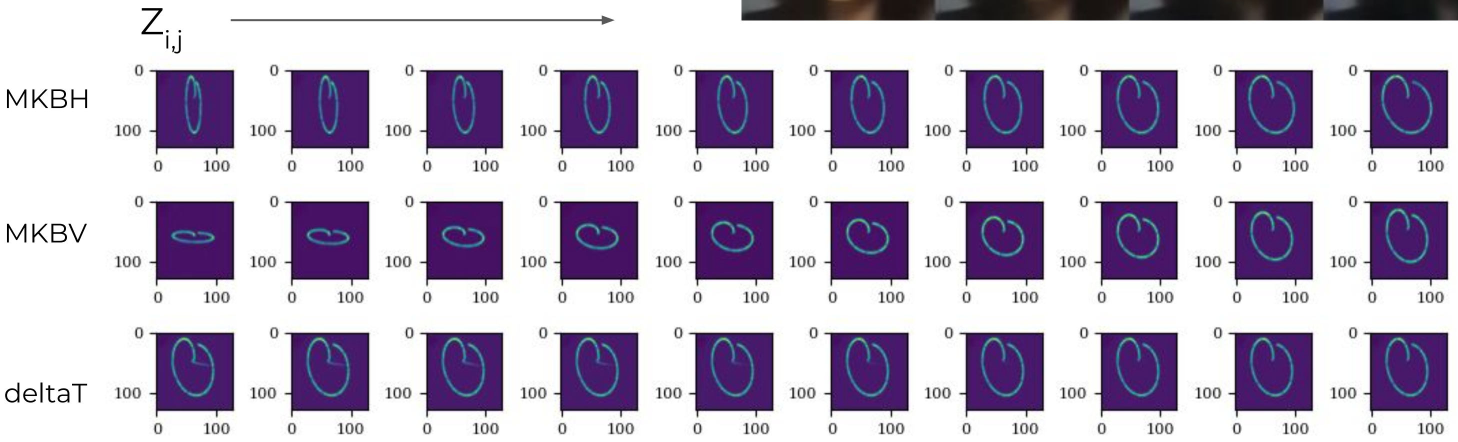
- ➔ With this architecture, we can generate BTVDD images from generative parameters (number of kickers...) using the decoder by itself
- ➔ Orthogonal scan possible



Latent space scan

- ➔ With this architecture, we can generate BTVDD images from generative parameters (number of kickers...) using the decoder by itself
- ➔ Orthogonal scan possible





Deploy on real data

➔ Of course the final goal is to predict real images...

➔ Using both generative parameters and image reconstruction score, anomalous case found!