Neural-Network-based Surrogate Simulator for Particle Accelerator with High Dimensional Control Settings

Emmanuel Goutierre / Hayg Guler

H. Guler[1], C. Bruni[1]
J. Cohen[2], M. Sebag[2]

[1]Laboratoire de Physique des 2 Infinis Irène Joliot-Curie (IJCLab)

[2]Laboratoire Interdisciplinaire des Sciences du Numérique (LISN)

Tuesday 28th November 2023

Introduction

Particle Accelerator Physics

- understand machine data could get tricky, much rely on simulations
- High fidelity simulations, often slow
- Machine (during commissioning) is not at the optimum, tuning might take time.
- Need to adapt the simulation on different working points.

Machine Learning

- Fast-executing
- Data Driven
- Questions on its guarantee
- Training time / need GPU

Needs : Data

- From accurate Simulations
- From Real-Time beam diagnostics / controls

Table of Contents

Table of Contents

How does a surrogate model work?

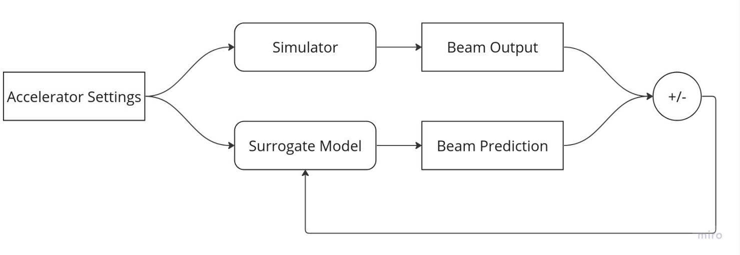


Figure: Training of a Surrogate Model

Why Surrogate Models of Particle Accelerator Simulator?

General motivation concerning needs for surrogate models for particle accelerators.

Fast Execution

- ms vs. several minutes

Optimization

- Offline & Online

Real-time Feedback

- Runnable in a control room during operations

Table of Contents

ThomX: A Compact Compton Source

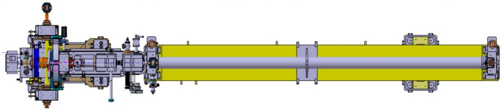


Figure: Linac of ThomX.

ThomX

- X-ray source by Compton backscattering
- Compact Accelerator $(70\mathrm{m}^2)$
- In commissioning at the IJCLab since May 2021

Linac

- Accelerate the electron beam up to 50 MeV

Goal

Use machine learning to tackle the problem of adjusting the Linac parameters to fulfill the beam requirements for the transfer line.

Accelerator Tuning

$\mathcal{A}$: Controllable Parameters

- 15 controllable parameters
  - Laser position and size
  - Gun and Cavity phase and field
  - Solenoid Fields
  - Steerer Fields
  - Quadrupoles Fields

Accelerator Tuning

$\mathcal{A}$: Controllable Parameters

- 15 controllable parameters
  - Laser position and size
  - Gun and Cavity phase and field
  - Solenoid Fields
  - Steerer Fields
  - Quadrupoles Fields

$\mathcal{B}$: Hidden Parameters

- Mechanical Misalignment
- Unknown initial particle distribution
- Slow drift of electromagnetic elements

Accelerator Tuning

$\mathcal{A}$: Controllable Parameters

- 15 controllable parameters
  - Laser position and size
  - Gun and Cavity phase and field
  - Solenoid Fields
  - Steerer Fields
  - Quadrupoles Fields

$\mathcal{O}$: Observables

- 17 Observables
  - Position and Charge at BPMs
  - Charge at ICTs
  - Position and Size at Screen
  - Charge at Faraday Cup

$\mathcal{B}$: Hidden Parameters

- Mechanical Misalignment
- Unknown initial particle distribution
- Slow drift of electromagnetic elements

Accelerator Tuning

$\mathcal{A}$: Controllable Parameters

- 15 controllable parameters
  - Laser position and size
  - Gun and Cavity phase and field
  - Solenoid Fields
  - Steerer Fields
  - Quadrupoles Fields

$\mathcal{B}$: Hidden Parameters

- Mechanical Misalignment
- Unknown initial particle distribution
- Slow drift of electromagnetic elements

$\mathcal{O}$: Observables

- 17 Observables
  - Position and Charge at BPMs
  - Charge at ICTs
  - Position and Size at Screen
  - Charge at Faraday Cup

F: Objective function

- Quality of the beam
- Function of (A, B)

Accelerator Tuning

$\mathcal{A}$: Controllable Parameters

- 15 controllable parameters
  - Laser position and size
  - Gun and Cavity phase and field
  - Solenoid Fields
  - Steerer Fields
  - Quadrupoles Fields

$\mathcal{B}$: Hidden Parameters

- Mechanical Misalignment
- Unknown initial particle distribution
- Slow drift of electromagnetic elements

$\mathcal{O}$: Observables

- 17 Observables
  - Position and Charge at BPMs
  - Charge at ICTs
  - Position and Size at Screen
  - Charge at Faraday Cup

F: Objective function

- Quality of the beam
- Function of (A, B)

Goal

- Optimize : find set of **parameters** (A) depending on **hidden parameters** (B) to get minimal **objective function** (F) with the aid of **observable**$(\mathcal{O})$
- Classical way : manual tuning, heavy load on expert

Table of Contents

"Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems"[1] (Apr. 2020)

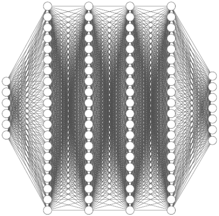


Figure: Neural Network architecture for the surrogate model of the AWA Linac.

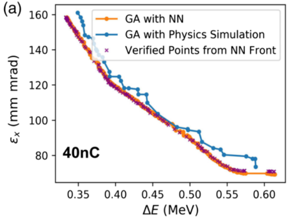


Figure: Optimization performed with the surrogate model.

Analysis

- Faster optimization than with direct call to simulator
- Only 6 input variables and 7 outputs on a narrow domain
- Optimization performed only on simulations

[1] Edelen et al., "Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems".

"An adaptive approach to machine learning for compact particle accelerators"[2] (Sept. 2021)

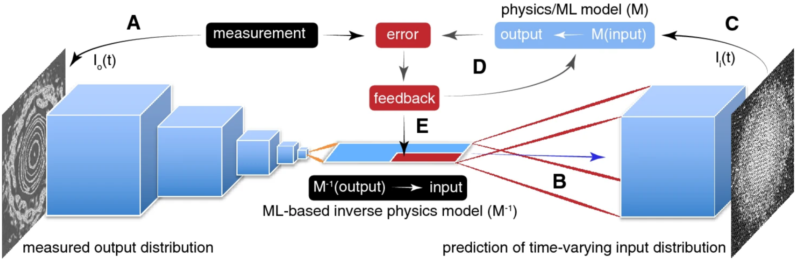


Figure: Neural Network architecture for the Hires UED.

Analysis

- Online tuning, adaptive to time varying perturbation.
- Use only 2D projections of the beam
- Need for lot of high quality experimental data

[2] Scheinker et al., "An adaptive approach to machine learning for compact particle accelerators".

Table of Contents

Table of Contents

Multi-Layer Perceptron: A First Model

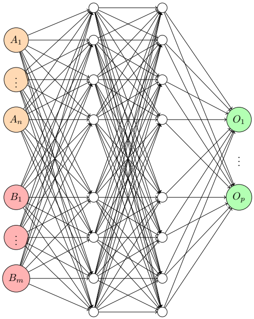


Figure: MLP as a surrogate model of a Linac

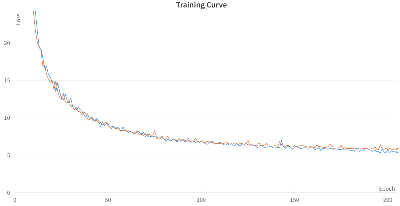


Figure: Training Curve

Multi Layer Perceptron

- Stack all inputs and outputs
- 10k simulations sampling $\mathcal{A}$ and $\mathcal{B}$
- Minimization of the L2 loss

Physics-aware: Cutting the non-causal links

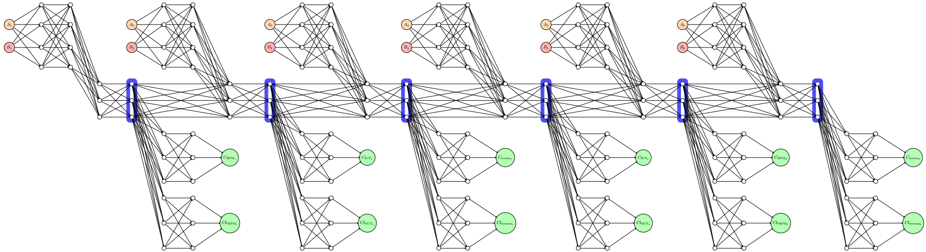


Figure: LinacNet with 6 modules corresponding to 6 diagnostic stations on the Linac

LinacNet

- Split input and output according to their position in the Linac
- Neural Network Architecture reflecting a Linac architecture
- Each Module models one Diagnostic (could be real or virtual)

Table of Contents

PointNet as a Beam Representation Network

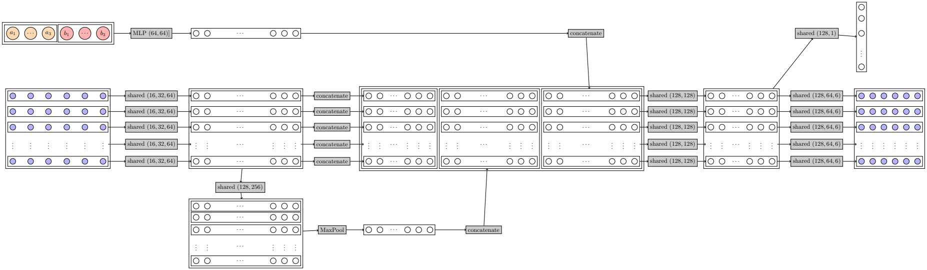


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

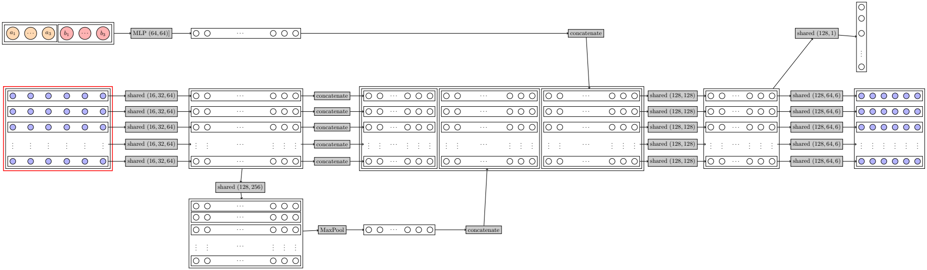


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

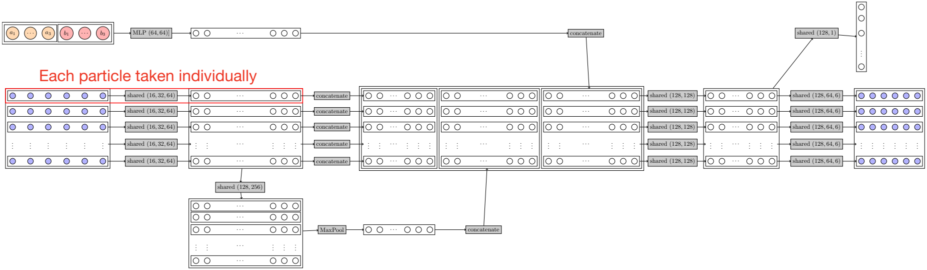


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

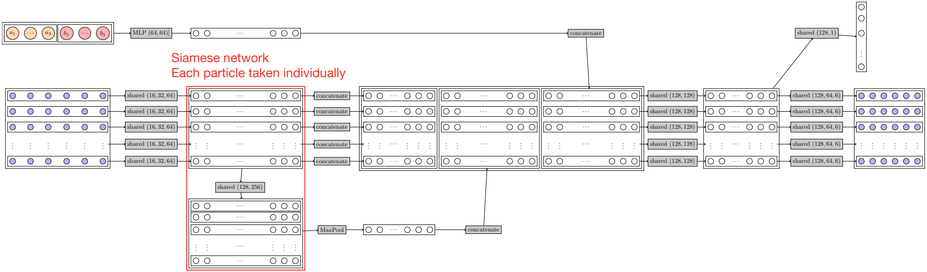


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

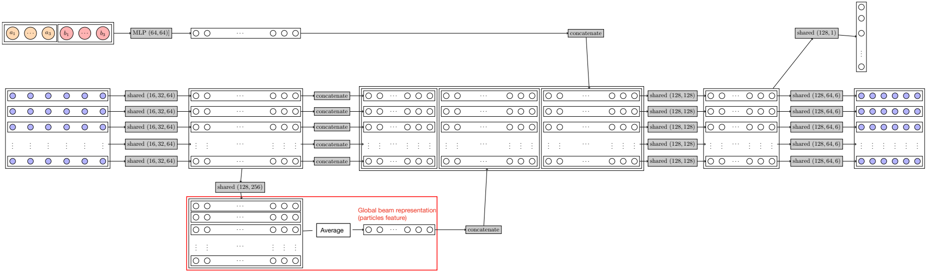


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

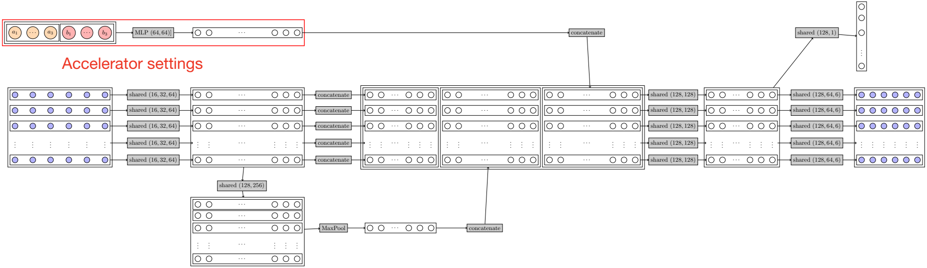


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

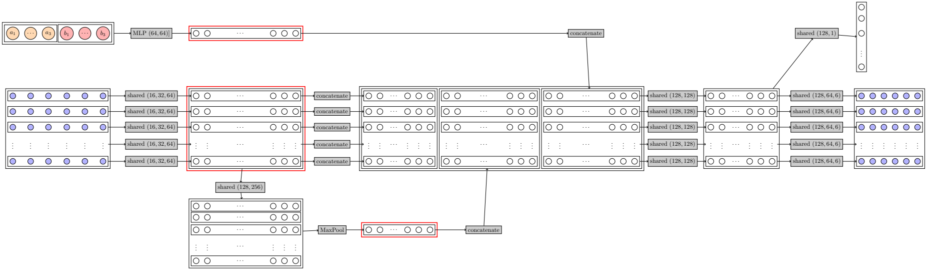


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” (CVPR 2017)

PointNet as a Beam Representation Network

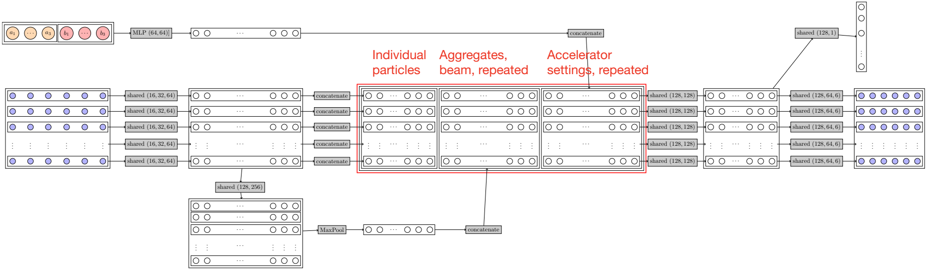


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

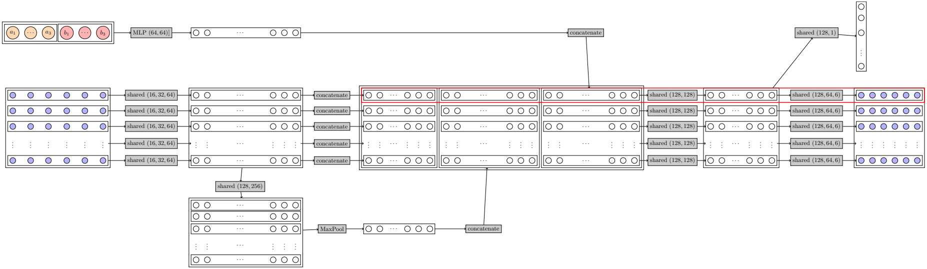


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

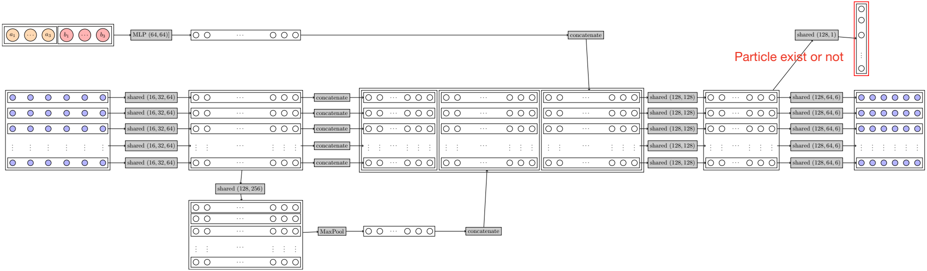


Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

PointNet as a Beam Representation Network

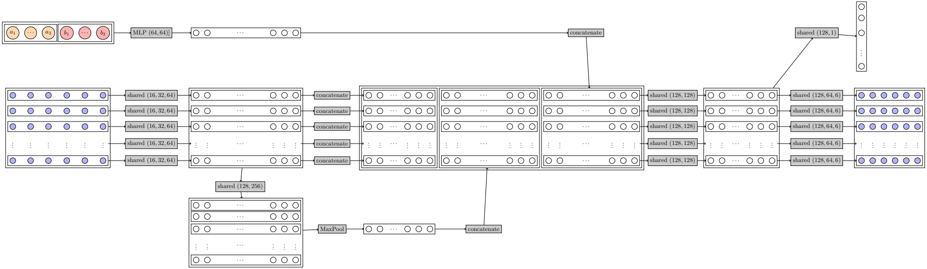


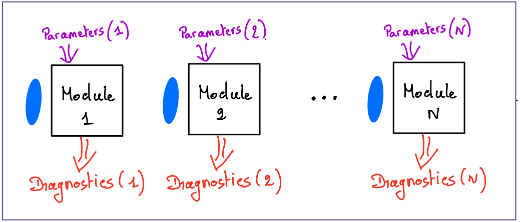
Figure: One module of ThomNet

ThomNet

- Track the full distribution of particles
- Inspired by Qi et al., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation" (CVPR 2017)

Table of Contents

Accelerator as a sequence of modules



- We divide our accelerator in a sequence of sub-parts
- Each part could contain controls / measurements (real or virtual)
- Learning a full machine could be complicated, costly

Good for

- transfer to a real machine
- optimize : could be done by part / module
- Retrain locally due to drift in the data
- Address larger machines

Sequential Network as a Multi-Objective Optimization

- General question in machine learning
- how to learn a sequence of models, tasks ?
- Could be heterogeneous : classification, regression, etc
- Conflicting between modules could deteriorate the global loss

- Independent Errors : $Err_{i,i+1}\left(d_i, d_{i+1}, a; \theta\right) = l\left(f_{i,i+1}\left(d_i, a; \theta\right), d_{i+1}\right)$
- End-to-End Errors : $Err_{0,i}\left(d_0, d_i, a; \theta\right) = l\left(f_{0,i}\left(d_0, a; \theta\right), d_i\right)$

Scalarization of the Multi-Objective Loss

$$\mathcal{L}_w\left(d, a; \theta\right) = \sum_{i=1}^{N} w_{i-1,i} Err_{i-1,i}\left(d_{i-1}, d_i, a; \theta\right) + w_{0,i} Err_{0,i}\left(d_0, d_i, a; \theta\right)$$

One example of learning a sequence : MGDA[3]

- Dynamic weighting of the module that moderates conflicting loss between modules

$$w^* = \arg\min_{w} \mathcal{L}_w, \qquad w > 0, \qquad \sum_{i=1}^{N} w_{i-1,i} + w_{0,i} = 1$$

Properties

- Common descent direction to all objectives
- Stop when encountering a Pareto-invariant point

[3] Sener and Koltun, "Multi-Task Learning as Multi-Objective Optimization".

Table of Contents

Numerical Results

The best model achieves results comparable with the diagnostic station accuracy.

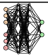
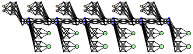
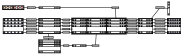
Architecture	BPM	ICT	YAG	ICT	BPM	YAG
FeedForward 	$776\mu\mathrm{m}$	$1084\mu\mathrm{m}$	$1692\mu\mathrm{m}$	$1106\mu\mathrm{m}$	$1261\mu\mathrm{m}$	$1554\mu\mathrm{m}$
LinacNet 	$198\mu\mathrm{m}$	$254\mu\mathrm{m}$	$541\mu\mathrm{m}$	$618\mu\mathrm{m}$	$719\mu\mathrm{m}$	$913\mu\mathrm{m}$
ThomNet 	$\mathbf{178\mu m}$	$\mathbf{134\mu m}$	$\mathbf{247\mu m}$	$\mathbf{224\mu m}$	$\mathbf{258\mu m}$	$\mathbf{336\mu m}$

Table: MAE of the position. The accuracy of the BPM is $\sim 100\mu\mathrm{m}$

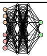
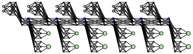
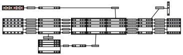
Architecture	BPM	ICT	YAG	ICT	BPM	YAG
FeedForward 	$176\mathrm{pC}$	$177\mathrm{pC}$	$167\mathrm{pC}$	$91\mathrm{pC}$	$91\mathrm{pC}$	$91\mathrm{pC}$
LinacNet 	$28\mathrm{pC}$	$28\mathrm{pC}$	$29\mathrm{pC}$	$34\mathrm{pC}$	$34\mathrm{pC}$	$35\mathrm{pC}$
ThomNet 	$\mathbf{8pC}$	$\mathbf{9pC}$	$\mathbf{9pC}$	$\mathbf{8pC}$	$\mathbf{8pC}$	$\mathbf{8pC}$

Table: MAE of the charge. The accuracy of the ICT is $\sim 10\mathrm{pC}$

Distributions

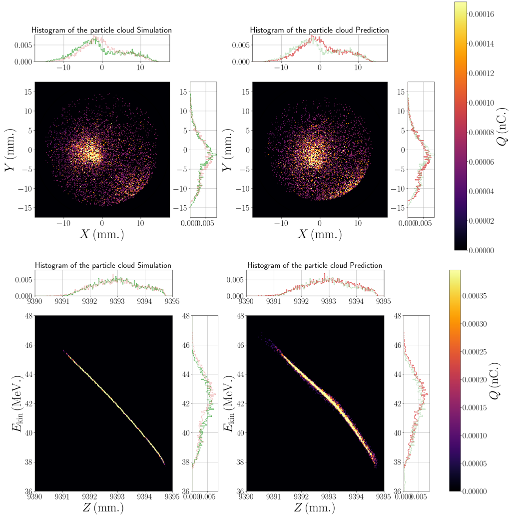


Figure: Comparison between the projection of the simulated beam (left) and predicted beam (right) on the transverse and longitudinal space.

Table of Contents

Perspectives

Results

- Reflecting the physical constraints in the neural architecture speed up the training and gives better results
- Precision of the same orders as the diagnostics installed on ThomX

Challenges

- Training of a modular model
- Performance for the optimization task to be tested

Questions?