Detector optimization in Muon Scattering Tomography

Maxime Lagrange on behalf of the **TomOpt authors***

Muographers2023 - International workshop on muography, June 19-22 2023 Naples

****G.Strong*[1], *T. Dorigo*[1], *P. Vischia*[2], *A. Giammanco*[3], *H. Zaraket*[4], *M. Lagrange*[3], *F. Bury*[5], *F. Nardi*[1], *A. Orio*[6], *M. Lamparth*[7], *A. Bordignon*[1], *M. Safieldin*[3], *J. Kieseler*[1]**

[1]INFN-Padova, [2]Universidad de Oviedo and ICTEA, [3]Université catholique de Louvain, [4]Lebanese University, [5]University of Bristol, [6]Muon systems, [7]Technical University of Munich

Contact: *maximelagrange98@gmail.com*

OUTLINE

I - Introduction to TomOpt concept

II - TomOpt demonstration

III - TomOpt and momentum estimation

IV - Conclusion

I - Introduction to TomOpt

Optimizing a detector for a desired task

What to **act** on?

Detector parameters x

- Position
- Efficiency
- Spatial resolution
- Span

Minimizing **cost**, **constraints** and **error on prediction**

Loss function $\mathcal{L}(x)$

- Error
- Cost & constraints

Optimization becomes a **minimization** problem

$$min\ \mathcal{L}(\ x\ )$$

"Finding the local-**minimum** of a **differentiable function**"

Iterative **gradient-descent** algorithm:

at iteration k:

$$x^{(k+1)} = x^{(k)} + \eta\ .\ \nabla_x\ \mathcal{L}(\ x^{(k)})$$

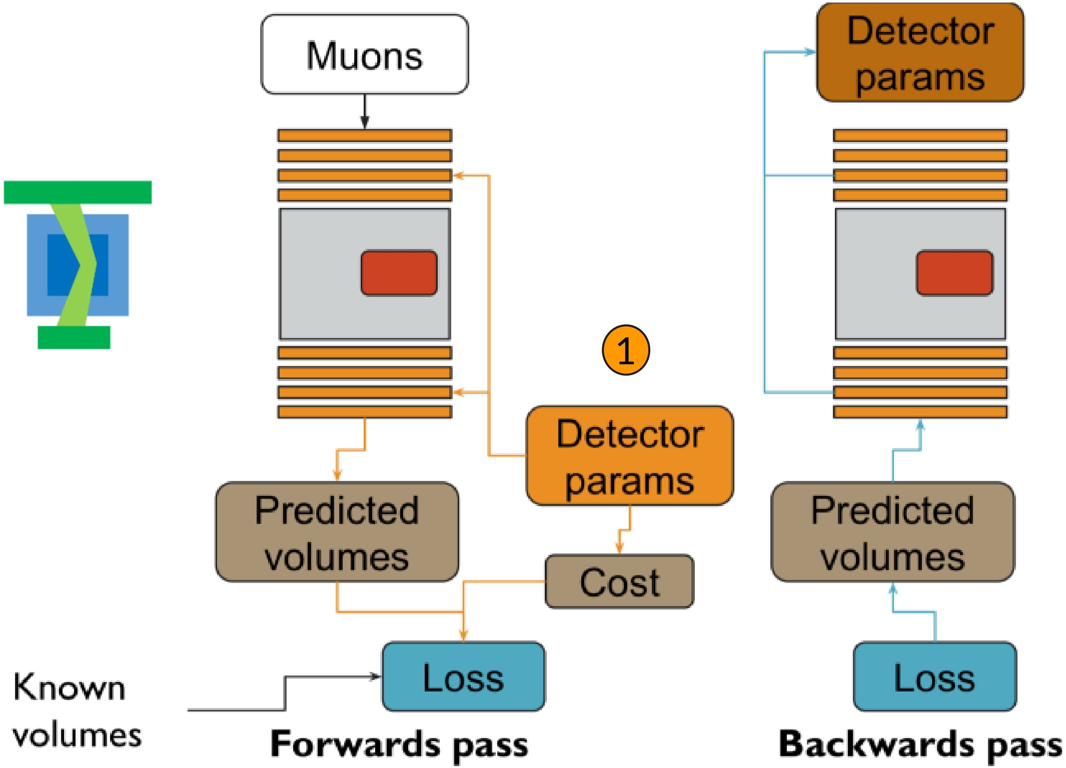
Requires a **fully differentiable simulation pipeline**

"Toward the end-to-end optimization of particle physics instruments with differentiable programming" T. Dorigo, A. Giammanco, P. Vischia (editors) MODE Collaboration

TomOpt: differentiable optimisation

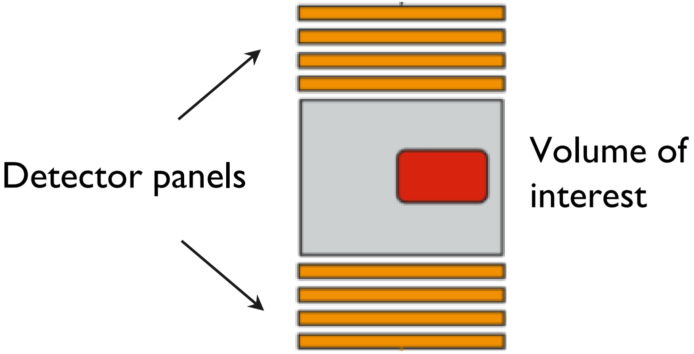
python™ PyTorch

TomOpt iteration routine



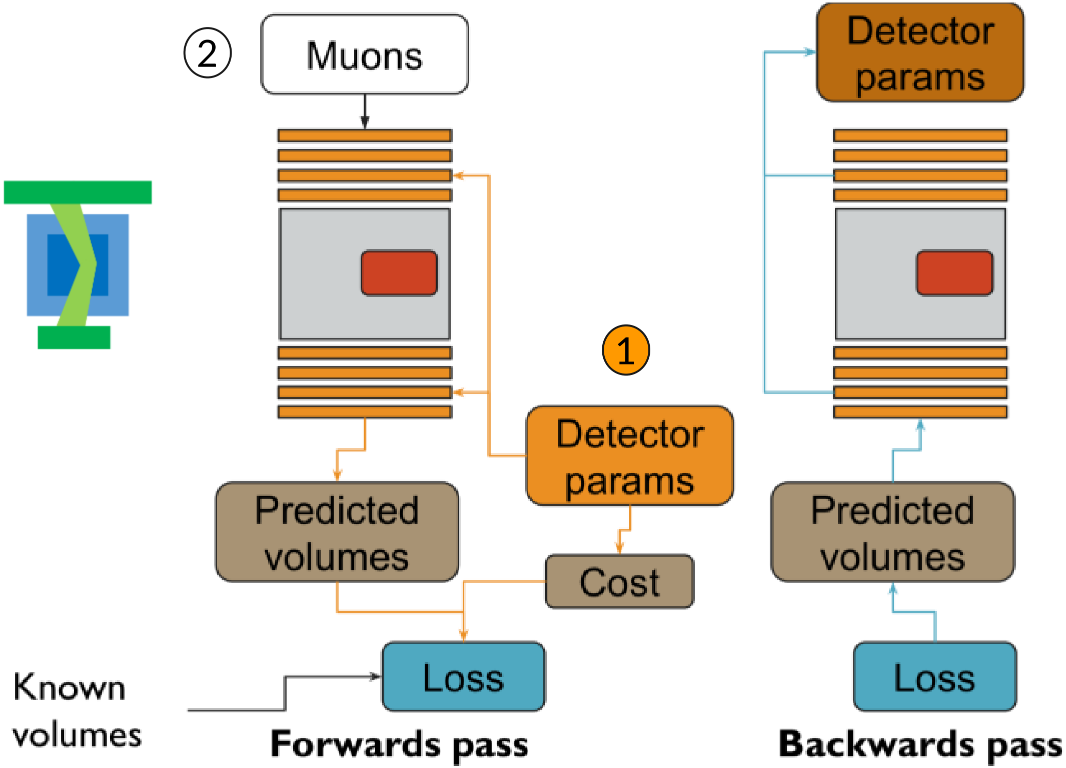
1 Initial **detector configuration**

- **Positions** x,y,z
- Spatial **resolution**
- **Efficiency**
- **Span** dx, dy

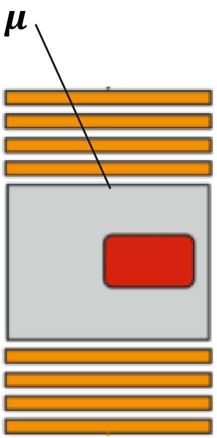


TomOpt: differentiable optimisation

TomOpt iteration routine

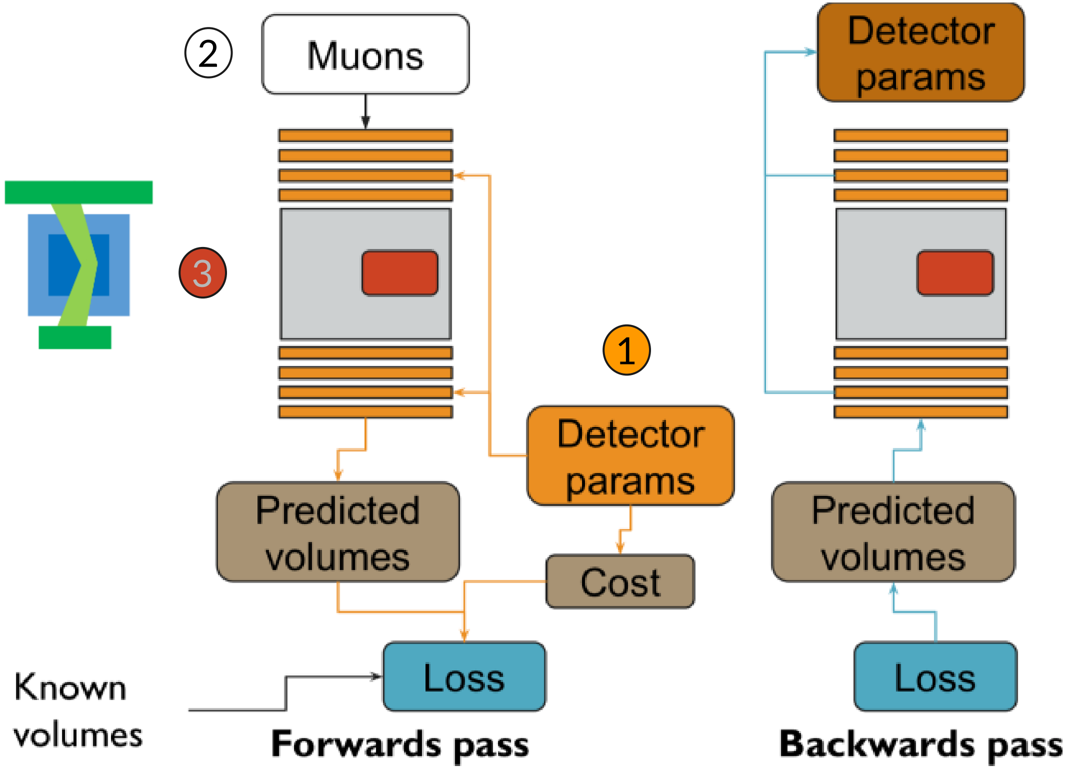


1. Initial **detector configuration**
2. **Cosmic** muon **source** sampled from literature

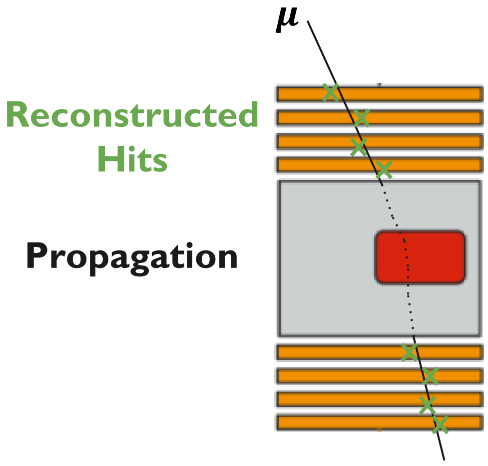


TomOpt: differentiable optimisation

TomOpt iteration routine



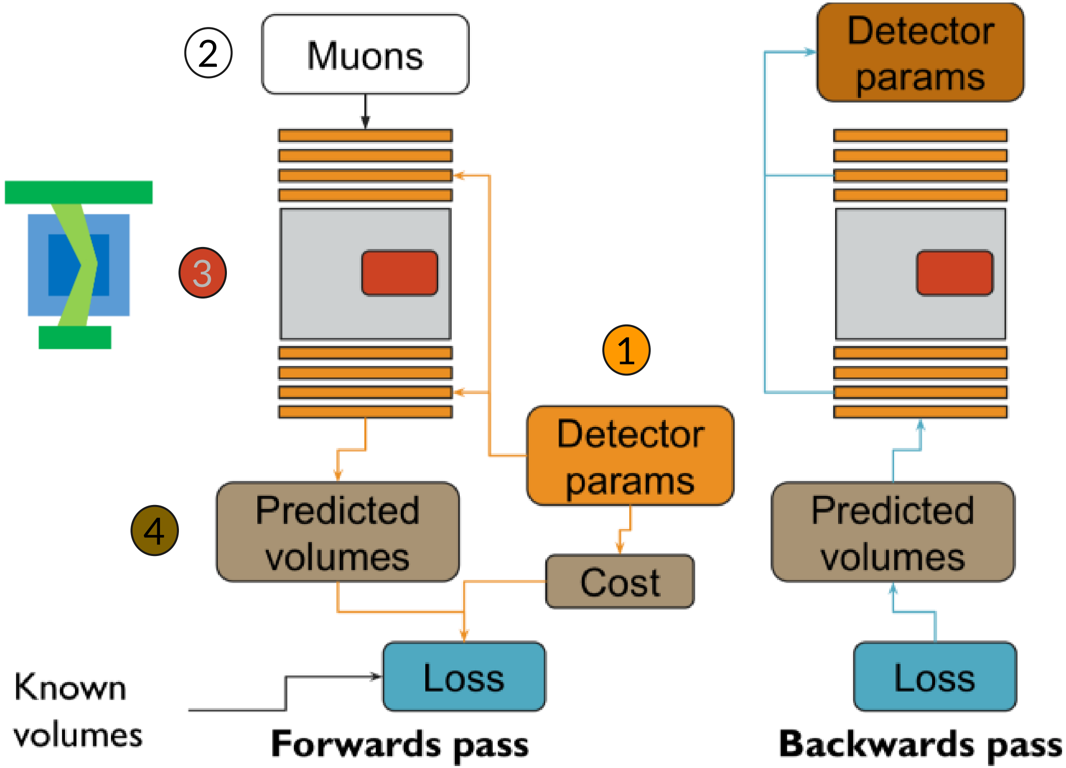
1. Initial **detector configuration**
2. **Cosmic** muon **source** sampled from literature
3. Muon **detection** and **propagation** through **matter**



Custom made **differentiable** muon **detection** and **propagation** models

TomOpt: differentiable optimisation

TomOpt iteration routine

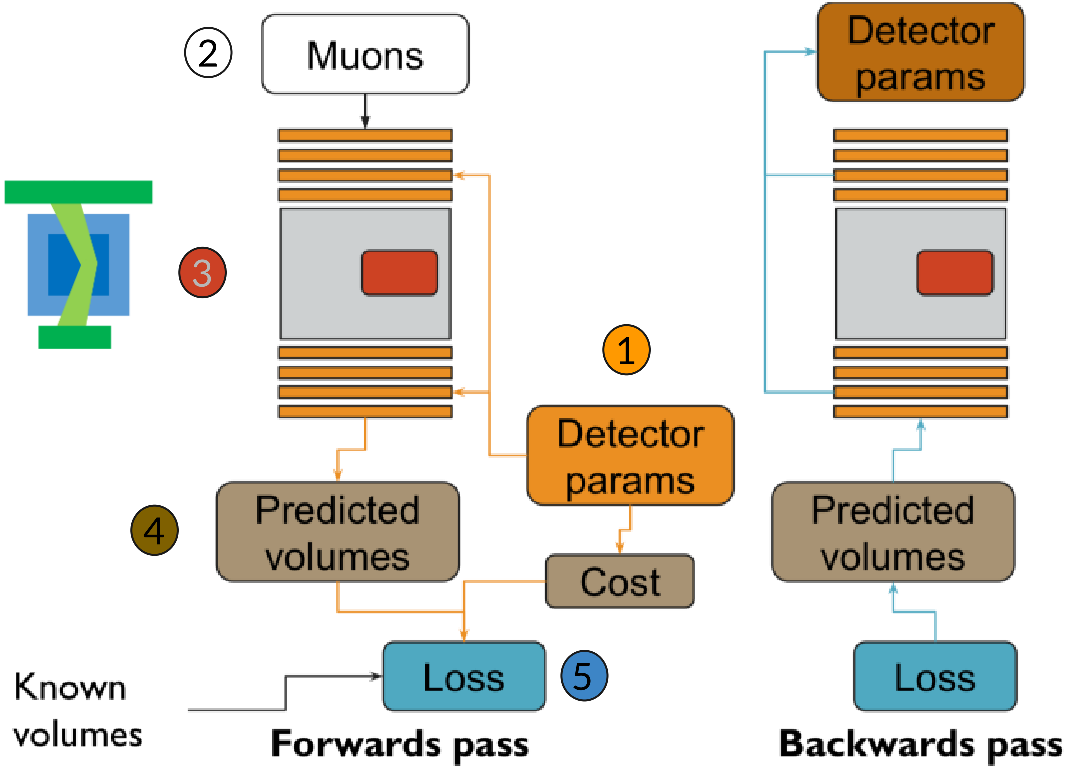


1. Initial **detector configuration**
2. **Cosmic** muon **source** sampled from literature
3. Muon **detection** and **propagation** through **matter**
4. Volume **prediction**

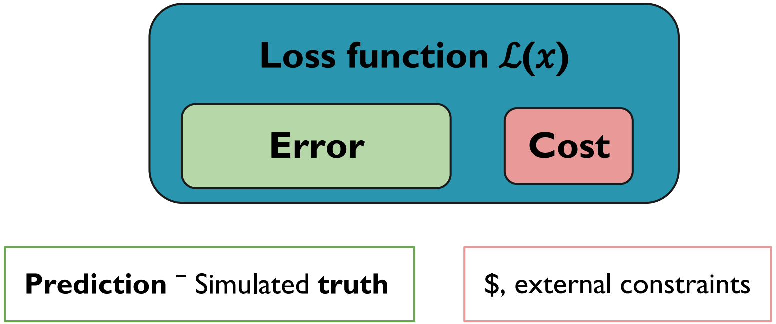
Using **reconstructed hits** and a given reconstruction **method, predict** the desired figure of merit

TomOpt: differentiable optimisation

TomOpt iteration routine



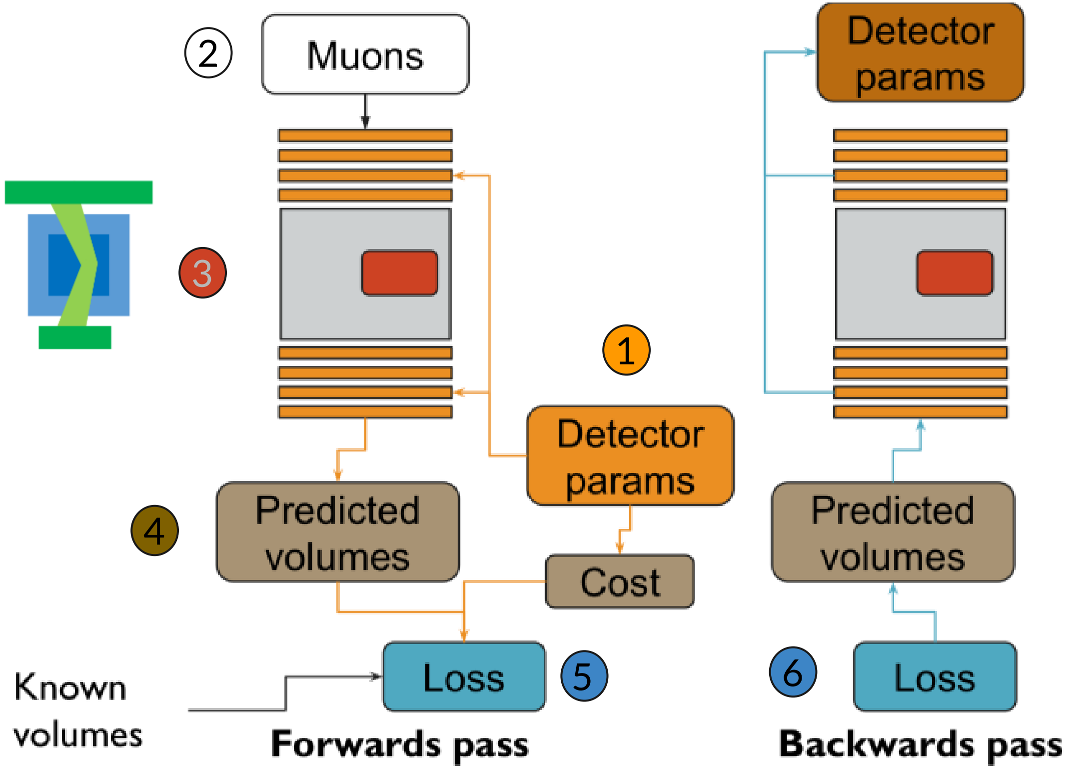
1. Initial **detector configuration**
2. **Cosmic** muon **source** sampled from literature
3. Muon **detection** and **propagation** through **matter**
4. Volume **prediction**
5. **Loss function** computation



TomOpt: differentiable optimisation

python™ PyTorch

TomOpt iteration routine



1. Initial **detector configuration**
2. **Cosmic** muon **source** sampled from literature
3. Muon **detection** and **propagation** through **matter**
4. Volume **prediction**
5. **Loss function** computation
6. Gradient-descent **optimisation**

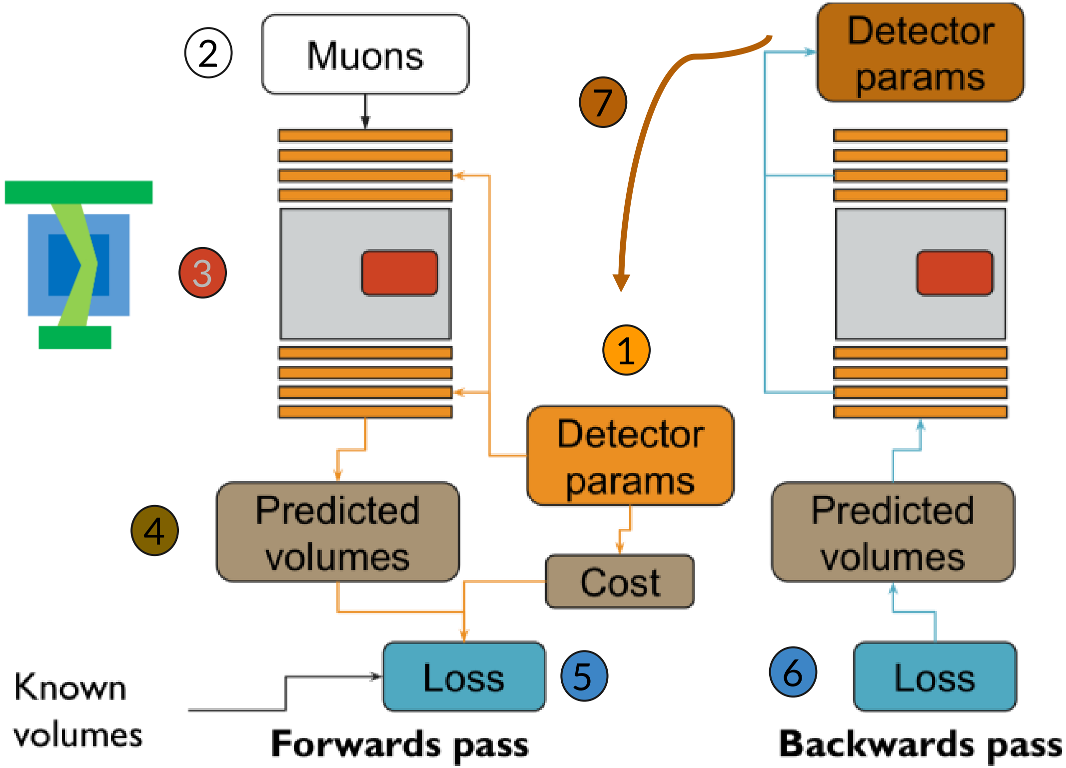
$$x^{(k+1)} = x^{(k)} + \eta \, . \, \nabla_x \mathcal{L}(x^{(k)})$$

Updated parameter — **Initial** parameter — **Loss** gradient

TomOpt: differentiable optimisation

python™ PyTorch

TomOpt iteration routine



1 Initial **detector configuration**

2 **Cosmic** muon **source** sampled from literature

3 Muon **detection** and **propagation** through **matter**

4 Volume **prediction**

5 **Loss function** computation

6 Gradient-descent **optimisation**

7 Modified geometry used at step 1

Repeat until **Loss function minimum** is found

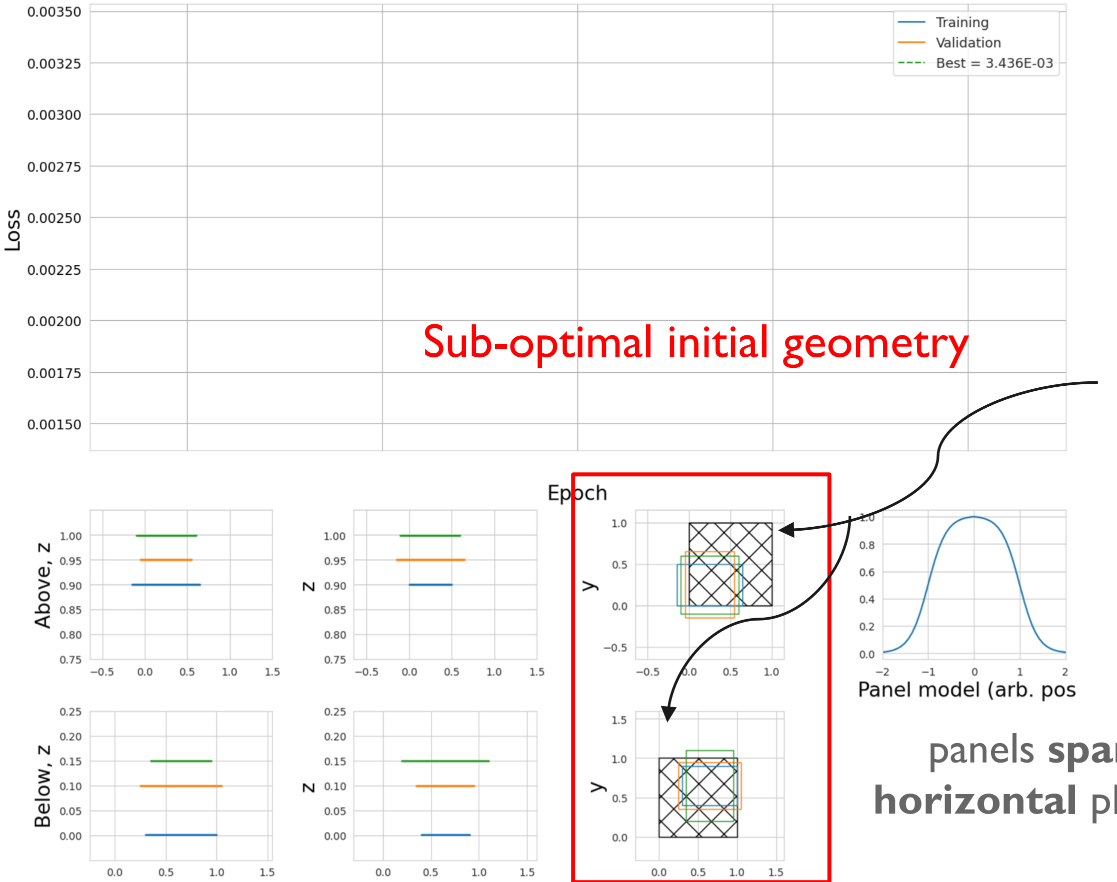
II - TomOpt demonstration

TomOpt: X_0 inference performance driven optimisation

Loss function
$\mathcal{L}$

30 free parameters:

- x,y,z position
- xy span



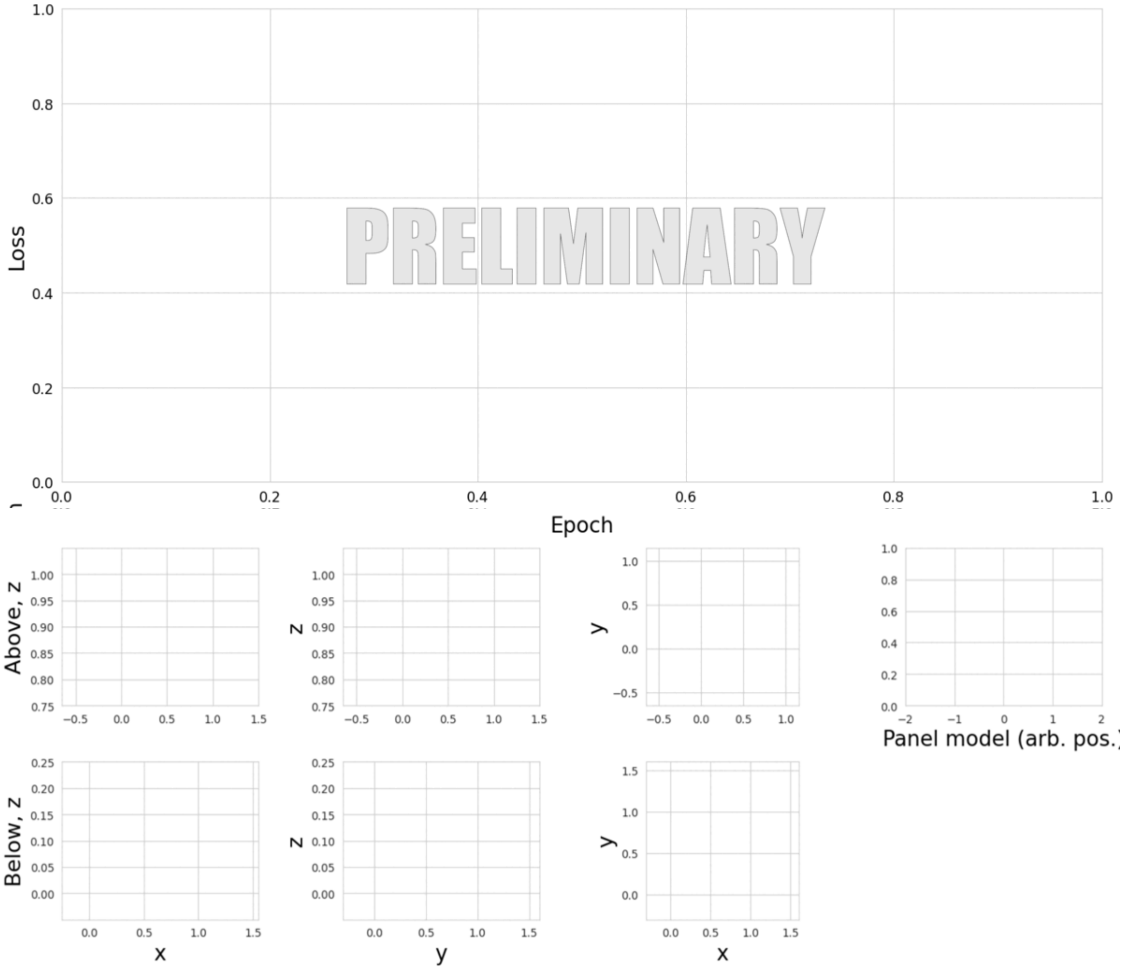
Volume of interest

Upper panels
vertical **placement**

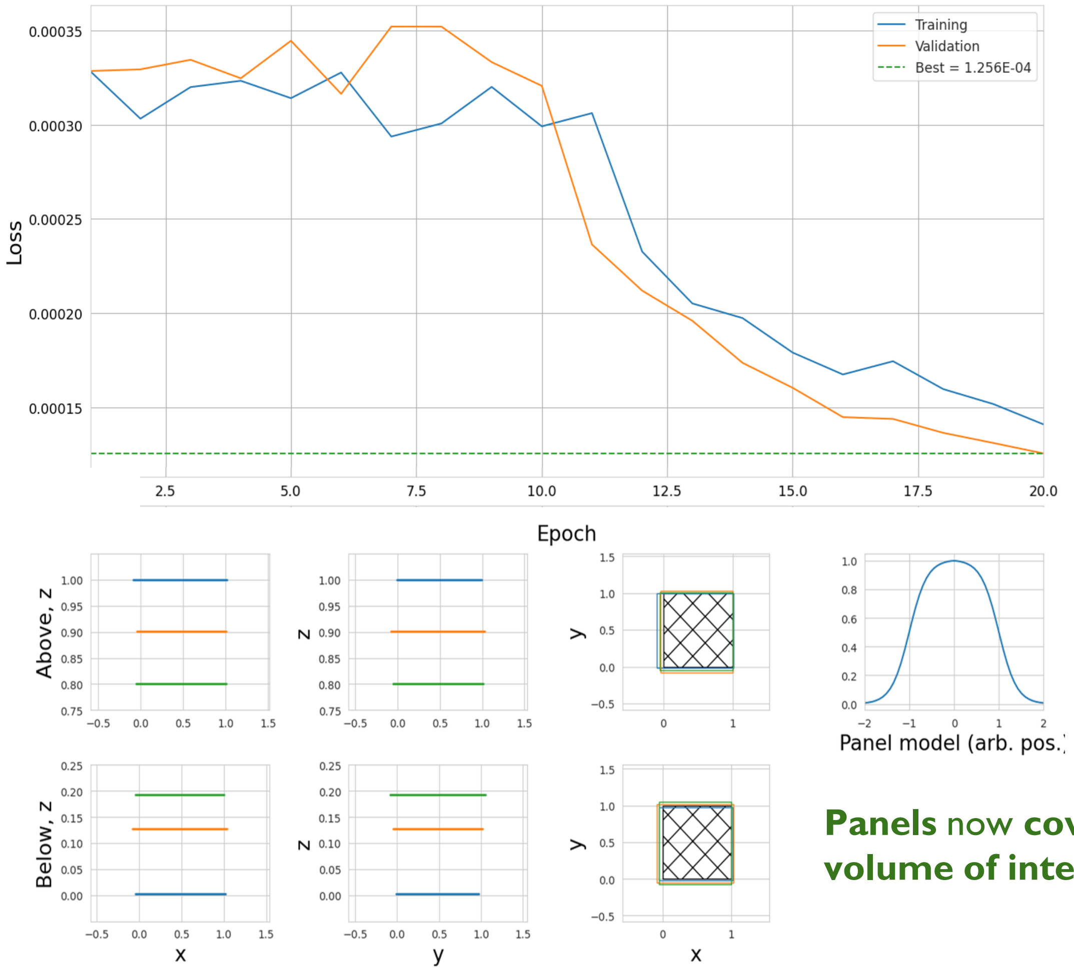
Lower panels
vertical **placement**

panels **span** and
horizontal placement

TomOpt: X_0 inference performance driven optimisation



TomOpt: X_0 inference performance driven optimisation



z placement grants **optimal angular resolution**

Acceptance is not ideal

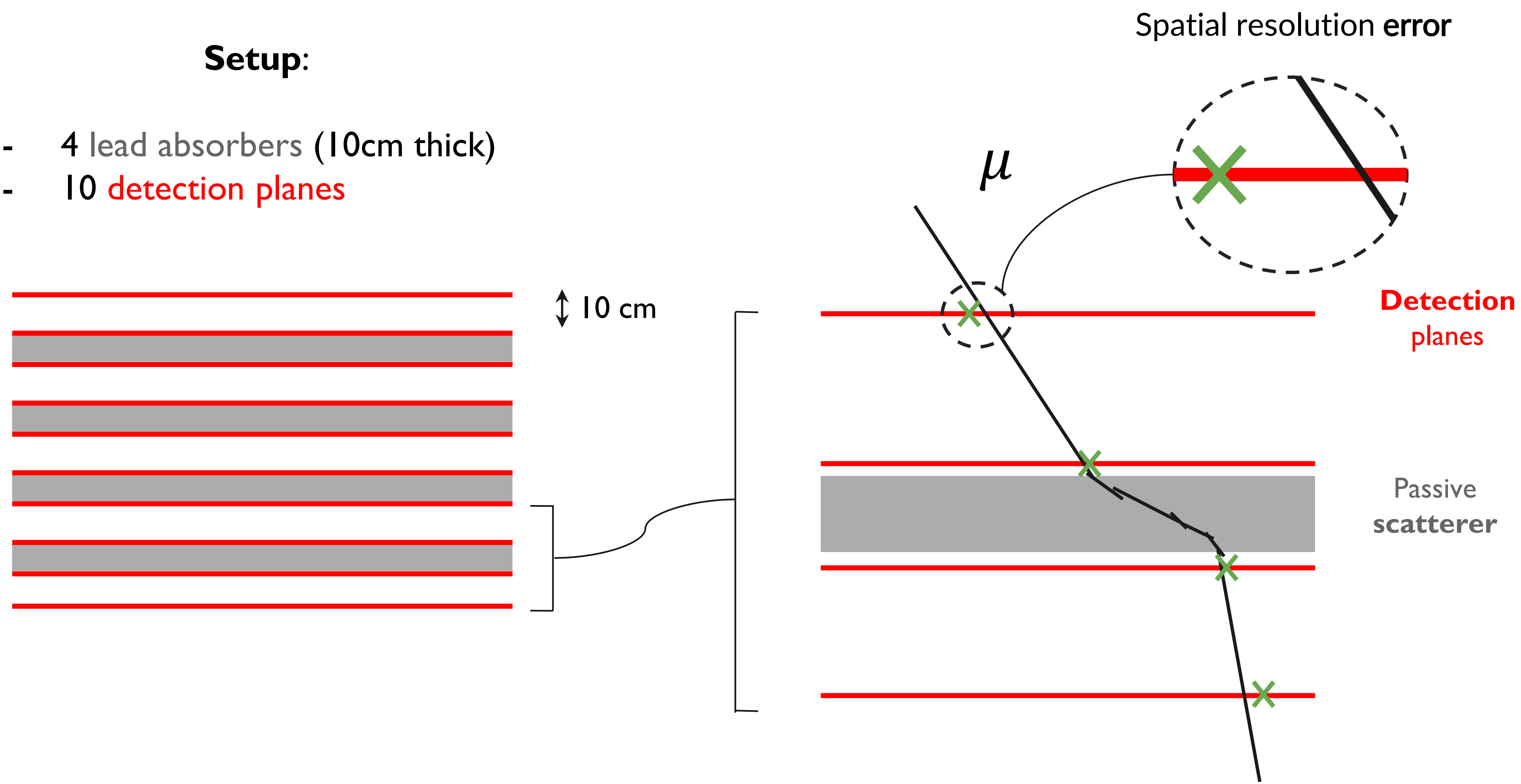
Panels now **cover** the whole **volume of interest**

III - Momentum measurement module

Momentum measurement module

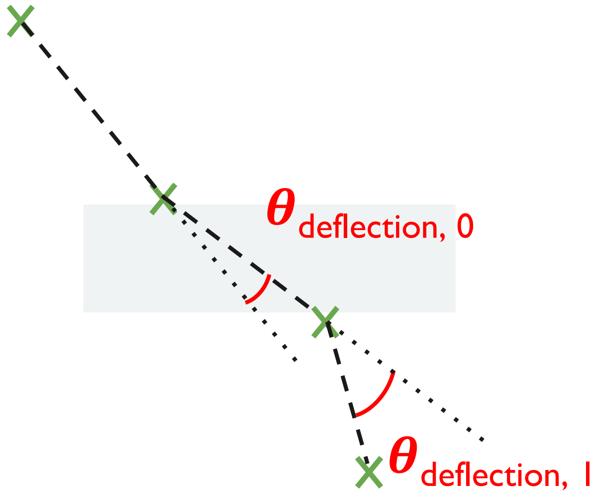
Setup:

- 4 lead absorbers (10cm thick)
- 10 detection planes

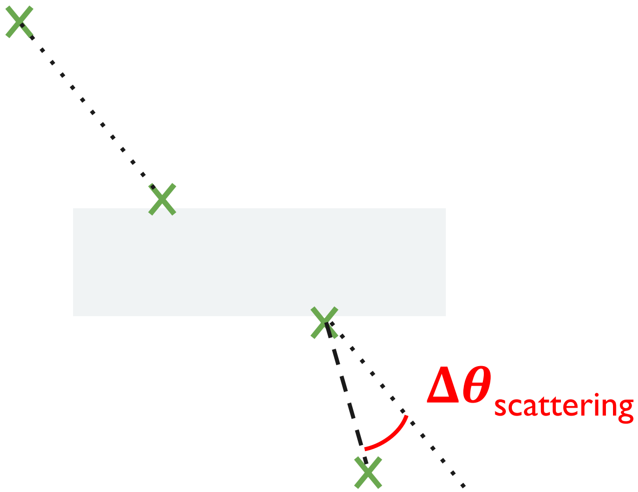


Momentum measurement module: input variables

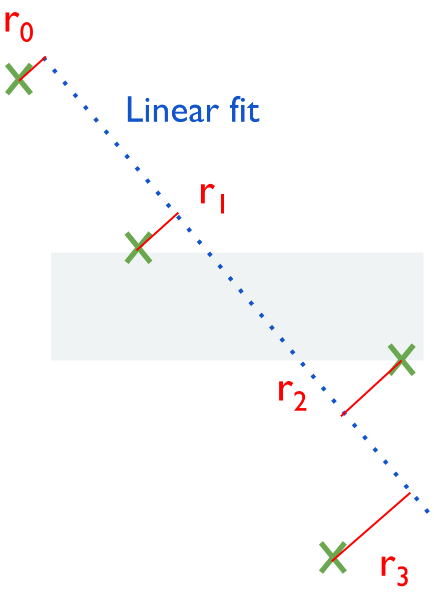
Deflection angles



Scattering angle



Fit residuals r



Momentum inference: Scattering formula regression

Input variable

4 scattering angle measurement

$\Delta\theta_{RMS}$ = ¼ Σ $\Delta\theta_i$

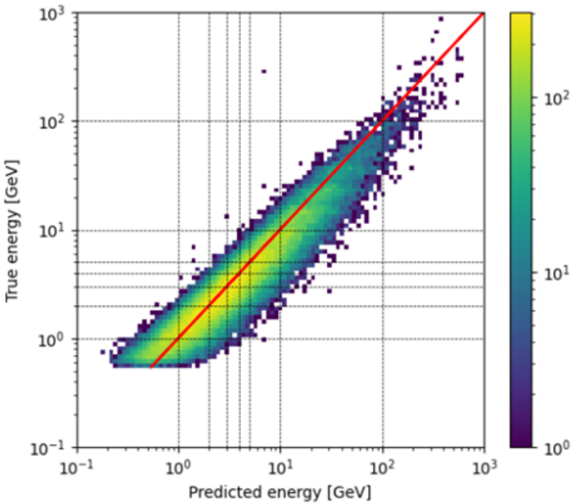
Perfect spatial resolution

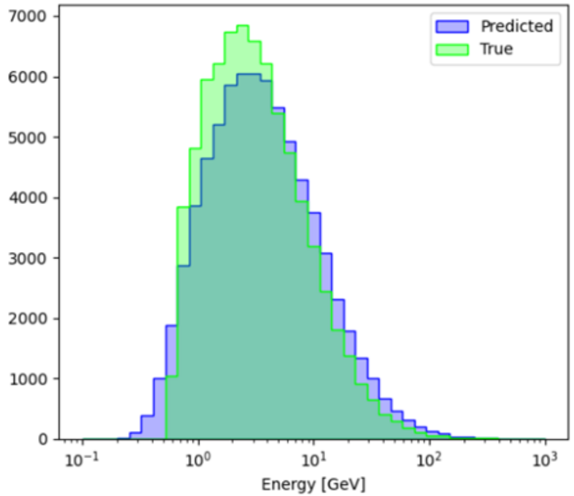
Inference method

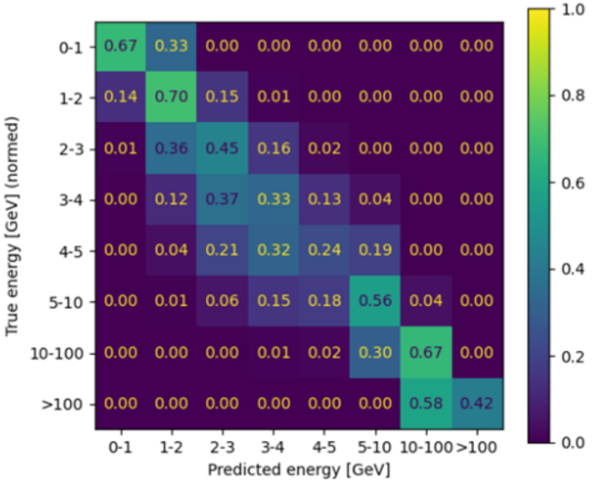
Multiple Coulomb scattering model

$$p = \frac{13.6MeV}{\theta_{RMS}}\sqrt{\frac{x}{X_0}}$$

Prediction summary (Formula)







True \ Predicted	0-1	1-2	2-3	3-4	4-5	5-10	10-100	>100
0-1	0.67	0.33	0.00	0.00	0.00	0.00	0.00	0.00
1-2	0.14	0.70	0.15	0.01	0.00	0.00	0.00	0.00
2-3	0.01	0.36	0.45	0.16	0.02	0.00	0.00	0.00
3-4	0.00	0.12	0.37	0.33	0.13	0.04	0.00	0.00
4-5	0.00	0.04	0.21	0.32	0.24	0.19	0.00	0.00
5-10	0.00	0.01	0.06	0.15	0.18	0.56	0.04	0.00
10-100	0.00	0.00	0.00	0.01	0.02	0.30	0.67	0.00
>100	0.00	0.00	0.00	0.00	0.00	0.00	0.58	0.42

Momentum inference: DNN regression

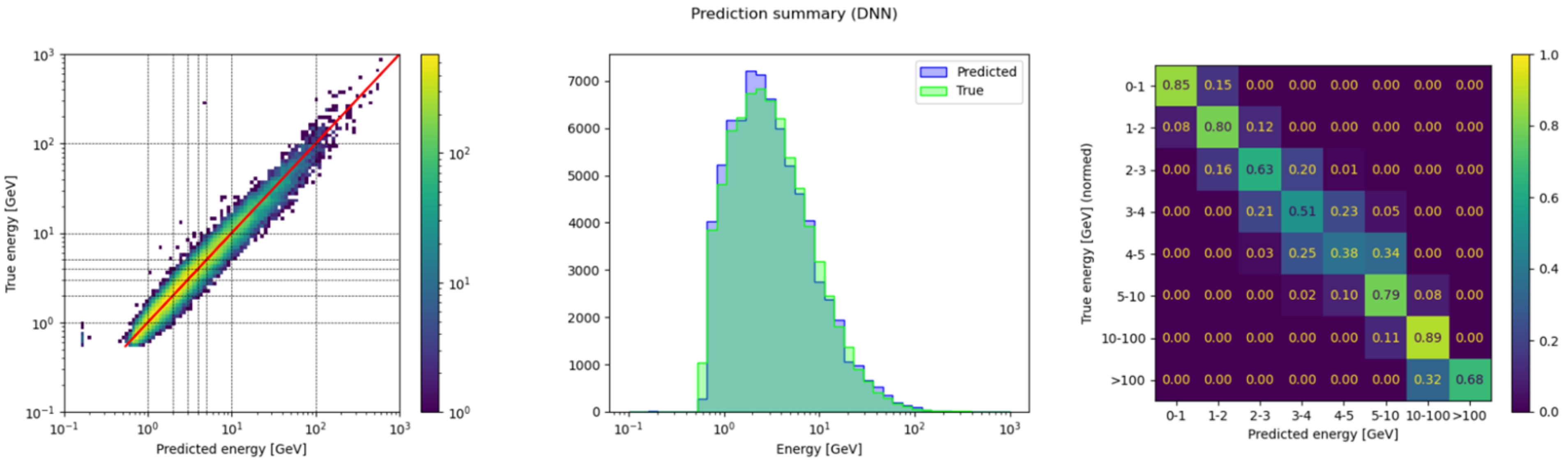
Input variable

- **8 deflection** angles
- 10 **residuals**
- 4 **scattering** angles

Perfect spatial resolution

Inference method

- **DNN**
- Three 64 neurons layers
- Infer on $\log(p)$



Momentum inference: DNN regression

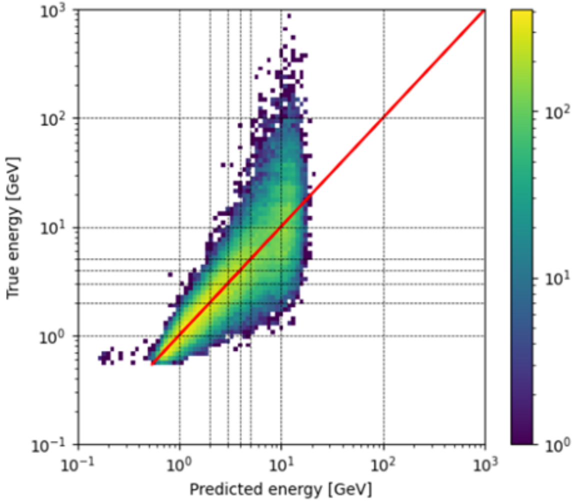
Input variable

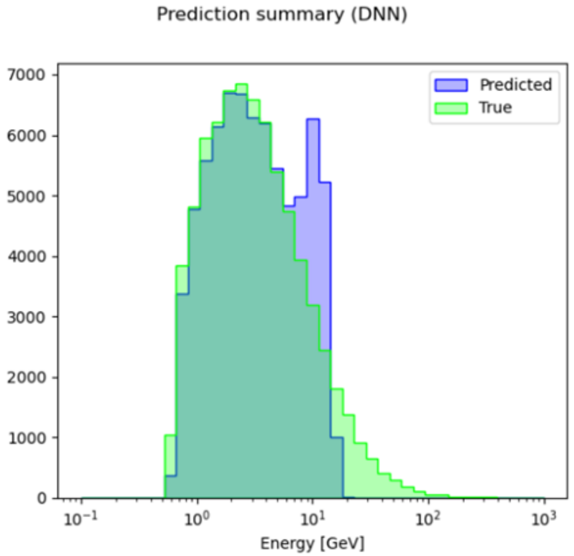
- **8 deflection** angles
- 10 **residuals**
- 4 **scattering** angles

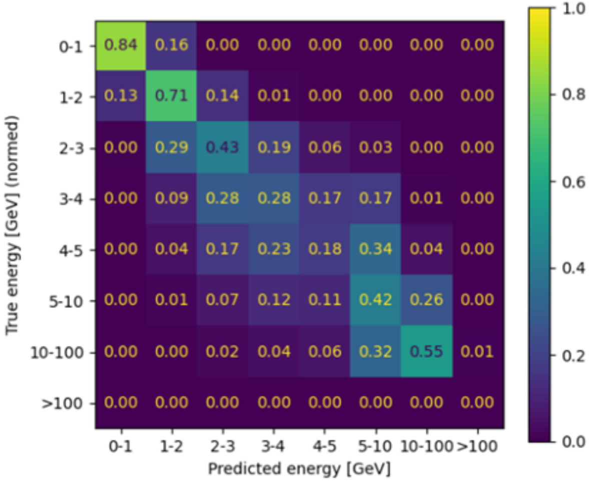
2mm spatial resolution

Inference method

- **DNN**
- Three 64 neurons layers
- Infer on $\log(p)$

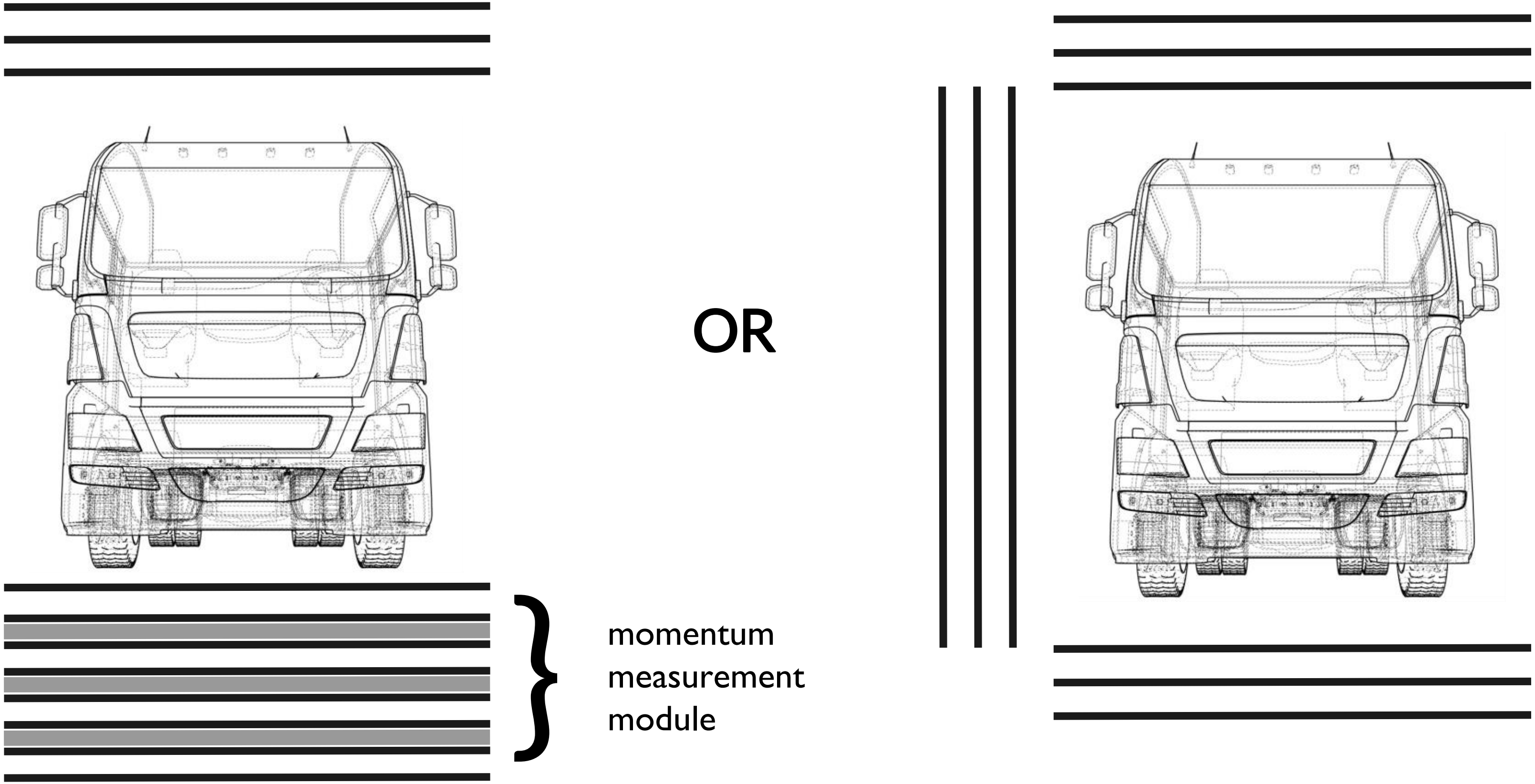






Question TomOpt will answer

Is it worth it to add a muon energy spectrometer to the detector?



IV - Conclusion

CONCLUSION

- **TomOpt optimises** detector configuration for **specific** Muon Scattering Tomography **tasks**
- **TomOpt** offers a **fully differentiable** MST **simulation pipeline**
- As long as differentiability is preserved, it can be **adapted** to **any** MST **experiment**

In the future

- **Open source release** of the software
- **Publication incoming!**
- **Include momentum measurement sub-detector** in the optimisation chain
- Don't hesitate to contact us!

Check out related **MODE** collaboration work:

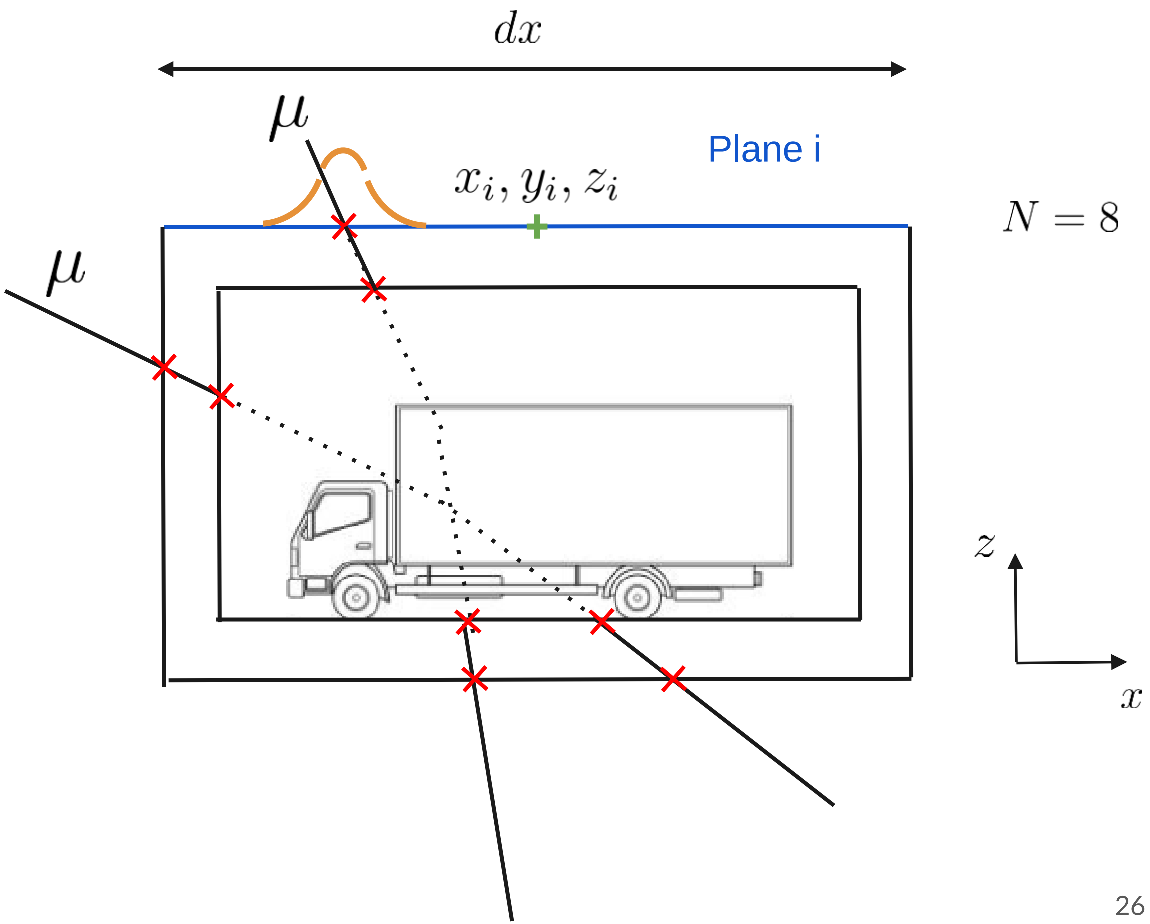
"Toward the end-to-end optimization of particle physics instruments with differentiable programming" T. Dorigo, A. Giammanco, P. Vischia (editors)

Contact: *maximelagrange98@gmail.com*

BACKUP SLIDES

Hardware parameters

Number of detection planes	N
Placement	x_i, y_i, z_i
Dimension	dx, dy
Spatial resolution	$\sigma_{x,y}$
Efficiency	ϵ
Technology	RPC's Scintillators MicroMegas

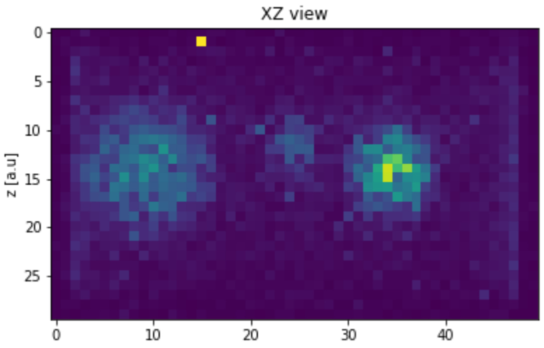


Software parameters

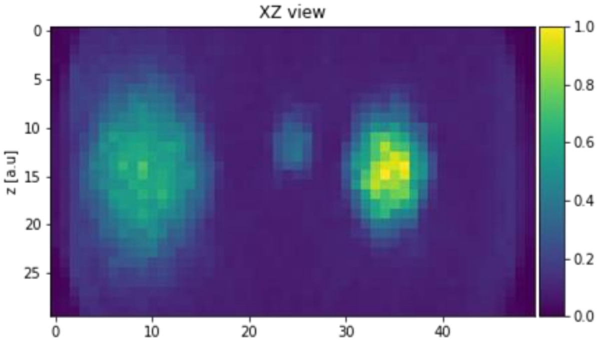
I - Reconstruction algorithm (POCA, ASR, Maximum Likelihood, Binned Clustered Algorithm, etc..)

INPUT

a - Point Of Closest Approach (POCA)



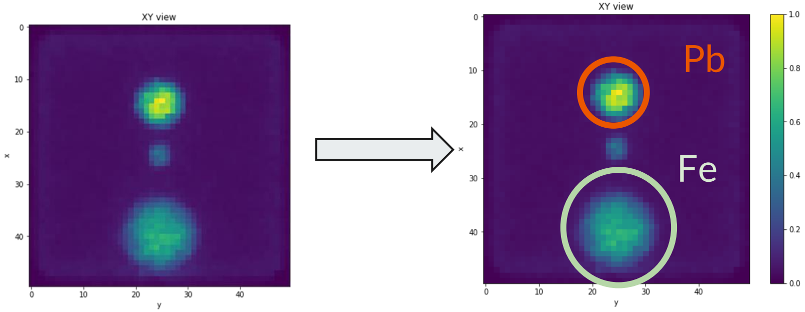
b - Angle Statistic Reconstruction algorithm (ASR)



Typical MST reconstruction parameters

- Cuts on scattering angles
- Noise reduction sensitivity

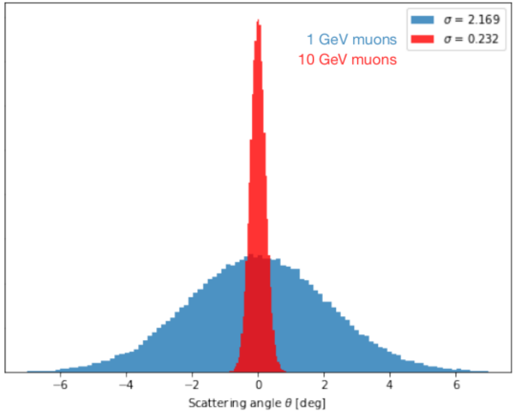
II - Material classifiers

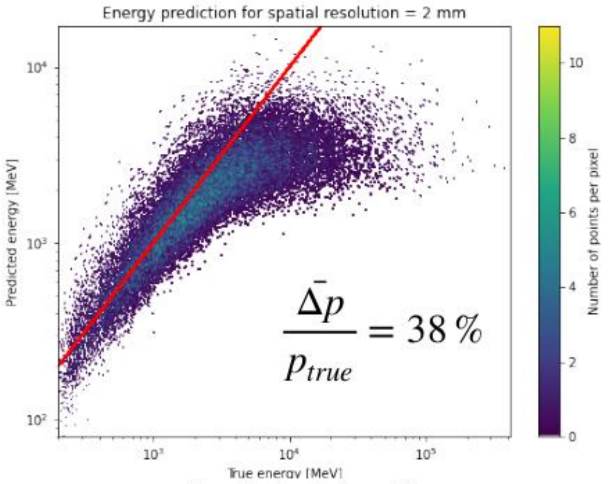


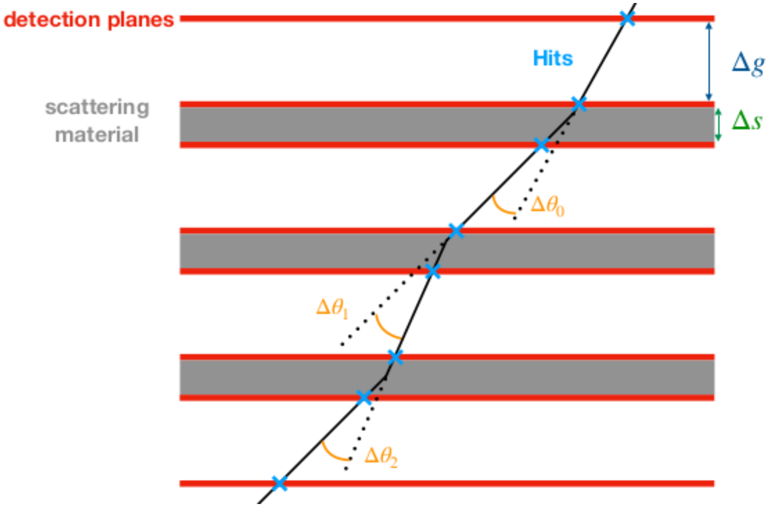
Muon momentum measurement

Muon **scattering amplitude**

$$\propto \frac{1}{p}\sqrt{\frac{x}{X_0}}$$





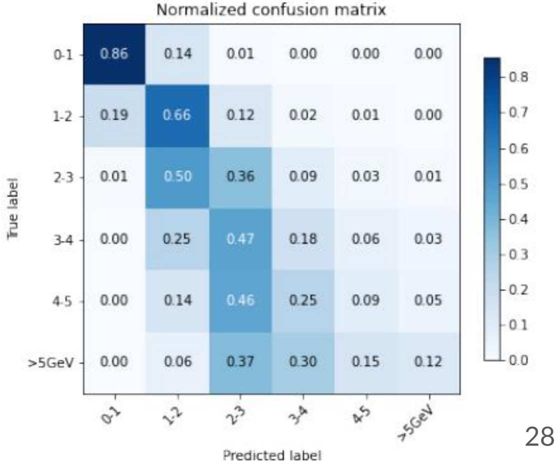


Scattering angle measurement

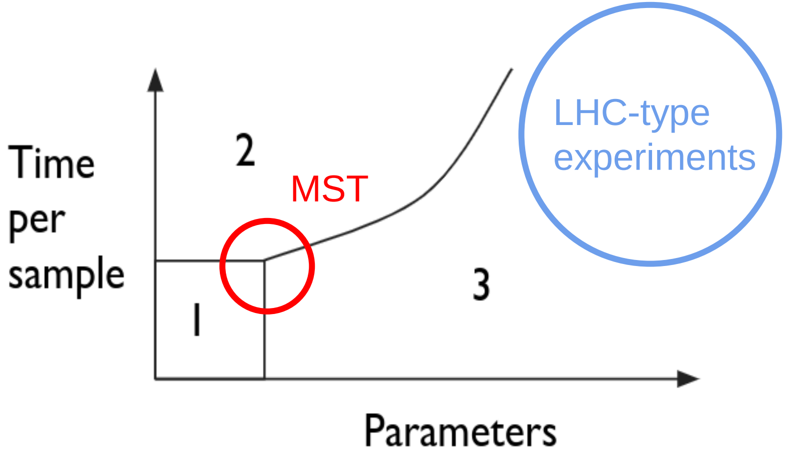
$$\theta_{RMS} = \frac{1}{3}(\theta_1^2 + \theta_2^2 + \theta_3^2)$$

Multiple Coulomb scattering model

$$p = \frac{13.6MeV}{\theta_{RMS}}\sqrt{\frac{x}{X_0}}$$



Parameter space in MST



1. Grid/random search
2. Bayesian optimisation, Simulated annealing, genetic algorithm, particle swap optimisation, …
3. Gradient-based optimisation: Newtonian, gradient descent, BFGS, …

Hardware

- Tracking system **technology** (RPC's, scintillators, micromegas, drift tubes, etc..)
- **Spatial resolution**
- **Efficiency**
- **Tracking system** (# planes, dimensions, geometry)

Software

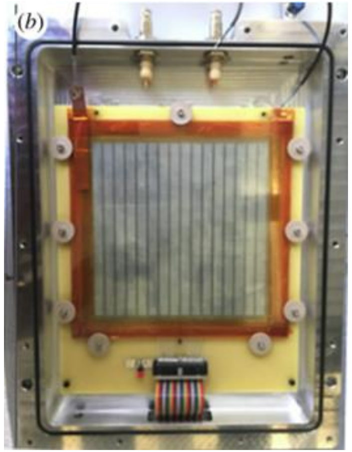
- **Reconstruction algorithms**
- **Material classifiers**
- **Image recognition, clustering**

Detector parameters and cost

Given its design and technology choices, how to estimate detector cost?

Local cost γ

Cost specific to the technology used



Sealed RPC prototype in development at UCLouvain

Local cost γ

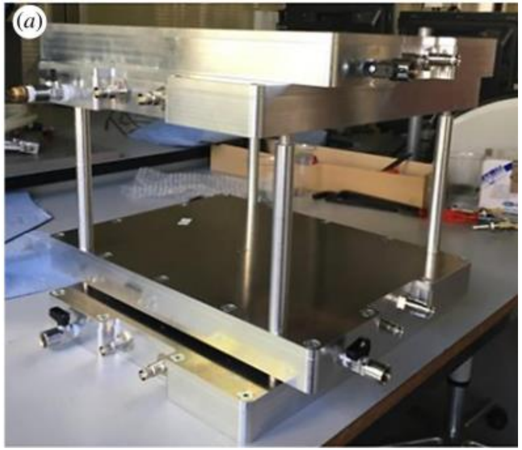
$$\gamma_{technology} = \gamma(x)$$

with x the performance properties of the given technology e.g time, spatial resolution, efficiency

$$\gamma\,[\,m^{-2}.readout^{-1}\,]$$

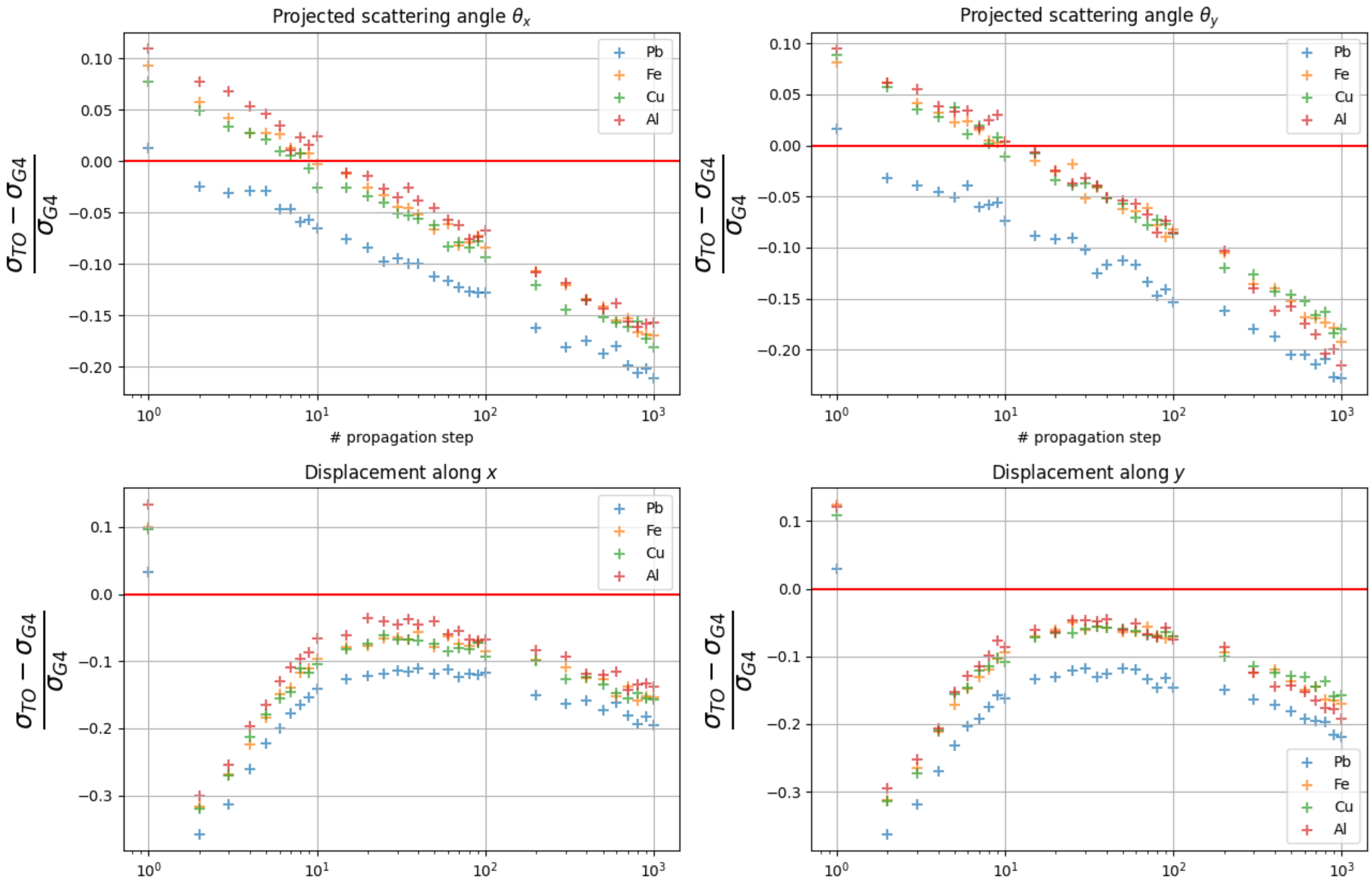
Global cost C(γ,φ)

Describe overall detector conception

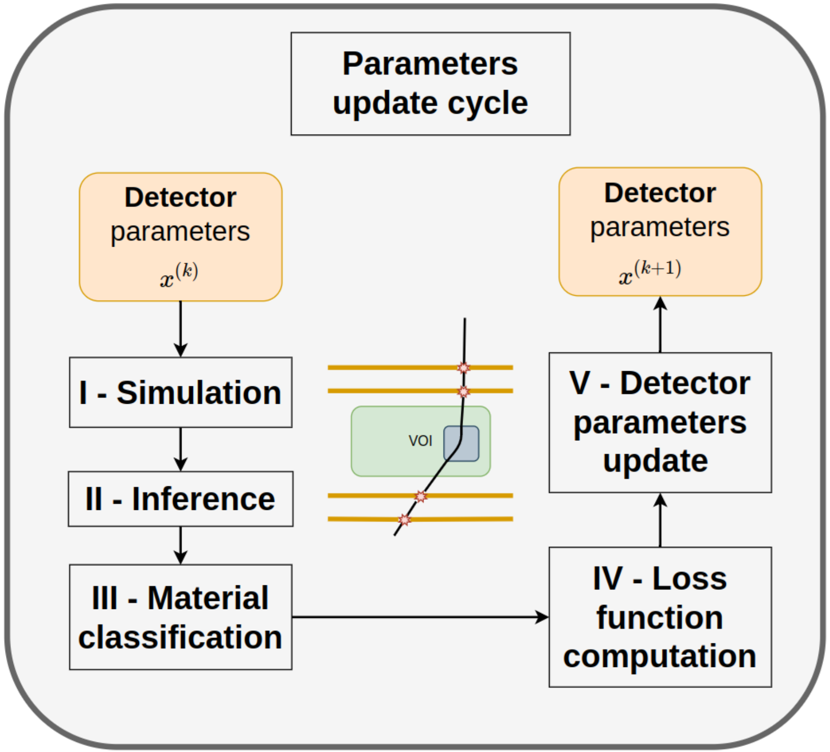


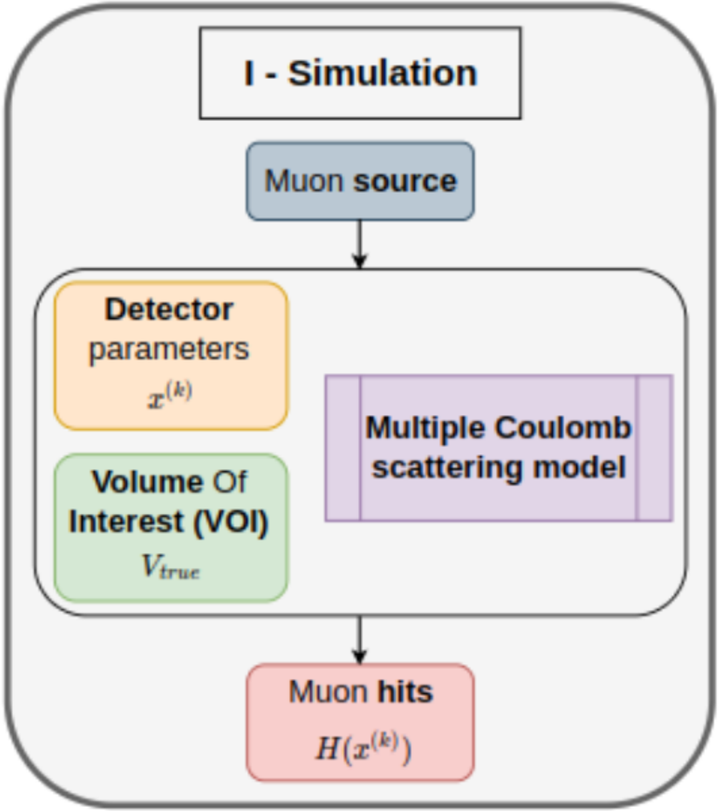
Portable muoscope in development at UCLouvain

50 GeV muon propagation through 50cm thick material block

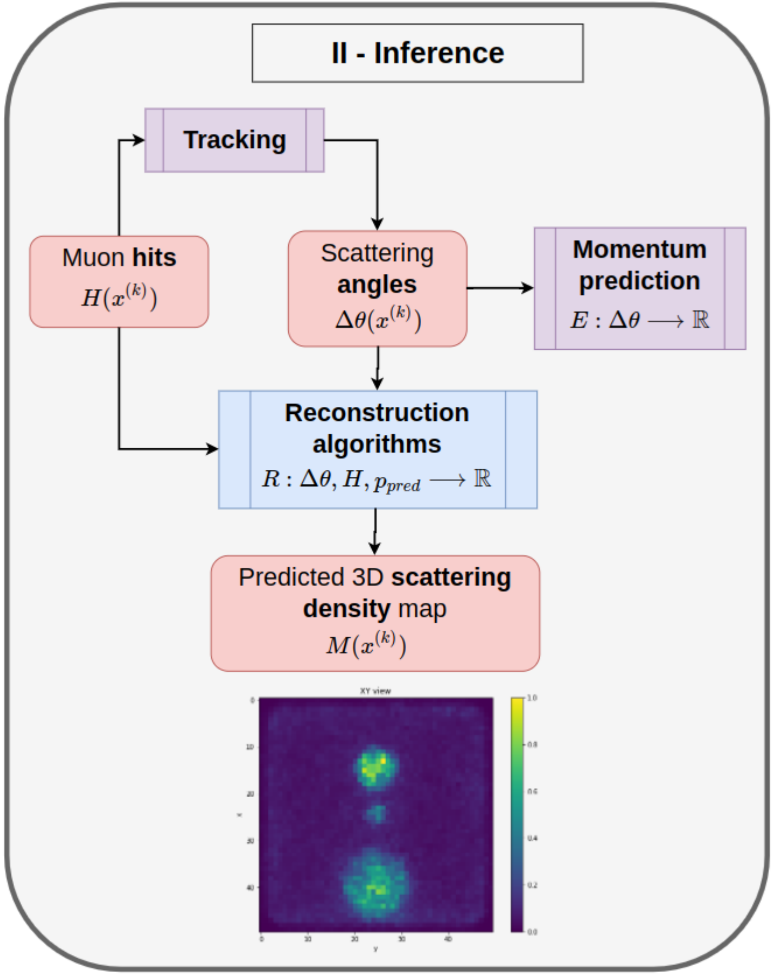


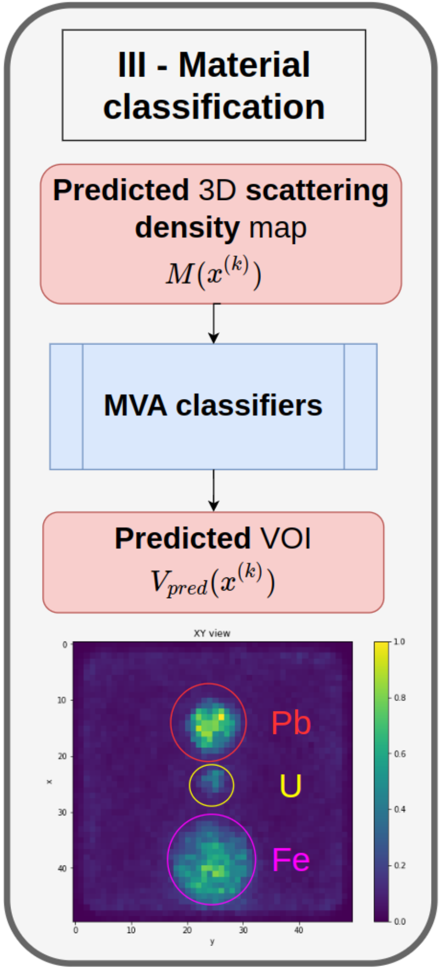
TomOpt


Parameters update cycle
Detector parameters $x^{(k)}$
Detector parameters $x^{(k+1)}$
I - Simulation
II - Inference
III - Material classification
IV - Loss function computation
V - Detector parameters update
VOI

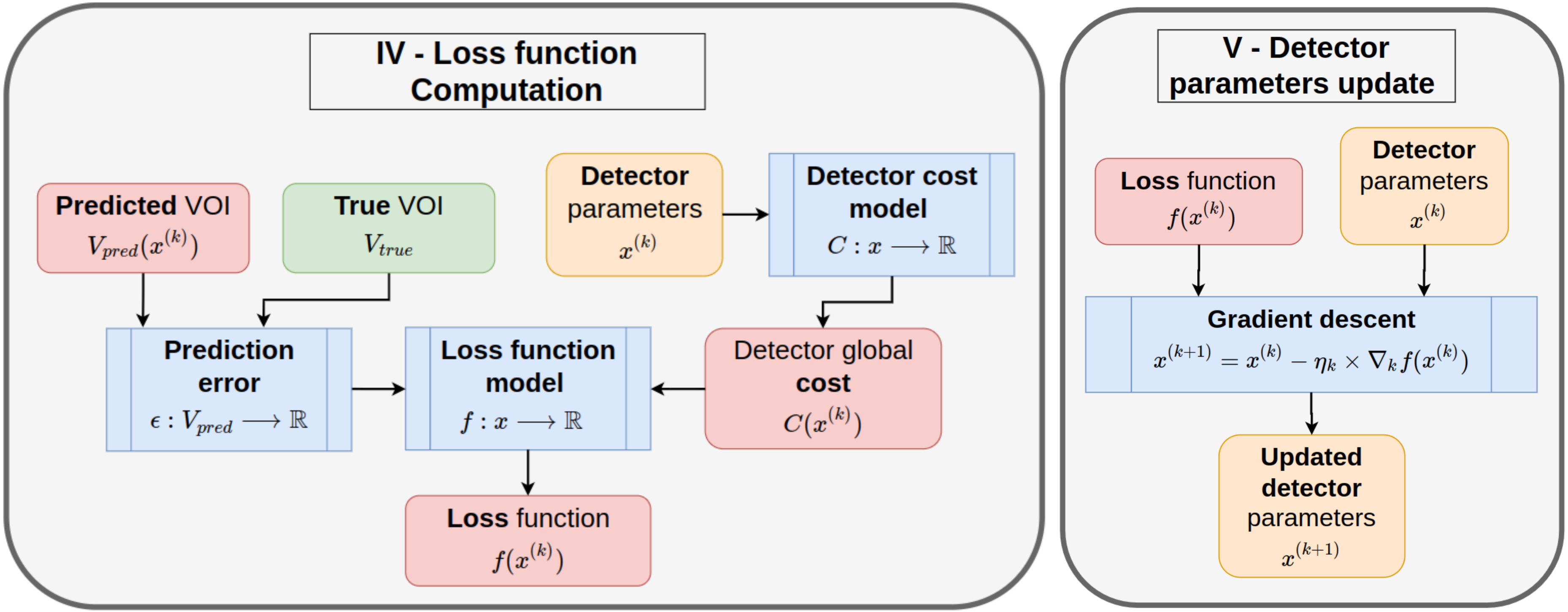

I - Simulation
Muon source
Detector parameters $x^{(k)}$
Volume Of Interest (VOI) V_{true}
Multiple Coulomb scattering model
Muon hits $H(x^{(k)})$

TomOpt


II - Inference
Tracking
Muon hits
$H(x^{(k)})$
Scattering angles
$\Delta\theta(x^{(k)})$
Momentum prediction
$E : \Delta\theta \longrightarrow \mathbb{R}$
Reconstruction algorithms
$R : \Delta\theta, H, p_{pred} \longrightarrow \mathbb{R}$
Predicted 3D scattering density map
$M(x^{(k)})$
XY view


III - Material classification
Predicted 3D scattering density map
$M(x^{(k)})$
MVA classifiers
Predicted VOI
$V_{pred}(x^{(k)})$
XY view
Pb
U
Fe

TomOpt



IV - Loss function Computation
Predicted VOI
$V_{pred}(x^{(k)})$
True VOI
V_{true}
Detector parameters
$x^{(k)}$
Detector cost model
$C : x \longrightarrow \mathbb{R}$
Prediction error
$\epsilon : V_{pred} \longrightarrow \mathbb{R}$
Loss function model
$f : x \longrightarrow \mathbb{R}$
Detector global cost
$C(x^{(k)})$
Loss function
$f(x^{(k)})$
V - Detector parameters update
Loss function
$f(x^{(k)})$
Detector parameters
$x^{(k)}$
Gradient descent
$x^{(k+1)} = x^{(k)} - \eta_k \times \nabla_k f(x^{(k)})$
Updated detector parameters
$x^{(k+1)}$