IGFAE INSTITUTO GALEGO DE FÍSICA DE ALTAS ENERXÍAS
25 1999 2024
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
AGENCIA ESTATAL DE INVESTIGACIÓN
EXCELENCIA MARÍA DE MAEZTU 2024-2029

Automatic Optimization of a Parallel-Plate Avalanche Counter with Optical Readout

María Pereira Martínez, Pietro Vischia, Xabier Cid Vidal

2nd Computing Challenges Workshop, A Coruña

October 4th 2024

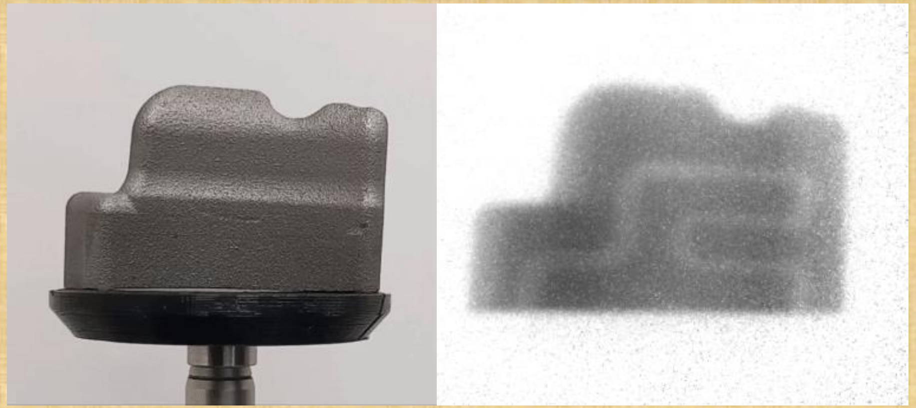
USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
XUNTA DE GALICIA
CENTROS
Cofinanciado por la Unión Europea
Fondos Europeos

Neutron tomography

Tomography by **emission and detection of neutrons for non-destructive tests (NDT).**

High penetration, effective for dense materials like metals and alloys.

Metal industry, additive manufacturing, border security…

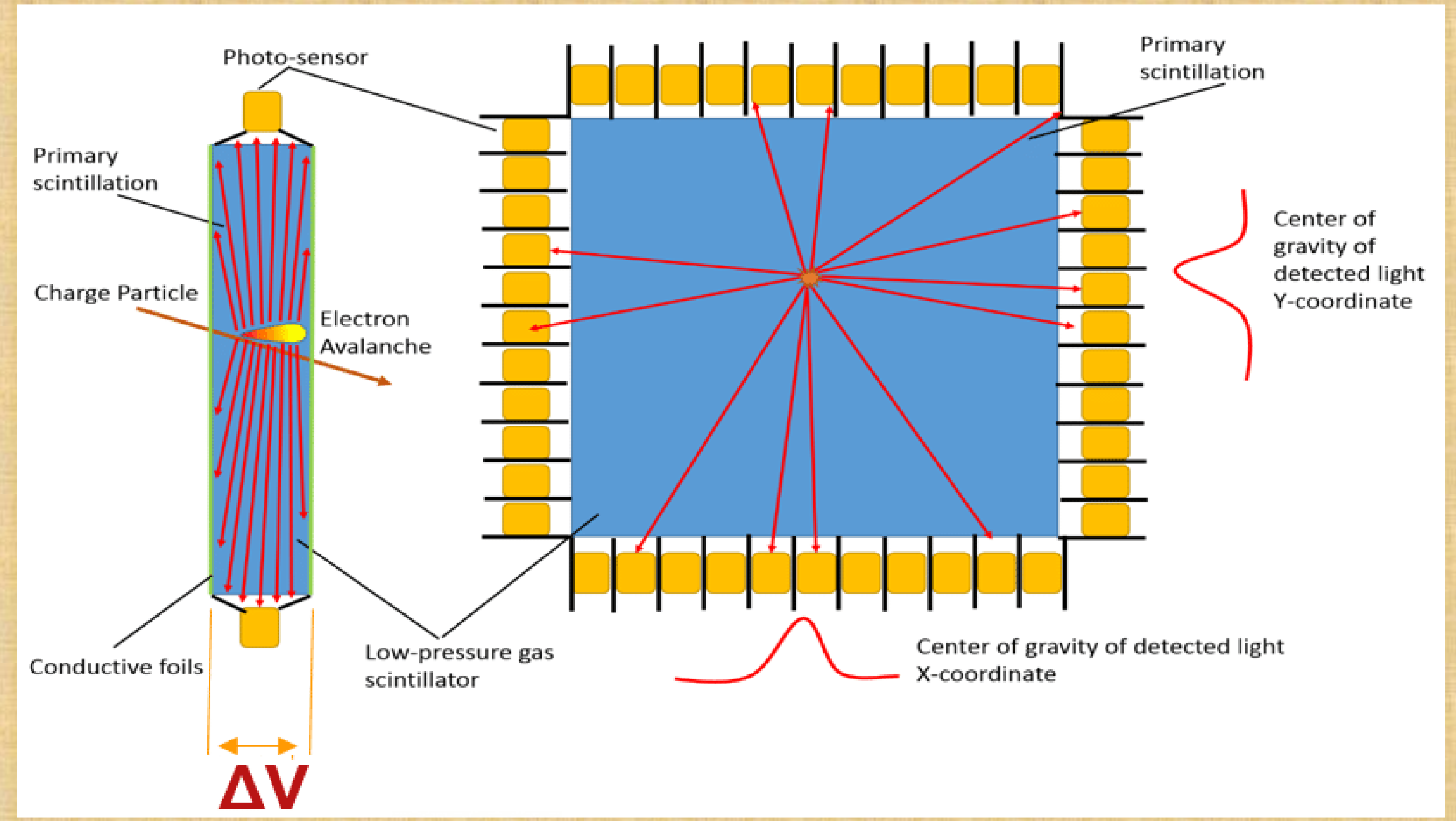


Images from Neutron Insights website

What do we want to do?

Optimize the neutron tomography system but… **where do we start?**

Optical Parallel-Plate Avalanche Counter



- Parallel-plates filled with a **high electroluminiscense yield gas** (CF4).
- Charged particles crossing active volume ionize medium and produce an avalanche.
- Electroluminiscense light detected by 4 arrays of small, collimated silicon photomultipliers (SiPMs).

Image from 1808.05882

Optical Parallel Plate Avalanche Counter

Geant4 model of a 10×10 cm^2 O-PPAC, 33 SiPMs per array

Example of an event triggered by an impining alpha particle:

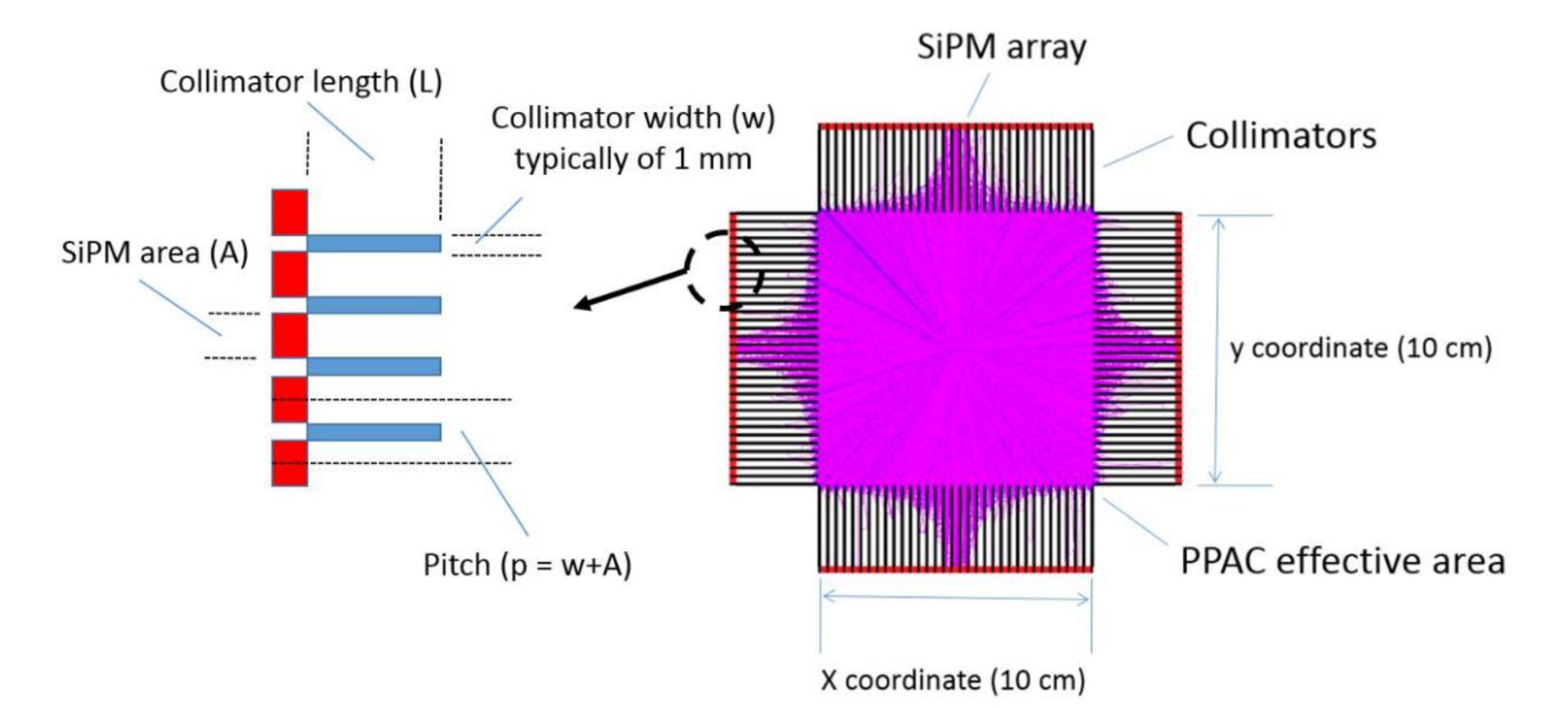
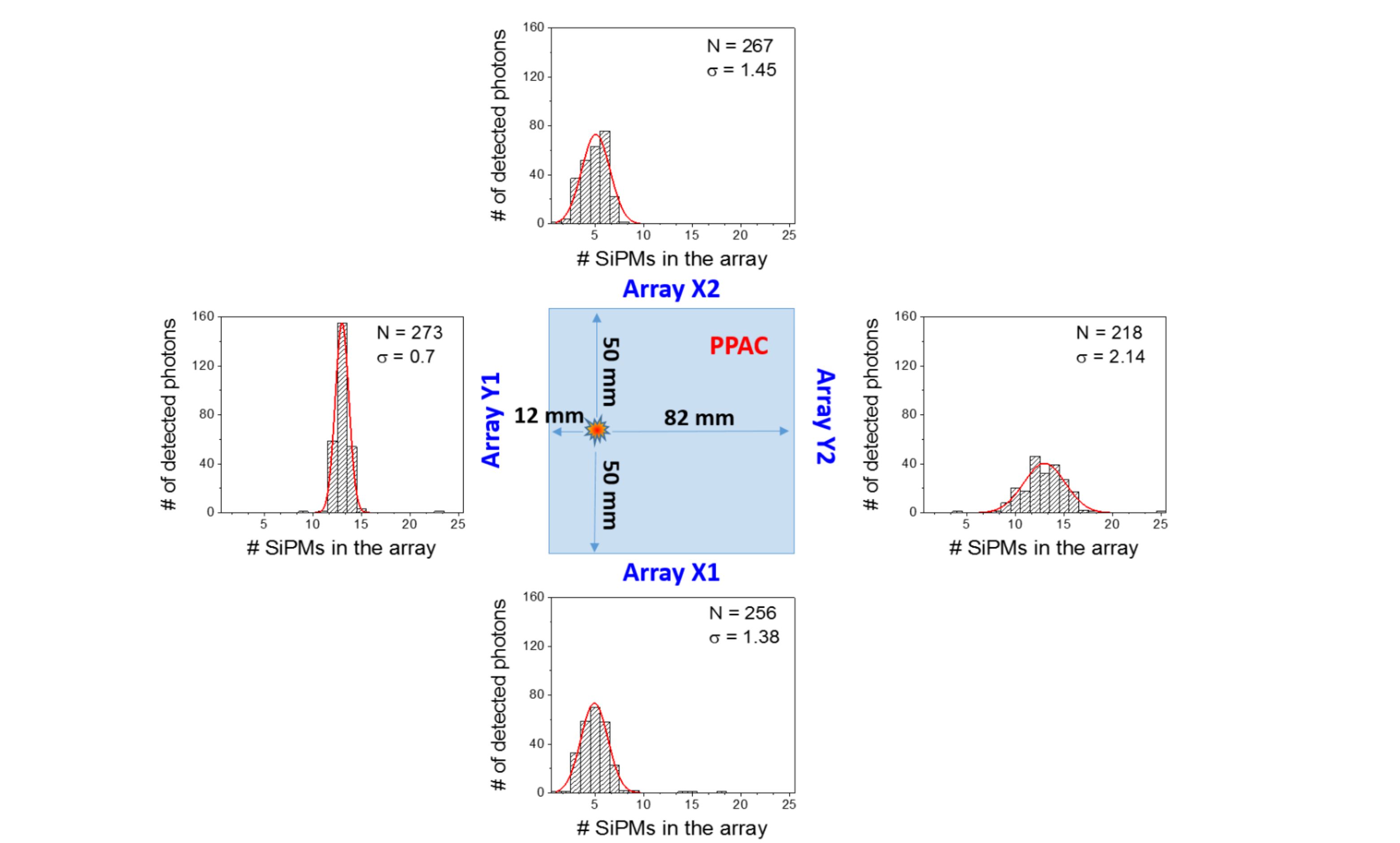


Image from 1808.05882

Reconstruction of the position

Reconstructed position ($\hat{x}$, $\hat{y}$) obtained from the number of photons detected in each SiPM

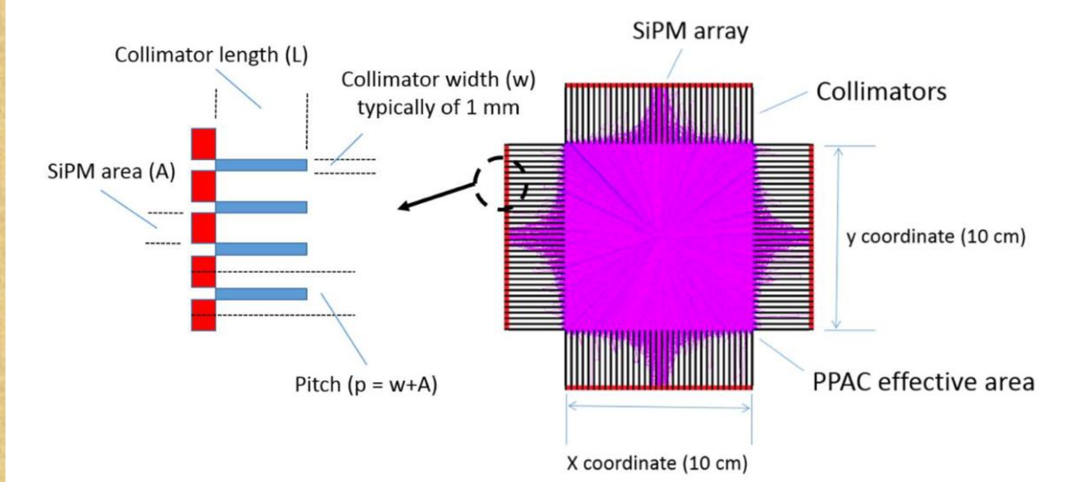


Weighted average

$$\hat{x} = \frac{\left(\frac{P_{x1} \cdot N_{x1}}{\sigma_{x1}} + \frac{P_{x2} \cdot N_{x2}}{\sigma_{x2}}\right)}{\left(\frac{N_{x1}}{\sigma_{x1}} + \frac{N_{x2}}{\sigma_{x2}}\right)}$$

$$\hat{y} = \frac{\left(\frac{P_{y1} \cdot N_{y1}}{\sigma_{y1}} + \frac{P_{y2} \cdot N_{y2}}{\sigma_{y2}}\right)}{\left(\frac{N_{y1}}{\sigma_{y1}} + \frac{N_{y2}}{\sigma_{y2}}\right)}$$

Parameters of interest



- **Collimator Length (L):**
  - **Large L** → better resolution, poor statistics
  - **Small L** → worse resolution, better statistics
- **Pressure (p):**
  - High pressure → higher photon statistics

What is the optimal combination of these parameters?

- Traditional approach: simulate the detector for many configurations and test all of them

- New approach: use differentiable programming and automatic differentiation

Image from 1808.05882

Differentiable programming for experiment design

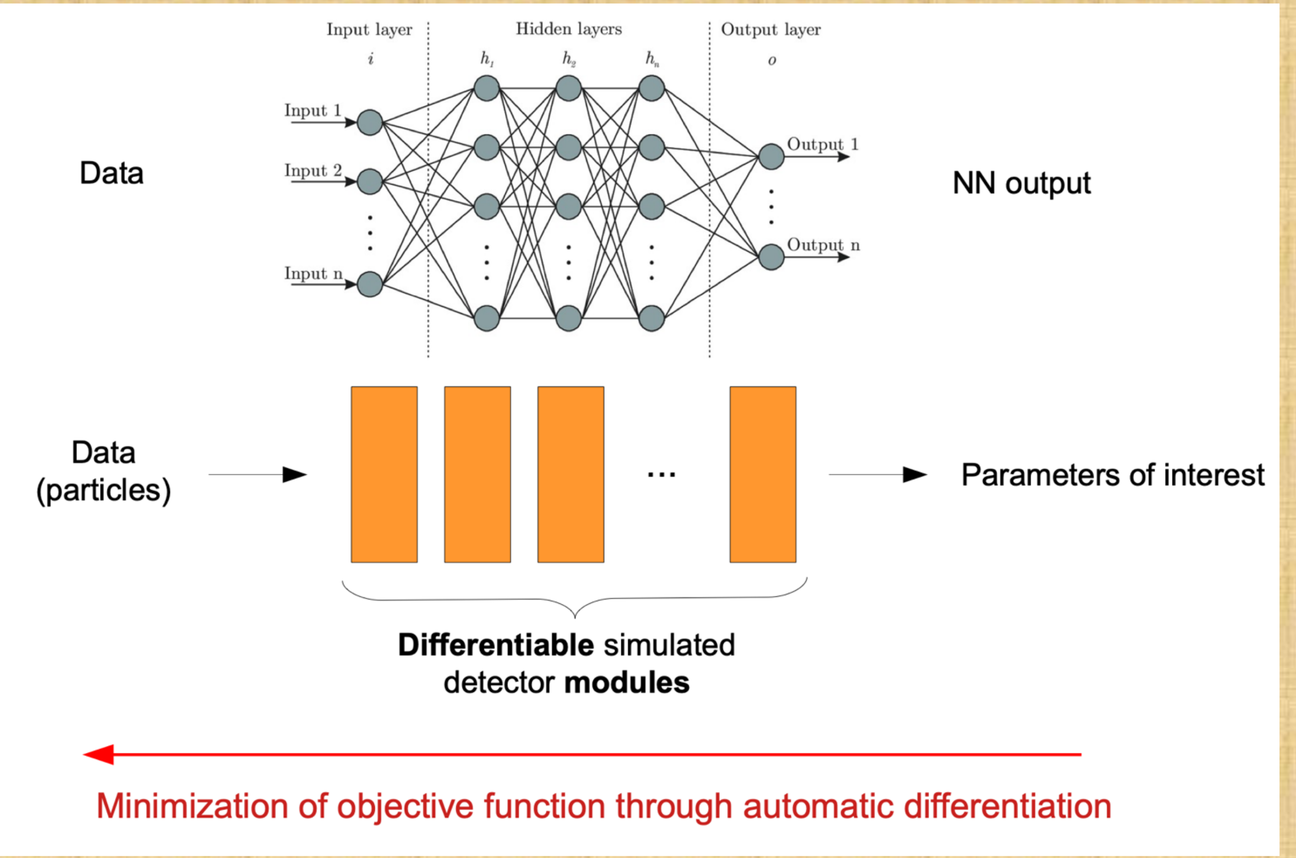
Designing experiments is a challenging task

- Number of parameters can be too high

- Correlations between parameters can be non trivial

- Traditional approaches are computationally costly

Development of deep learning techniques allows us to take a new approach



NN weights and biases → detector parameters

Image from [Julien Donini - Seminaire LPNHE - 14/02/2022]

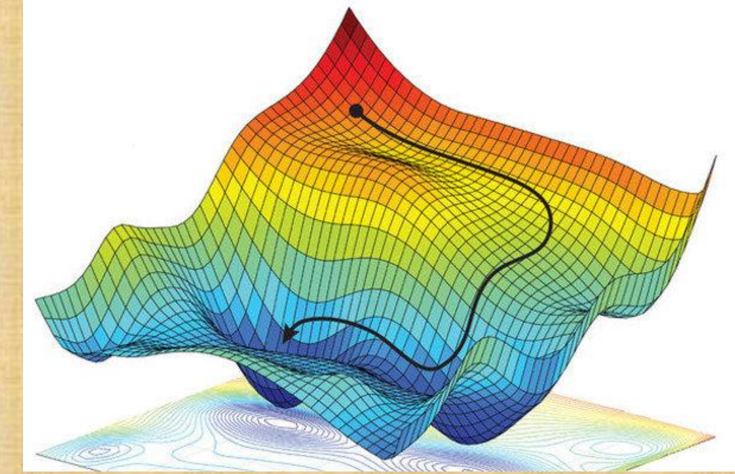
Automatic optimization of O-PPAC: Steps

1. Model detector response as a **differentiable function of the parameters**

2. Set loss function (MSE):

$$\mathcal{L}(p, L, x, y) = \frac{1}{2}\Big[\big(x - \hat{x}(p, L, x, y)\big)^2 + \big(y - \hat{y}(p, L, x, y)\big)^2\Big]$$

* From step 1

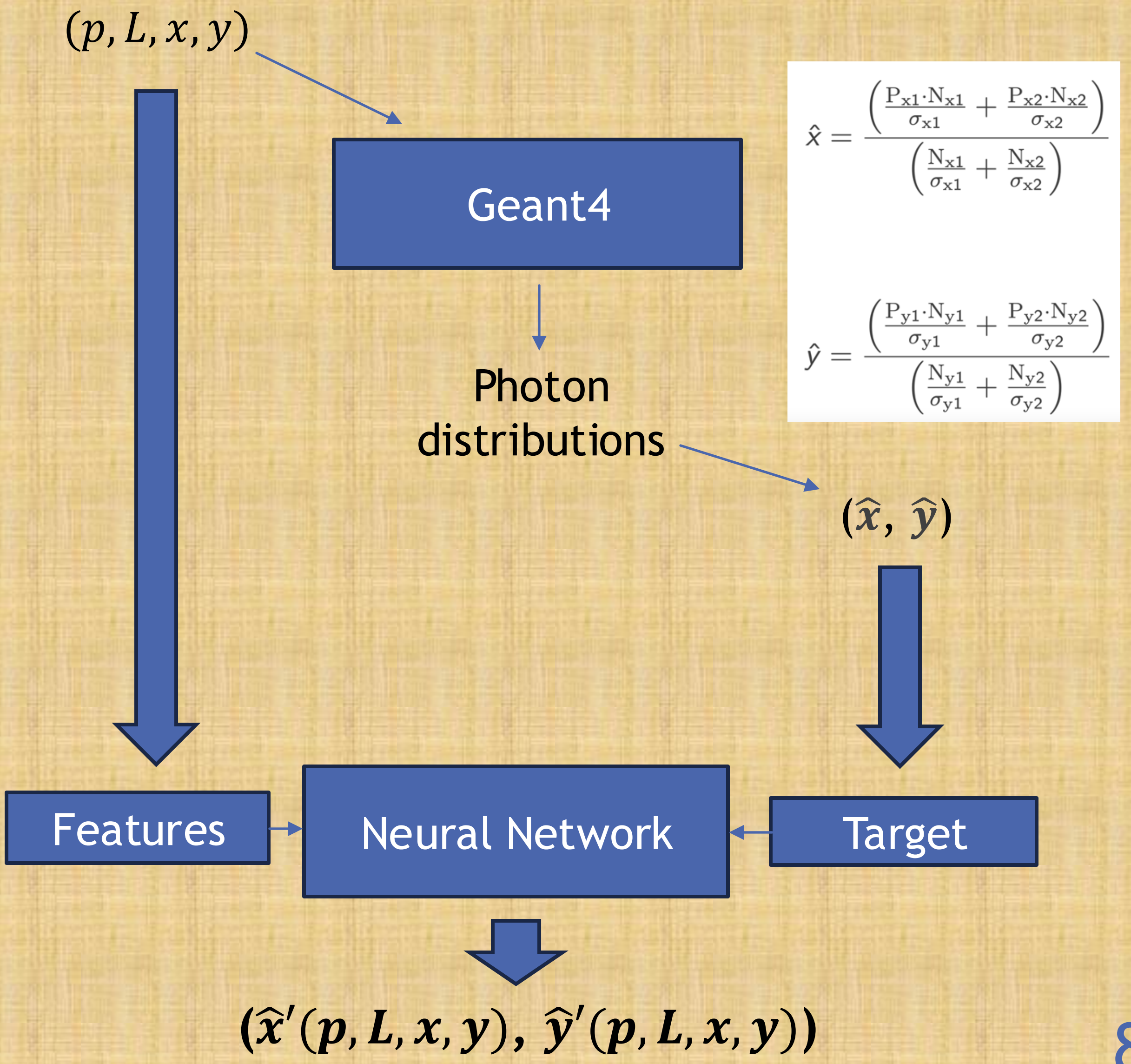


3. Minimise the loss w.r.t. p and L **using automatic differentiation**

Image from EasyAI. Gradient Descent

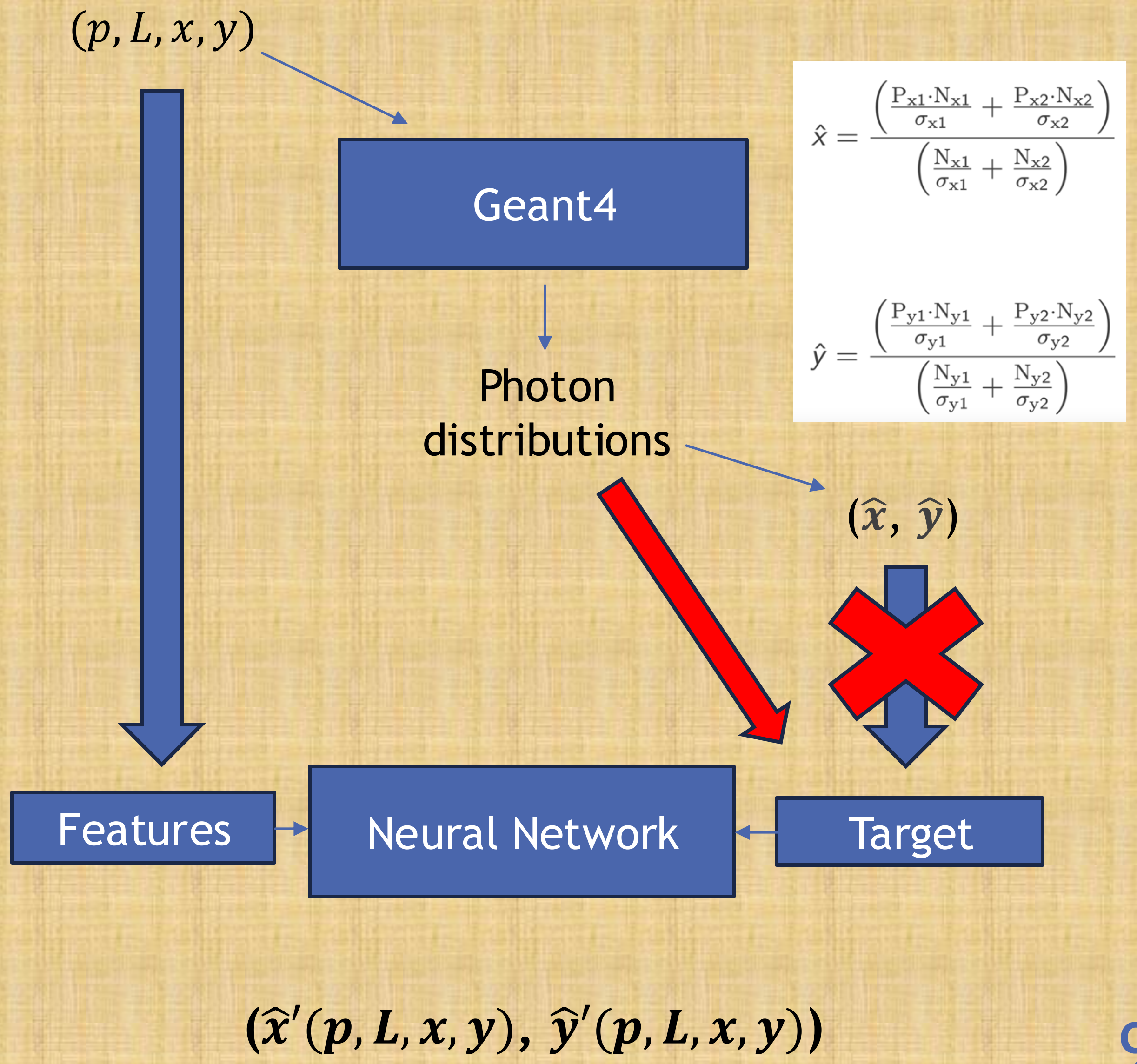
Automatic optimization of O-PPAC: 1. Surrogate model

- **Geant4 is not differentiable!**
- **We trained NN to predict the reconstructed position as a function of $(\boldsymbol{p}, \boldsymbol{L}, \boldsymbol{x}, \boldsymbol{y})$.**
- **Once trained, the NN is much faster than the simulation, inference is done in seconds while simulation takes ~hours.**



Automatic optimization of O-PPAC: Surrogate model

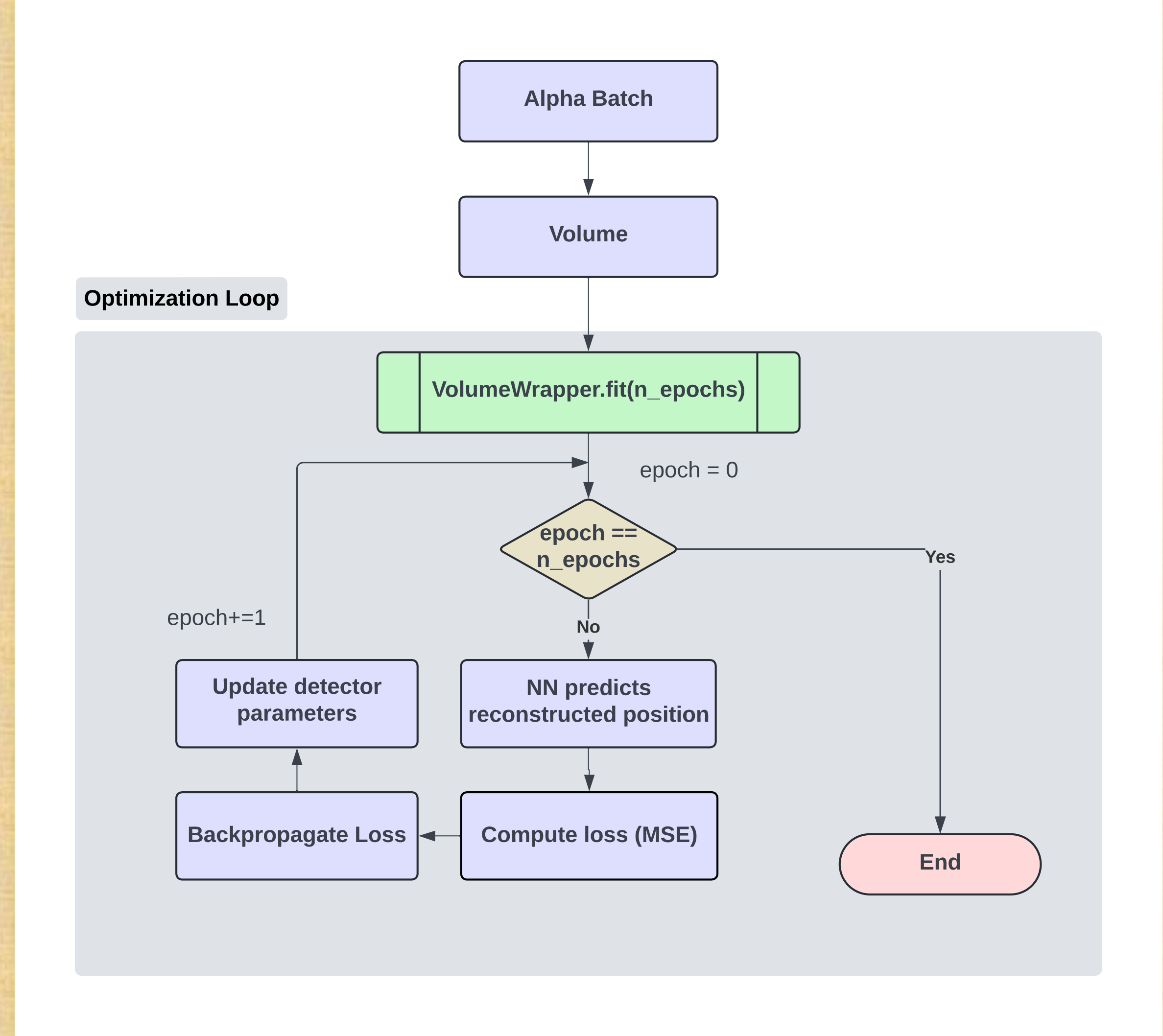
- Geant4 is not differentiable!
- We trained NN to **predict the reconstructed position as a function of (p, L, x, y).**
- Once trained, the NN is much faster than the simulation, inference is done in seconds while simulation takes ~hours.
- **Current efforts on including the reconstruction step into the differentiable pipeline**



Automatic optimization of O-PPAC: Optimization loop

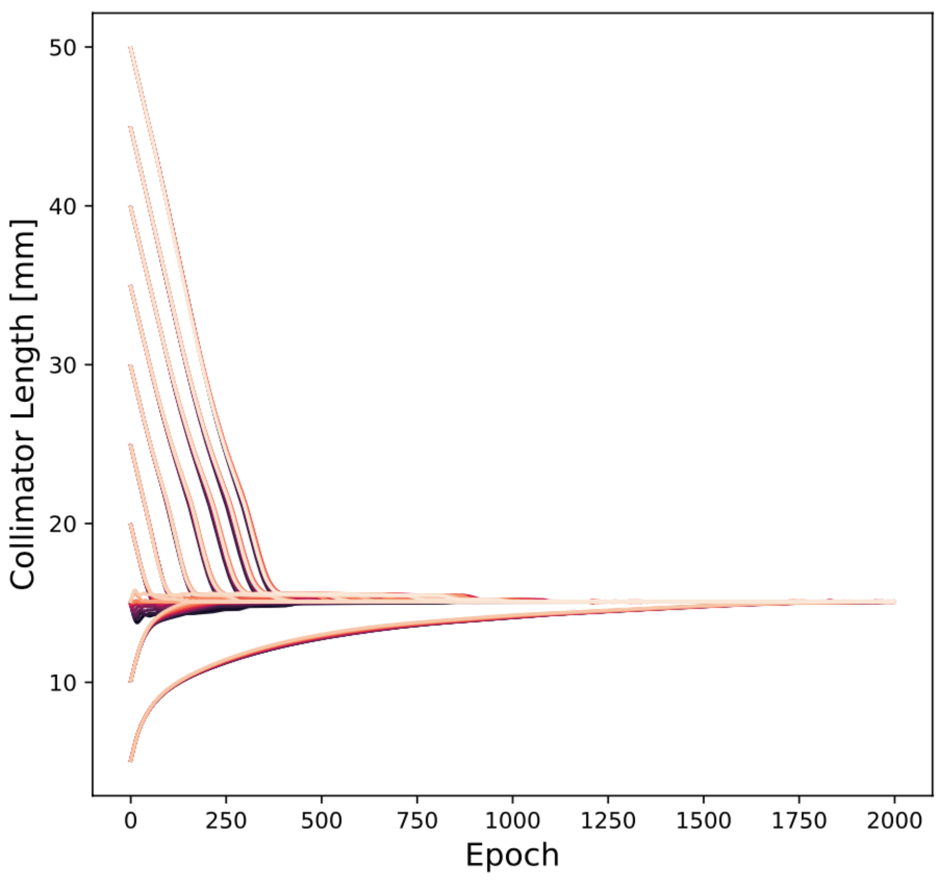
How it works?

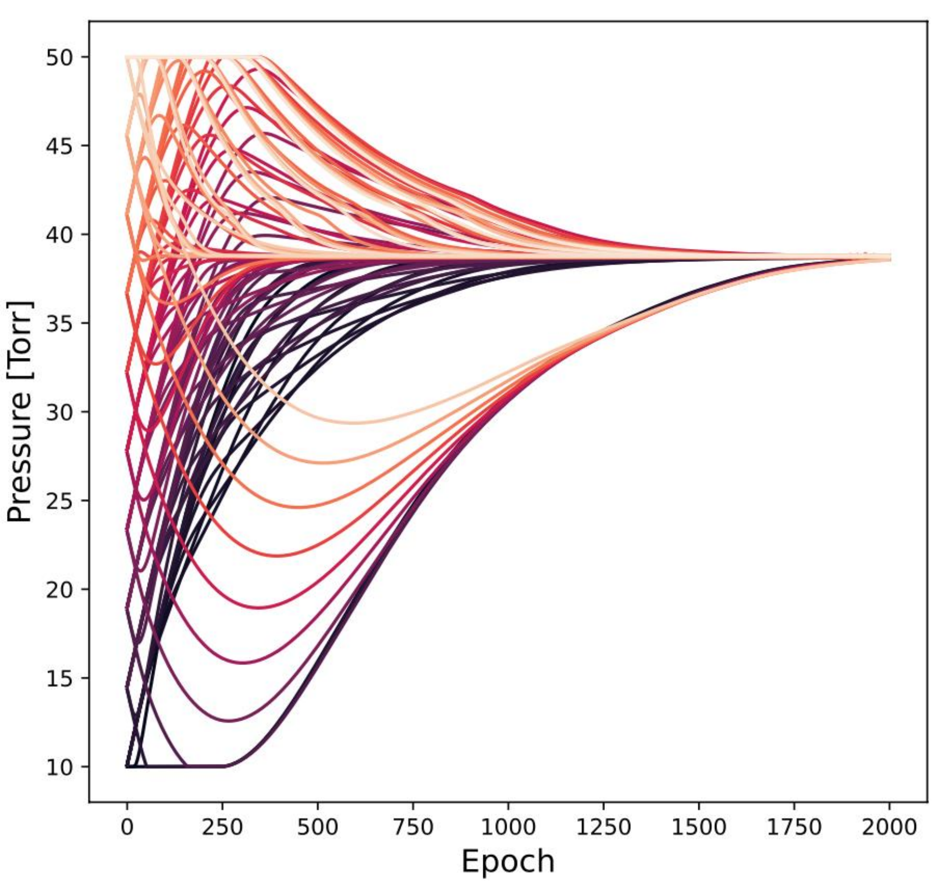
1. Generate random input for the NN.
2. NN predicts the reconstructed position.
3. Evaluate the loss, i.e. the reconstruction error.
4. Backpropagate loss
5. Update $\boldsymbol{p}, \boldsymbol{L}$ in the direction that minimizes the loss



Automatic optimization of O-PPAC: Results

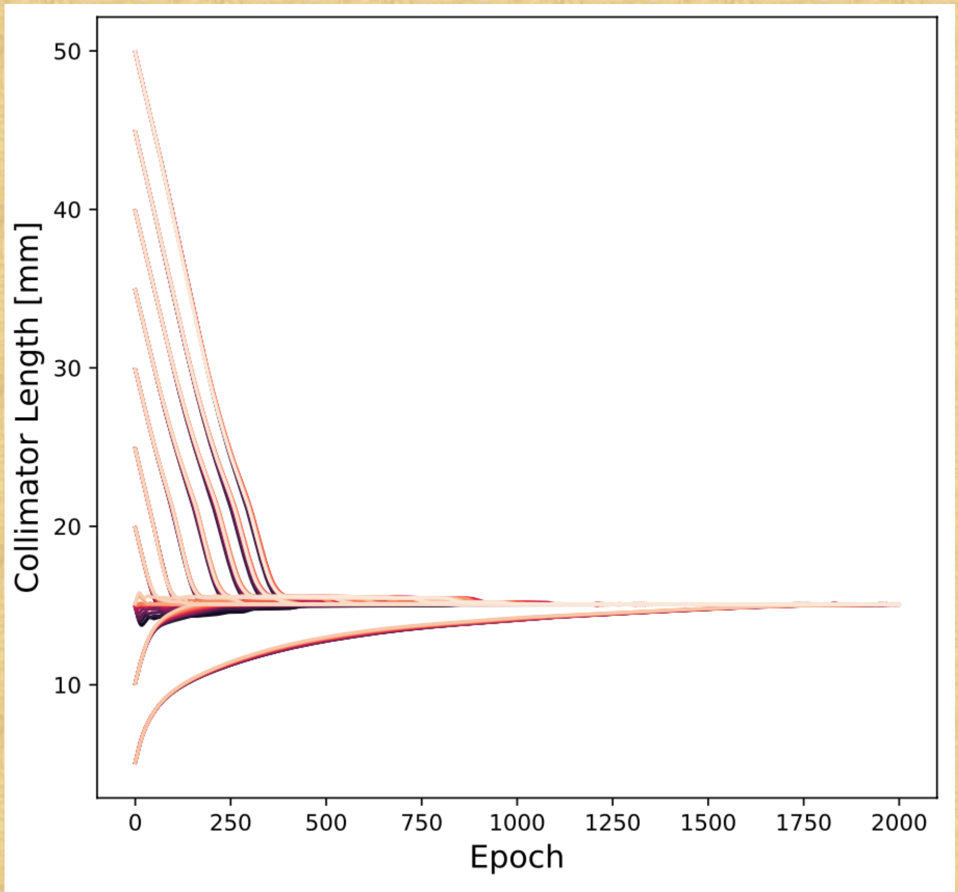
Solution remarkably stable regardless of initial configuration

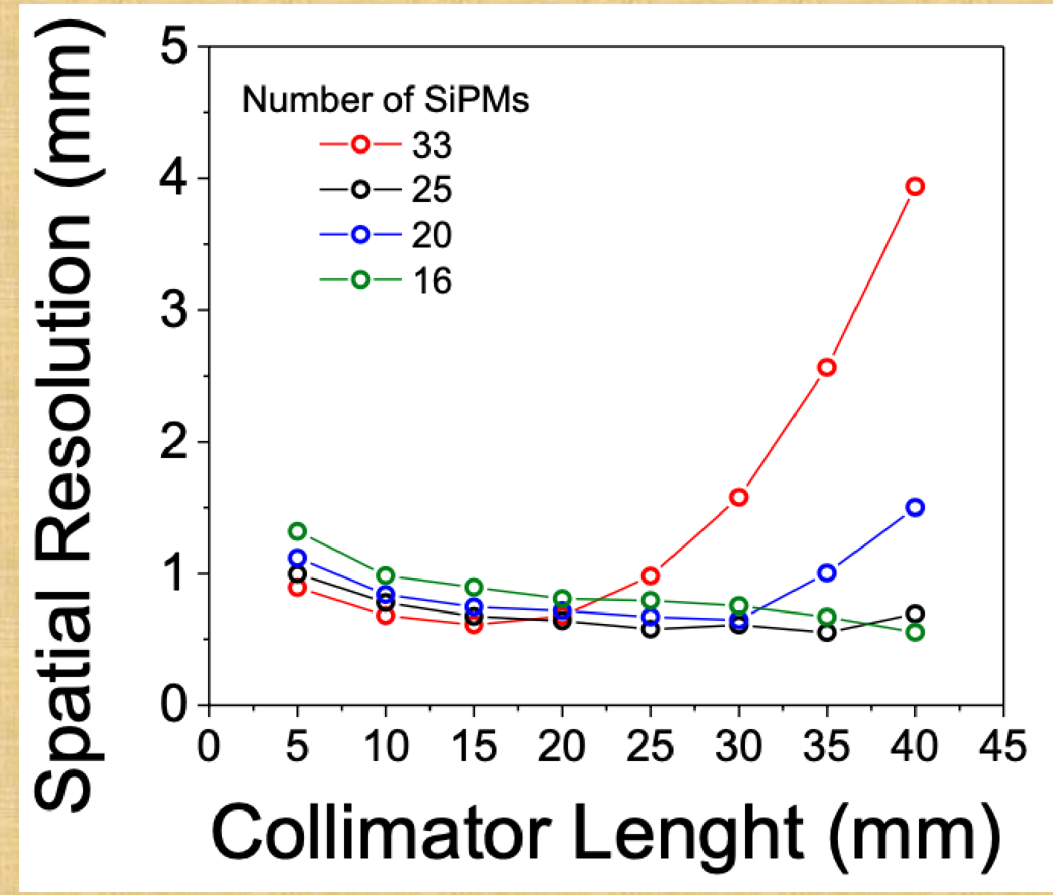




Automatic optimization of O-PPAC: Results

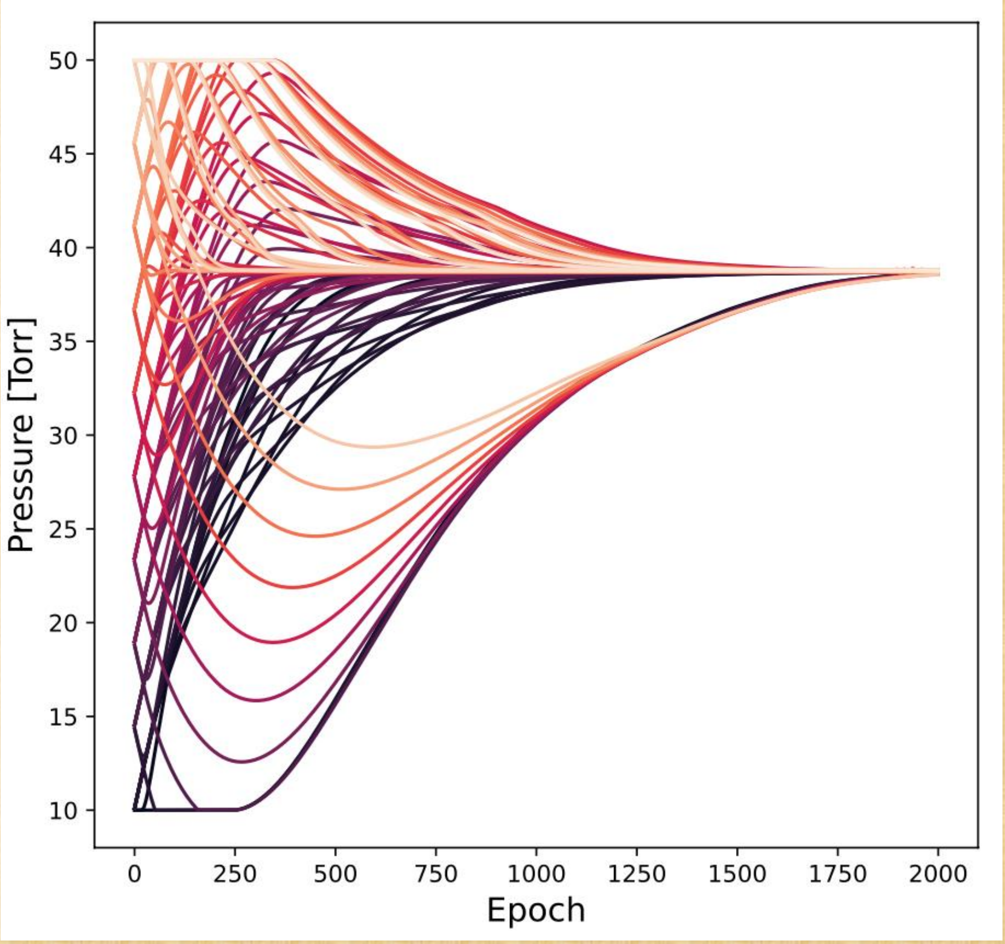
Collimator length result matches the traditional approach: 10.1088/1748-0221/13/10/P10006





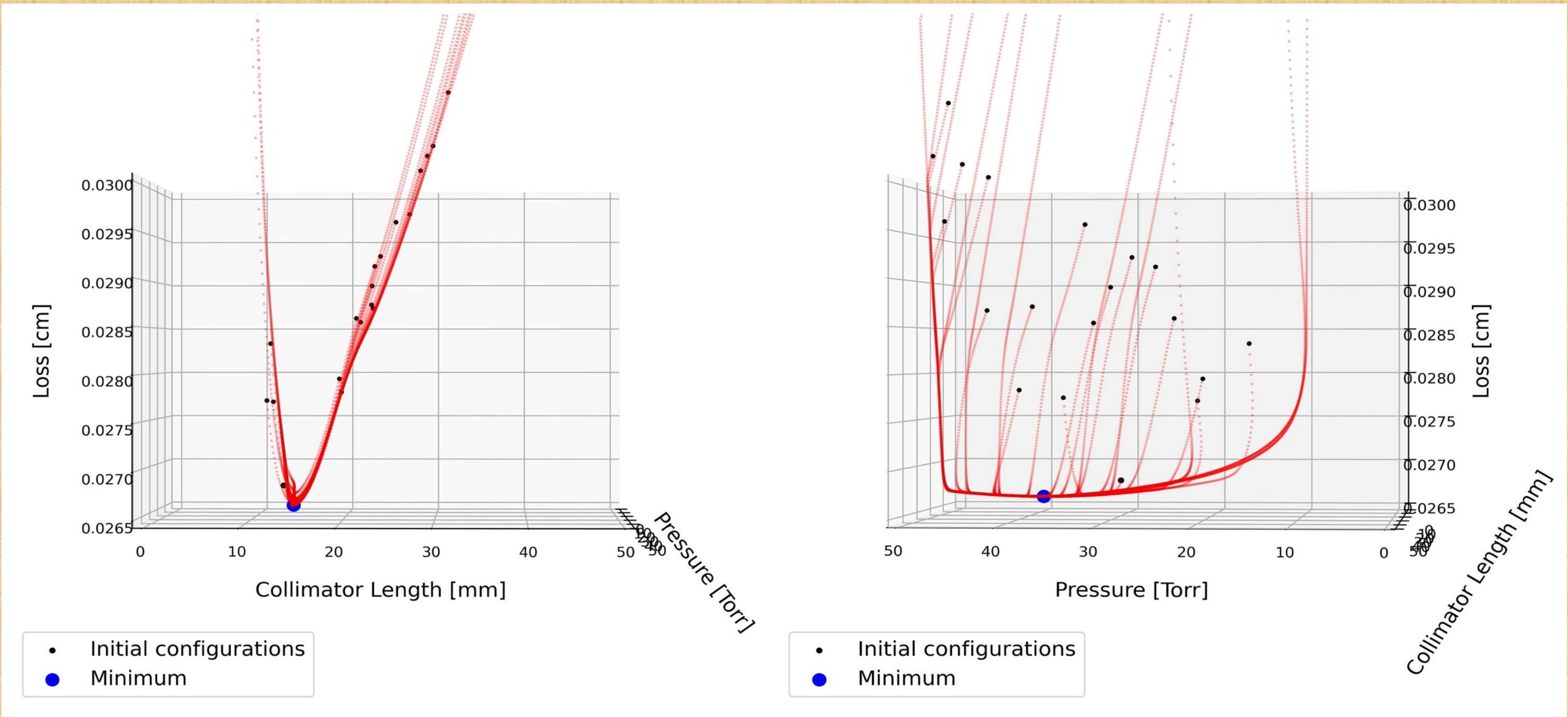
Automatic optimization of O-PPAC: Results

Pressure has a more complex behaviour



- Higher pressure means more photons
- We would expect **the highest value** (50 Torr)
- **This is not the case**:
  - Is this an effect of the simulation?
  - Is it an effect of the surrogate model?
  - Further research is needed.

Automatic optimization of O-PPAC: 3D visualization



Conclusions

- We are employing **differentiable programming and automatic differentiation** for the optimization of the O-PPAC detector.

- With this first approach:
  - Solution for optimal parameters is **stable regardless of the initial configuration.**
  - Collimator length result **aligns with traditional methods.**

Next steps

- Include reconstruction step in the differentiable pipeline

- Check pressure result

- Ultimately build the differentiable pipeline for the whole tomography system.

Thank you for your attention!

Questions?

Extra slides: Response model

- Pytorch Dense NN
- Hyperparameter tuning with **Optuna**
    - 3 layers
    - 64 neurons per layer
    - Learning rate scheduler ($\gamma = 0.9$)
    - Activation function: SELU
    - Optimizer: Adamax