ML AND DIFFERENTIABLE PROGRAMMING OPTIMIZATION FOR X-RAY EXPERIMENTS

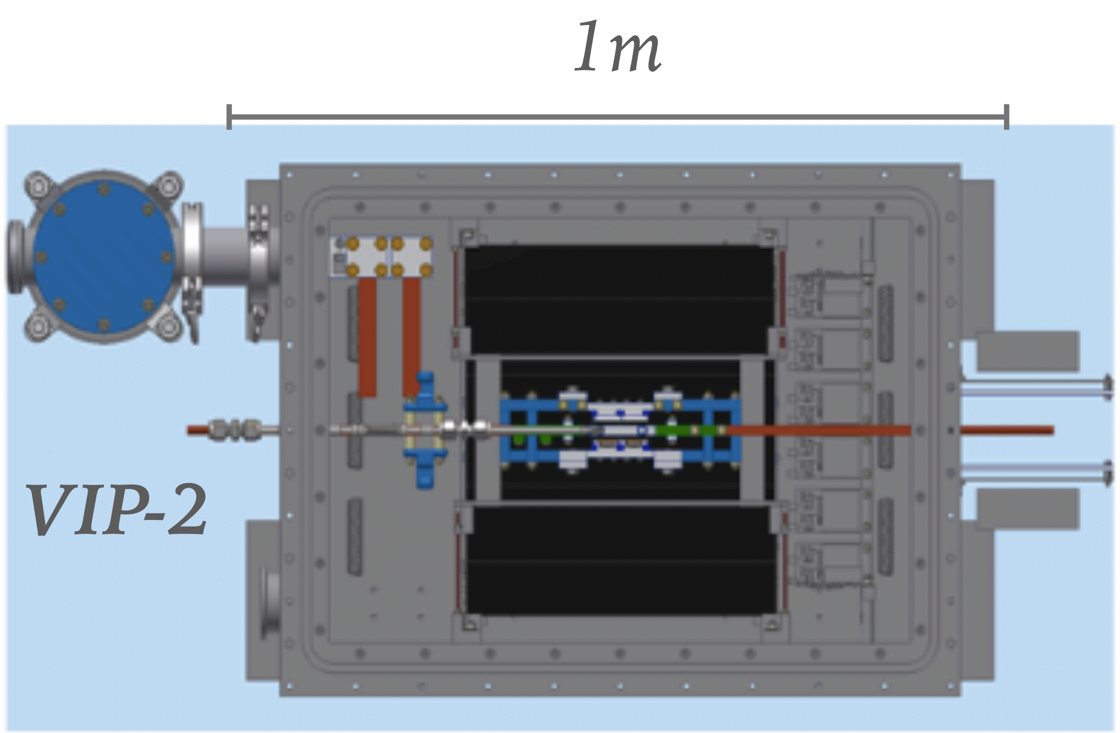
Fabrizio Napolitano

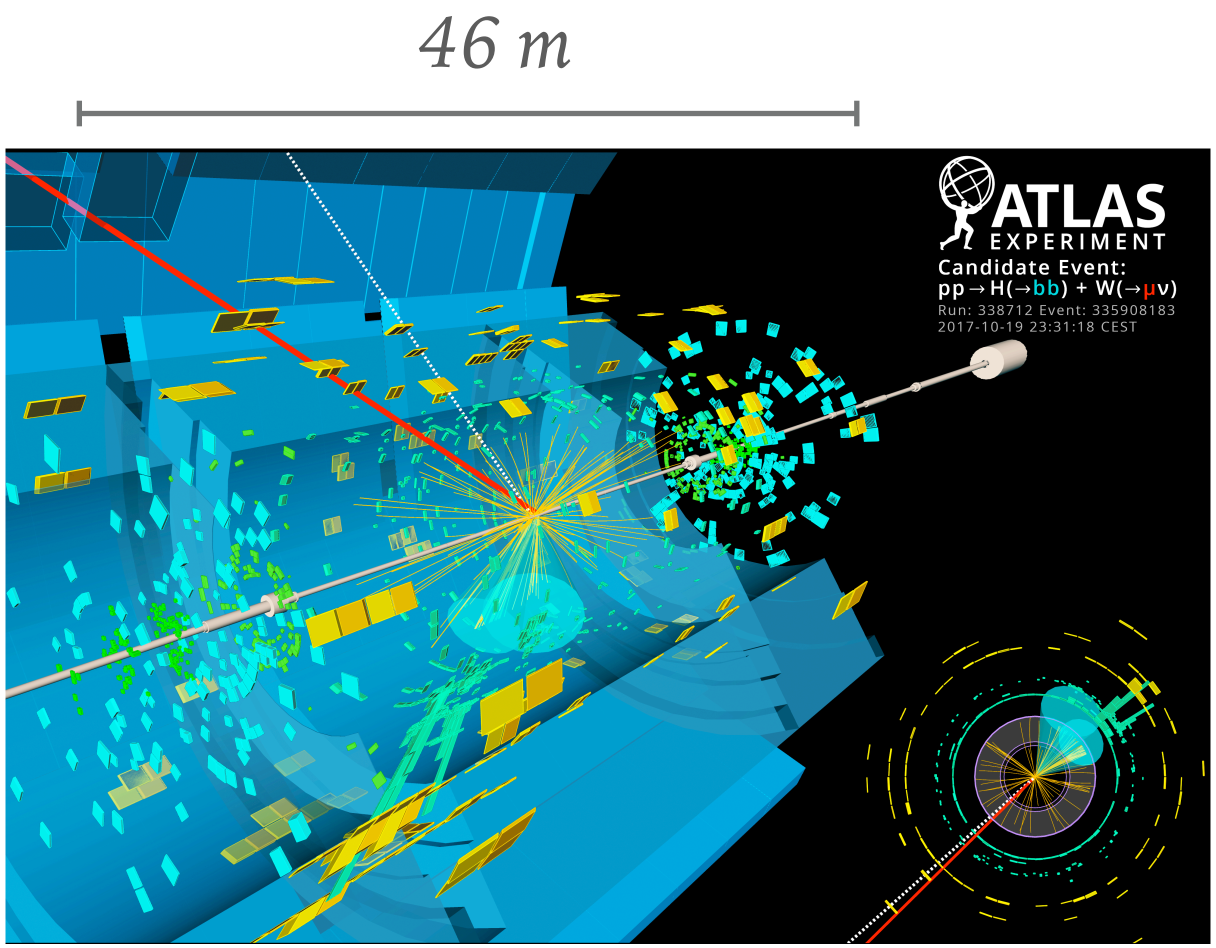
fabrizio.napolitano@lnf.infn.it

Bridging Scales: at the crossroads among renormalization group, multi-scale modeling, and deep learning

ECT Trento, 15/04/24*

Bridging Scales: ML & complexity





Size, Complexity, Energy

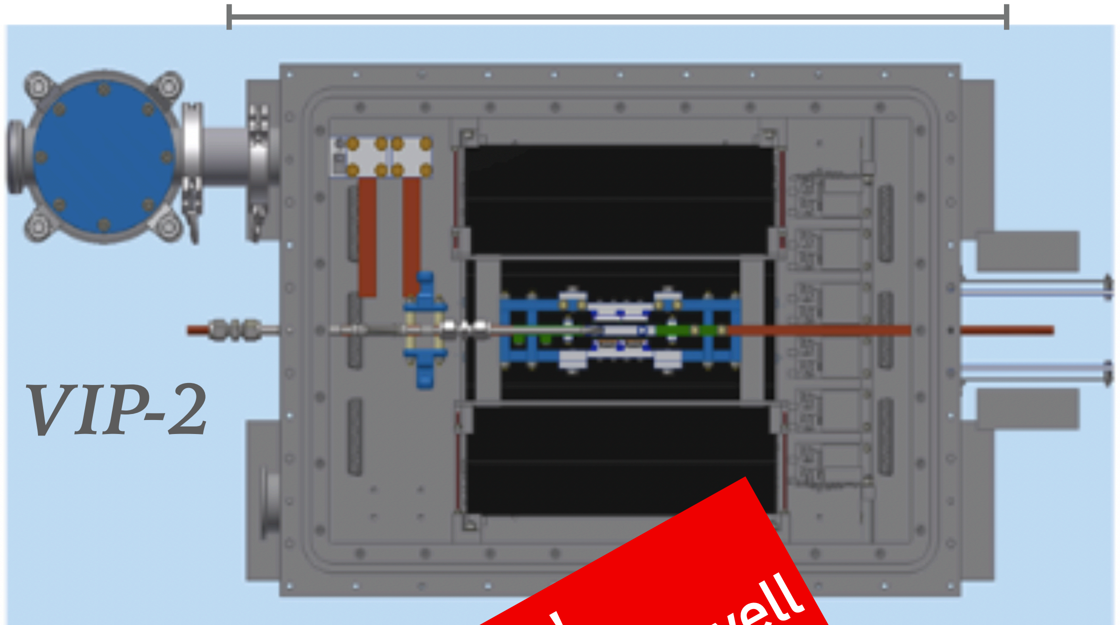
X-ray experiments
Low data dimensionality
Not obvious gains

High Energy Physics
Very high dimensional
A lot to (machine) learn

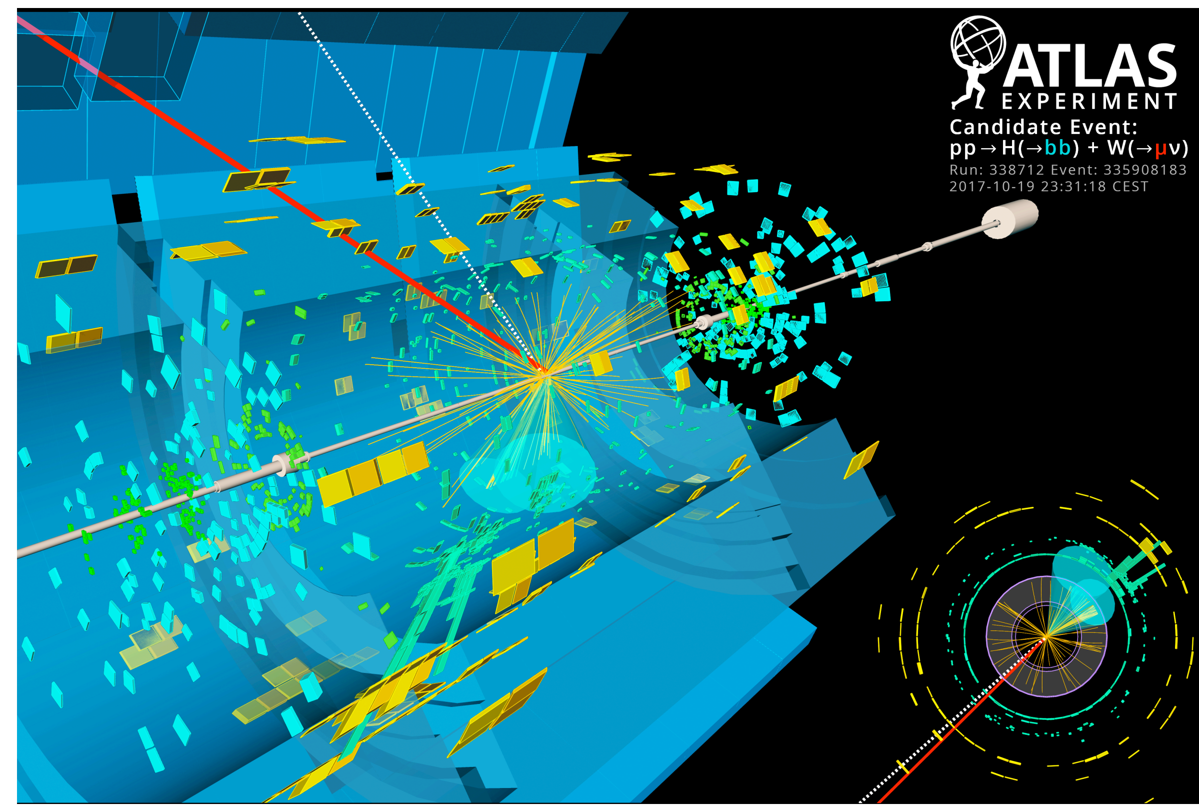
ALPACA @ECT* last year

Bridging Scales: ML & complexity

1m



46 m



Focus here!
Actually a lot to gain as well

Size, Complexity, Energy

X-ray experiments
Low data dimensionality
Not obvious gains

High Energy Physics
Very high dimensional
A lot to (machine) learn

ALPACA @ECT* last year

INFN Frascati National Labs

Our building

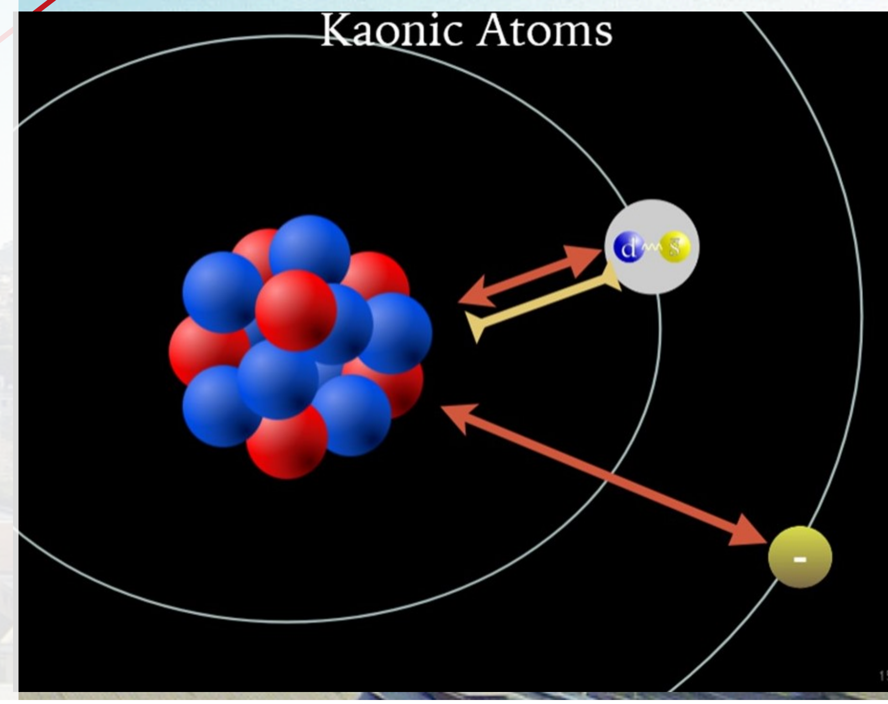
DAFNE collider

Frascati

LINAC

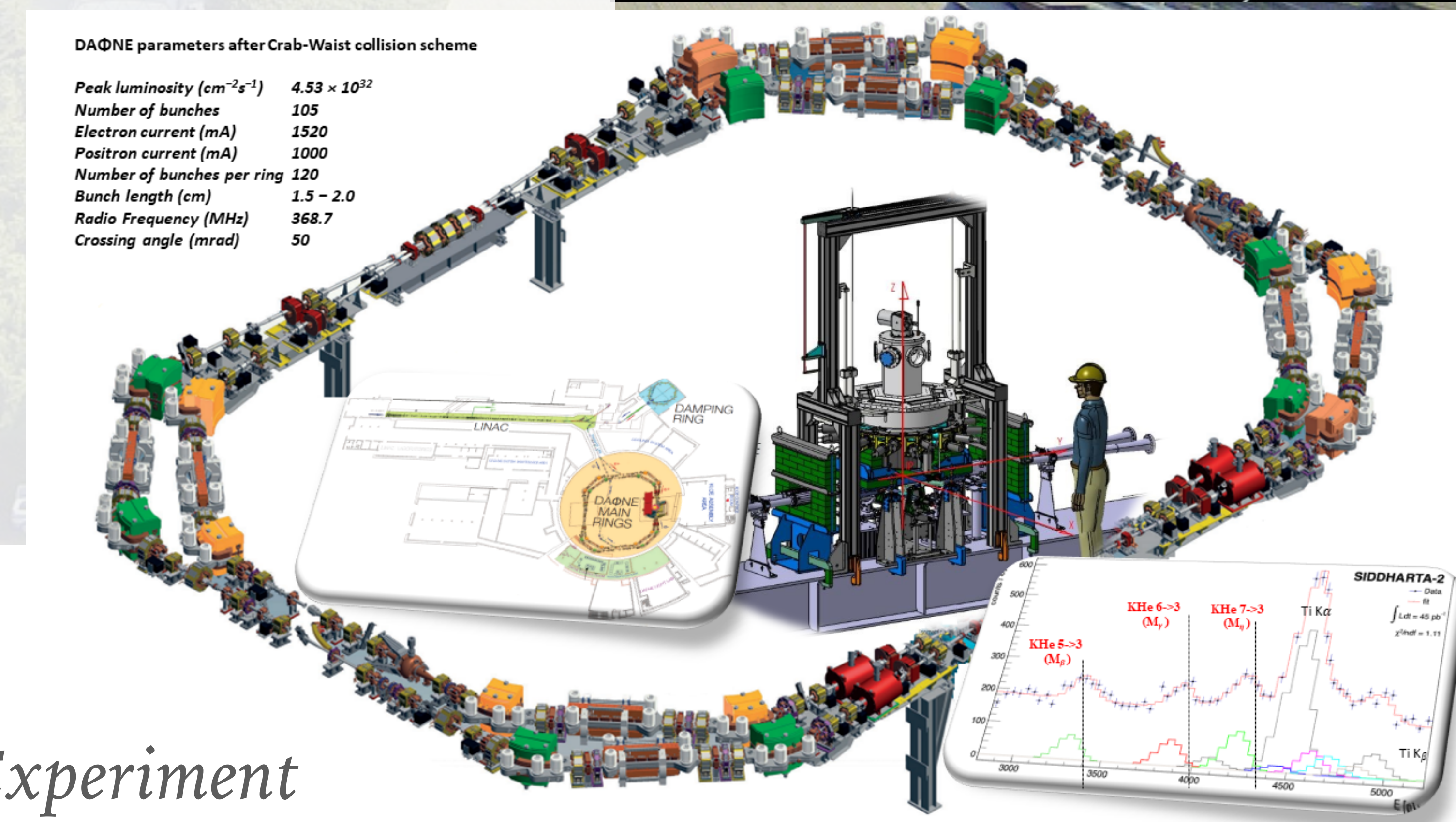
INFN Frascati National Labs

Our building *DAFNE collider* *Frascati*



- *Study kaonic matter via X-rays*
- *Interesting for low-energy QCD in the strangeness sector*
- *Implications for neutron stars EoS & astrophysics*
- *Dark Matter*
- *Nuclear and fundamental physics*
- *In data taking in this moment*
- *Precision measurement*

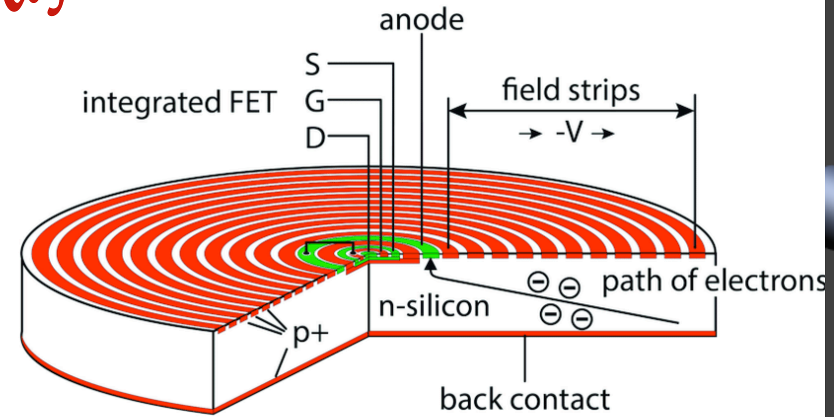
LINAC

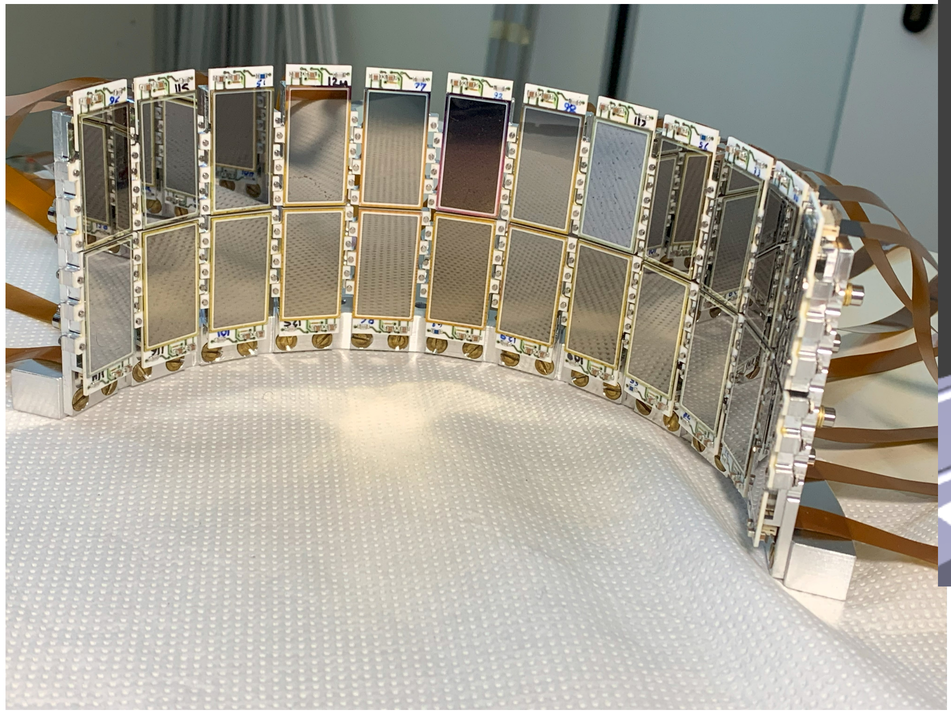


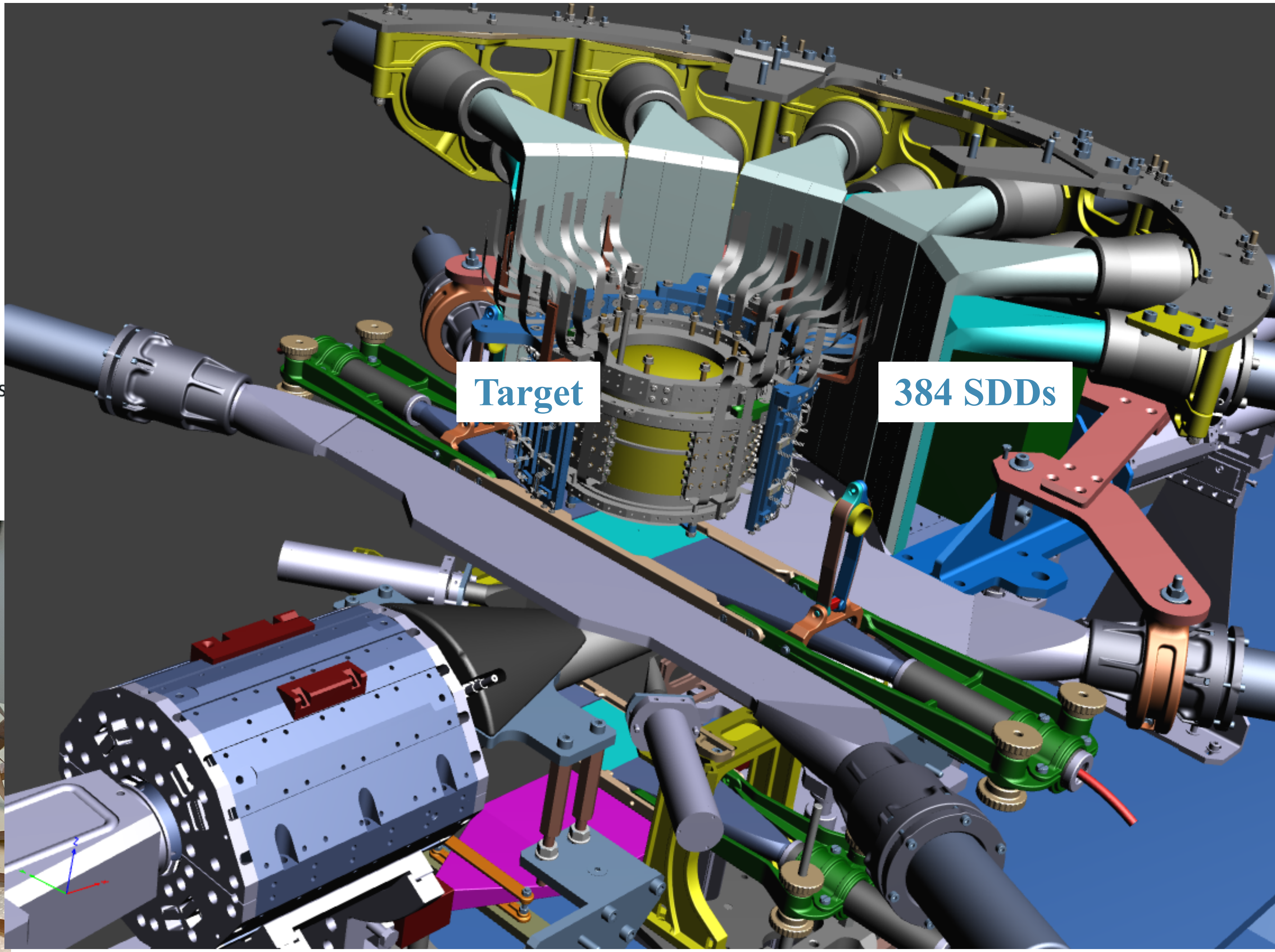
SIDDHARTA-2 Experiment

*Silicon Drift Detectors (**SDDs**) high (190 eV FWHM at 8.0 →keV), faster (triggerable) detectors. Arrays of 2 x 4 SDDs 8mm x 8mm each, 450 um thick*

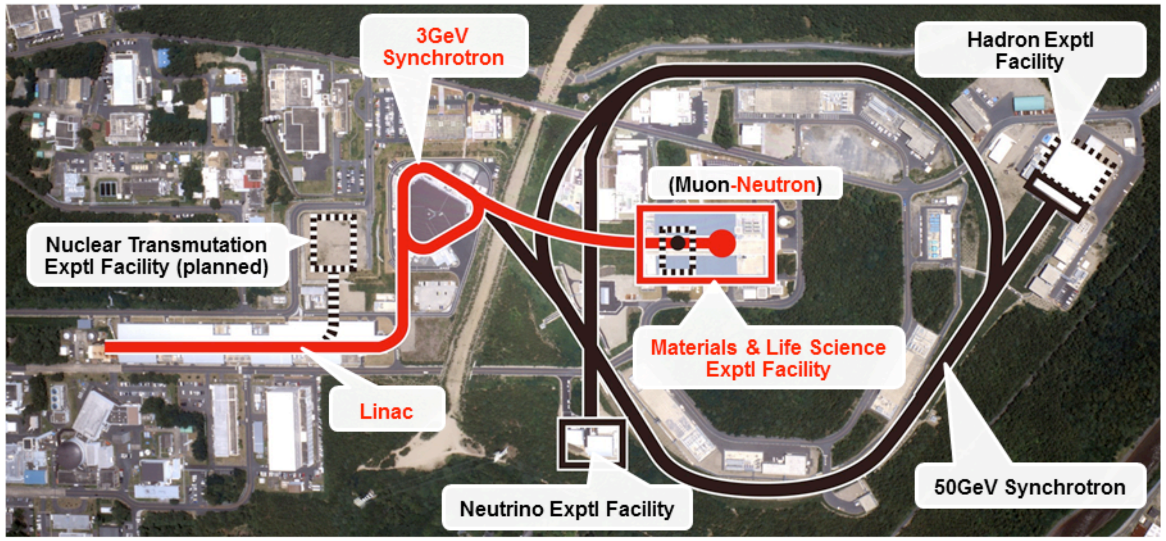
Ideal tool for precision X-ray spectroscopy





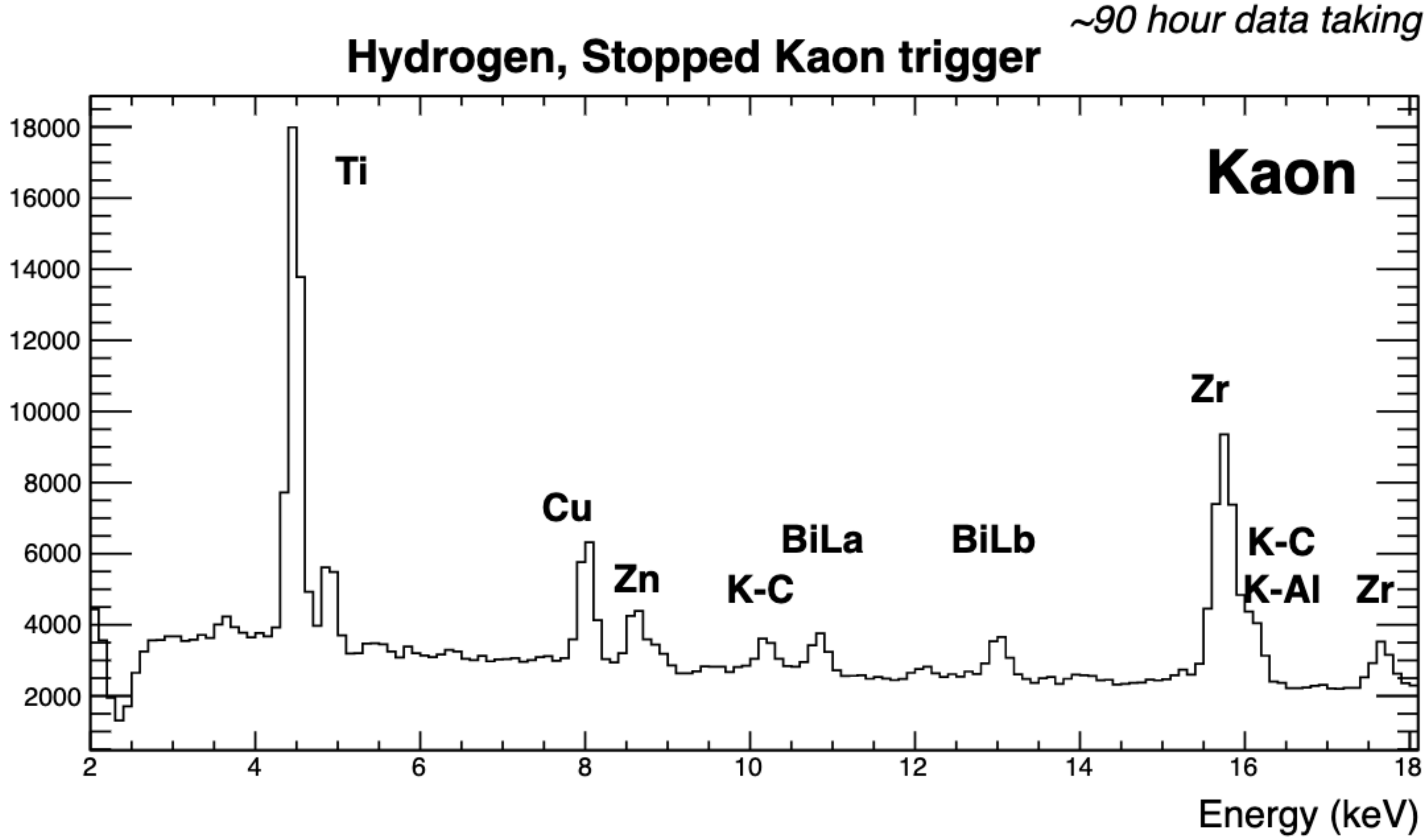


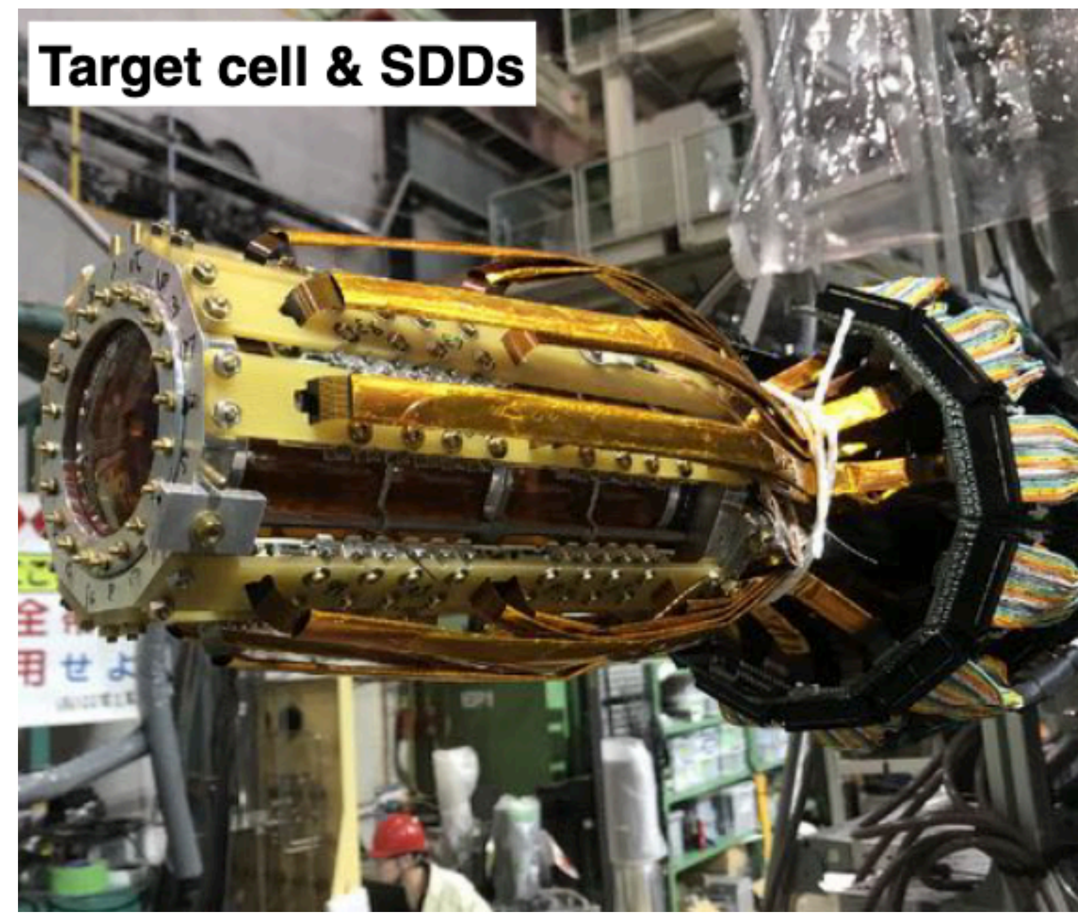
Kaonic atoms experiments at J-PARC



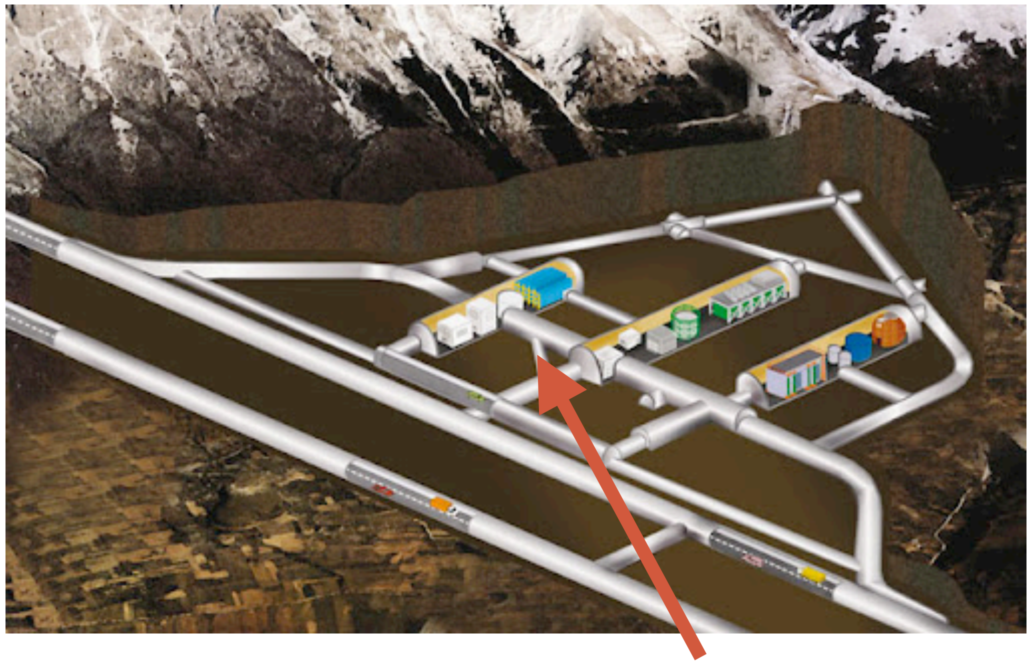
Hydrogen data in E57

~90 hour data taking



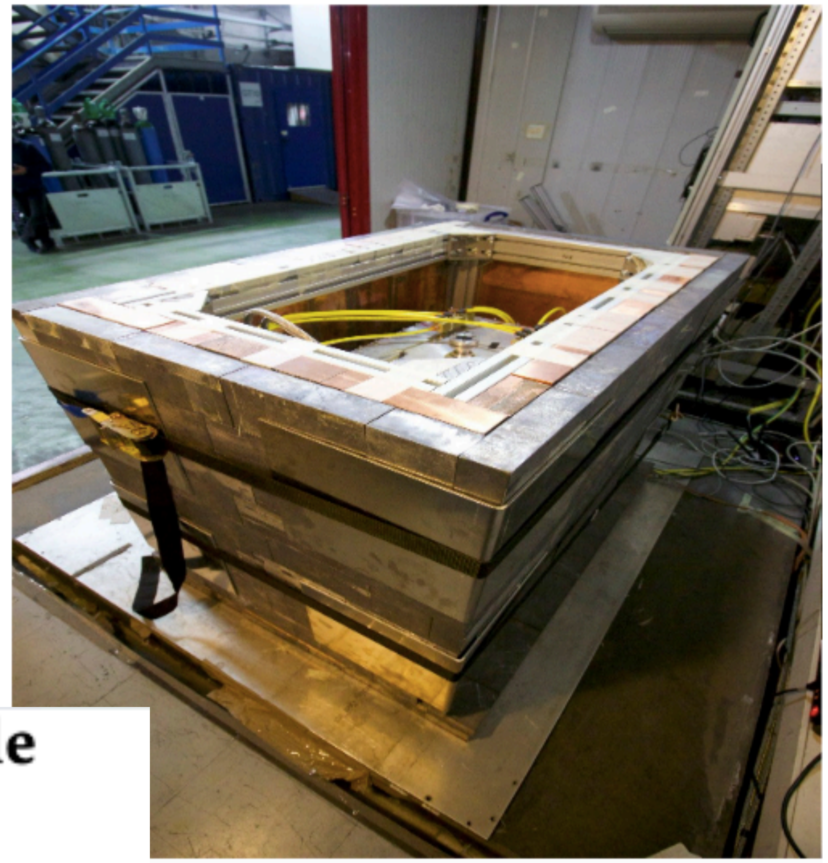


VIP-2 experiment at INFN-LNGS

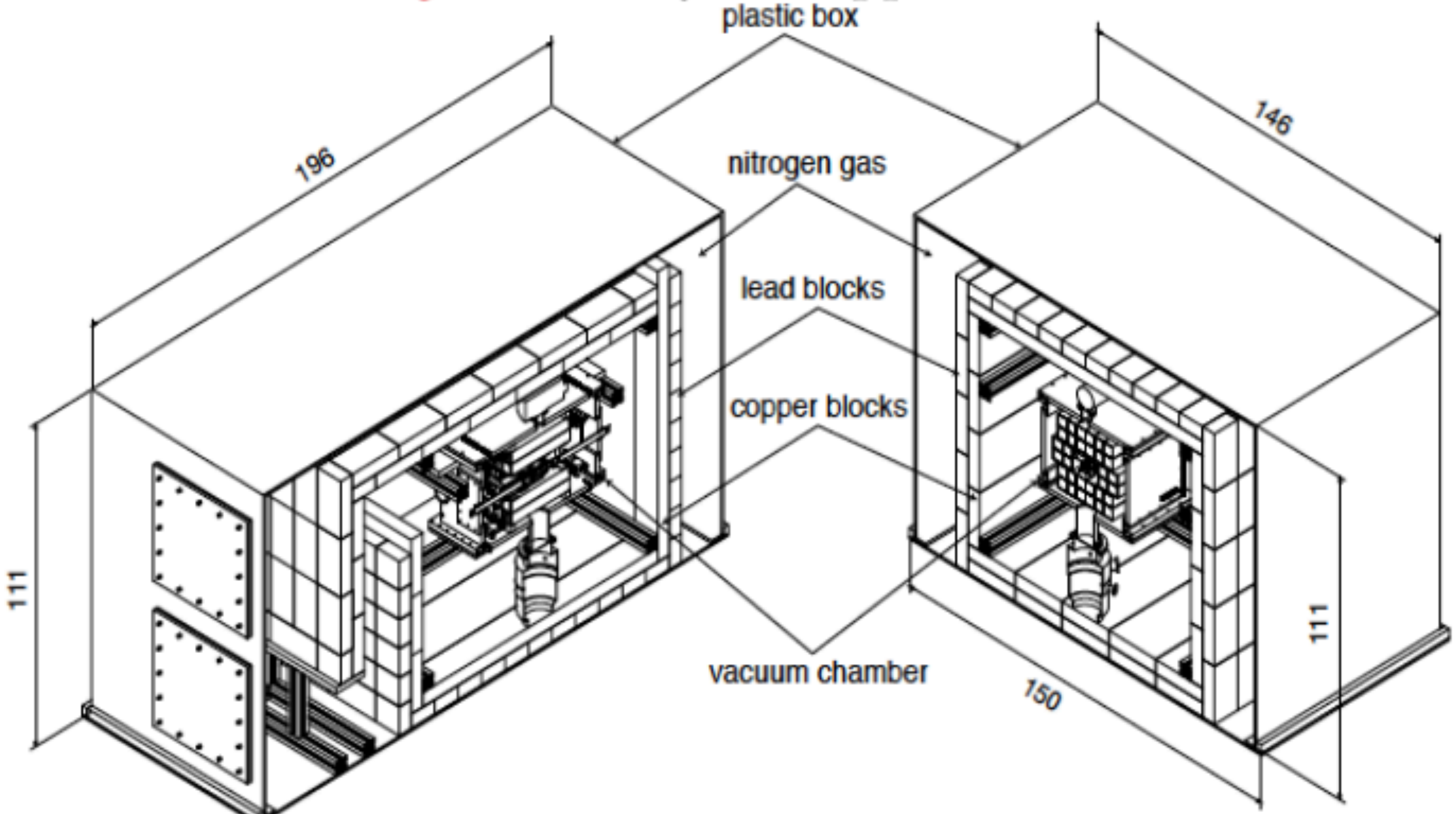


1400 m rock coverage

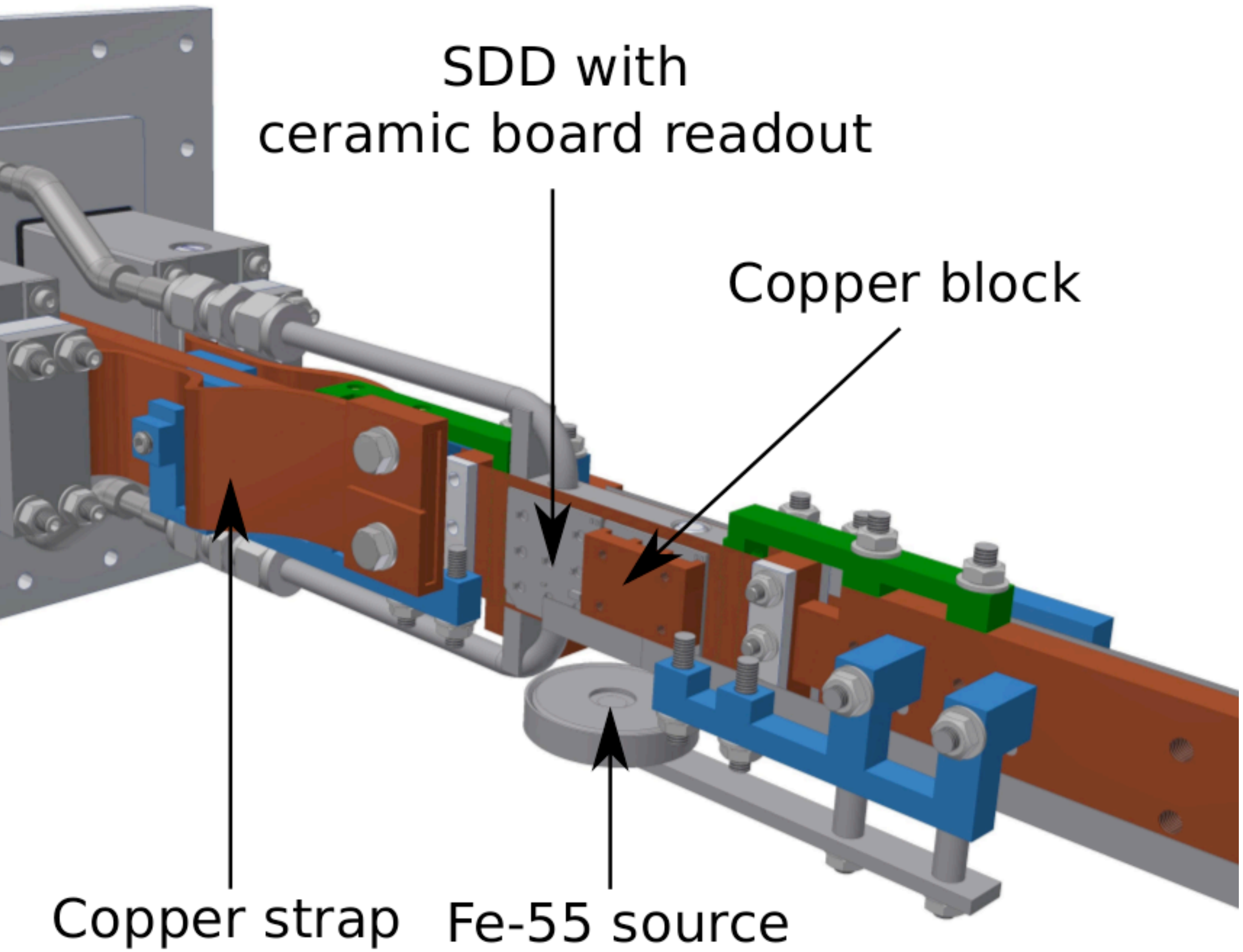
Upgrade concluded in April 2019:



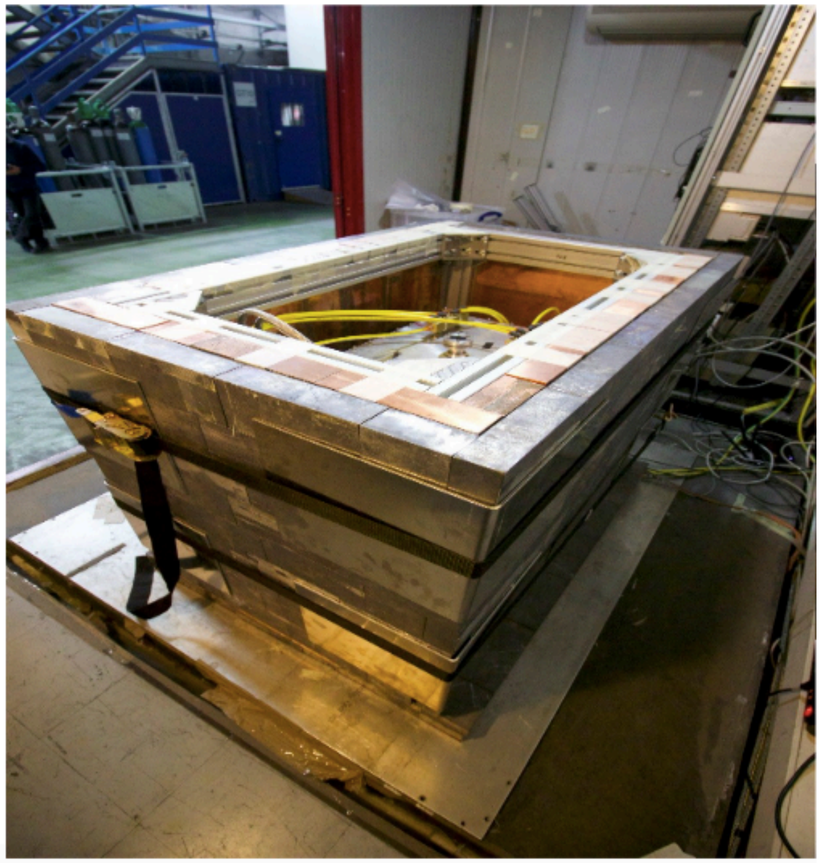
Passive scielding → two layers, copper inside lead outside



VIP-2 experiment at INFN-LNGS



Upgrade concluded in April 2019:

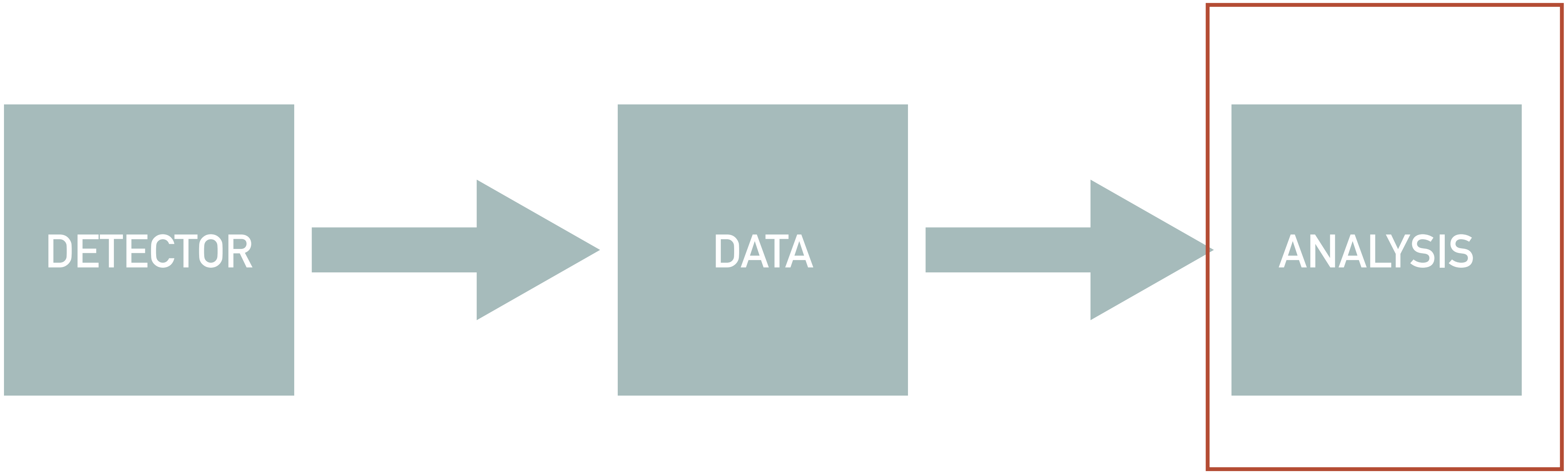


Searching for Pauli Exclusion Principle Violations

Quantum Gravity

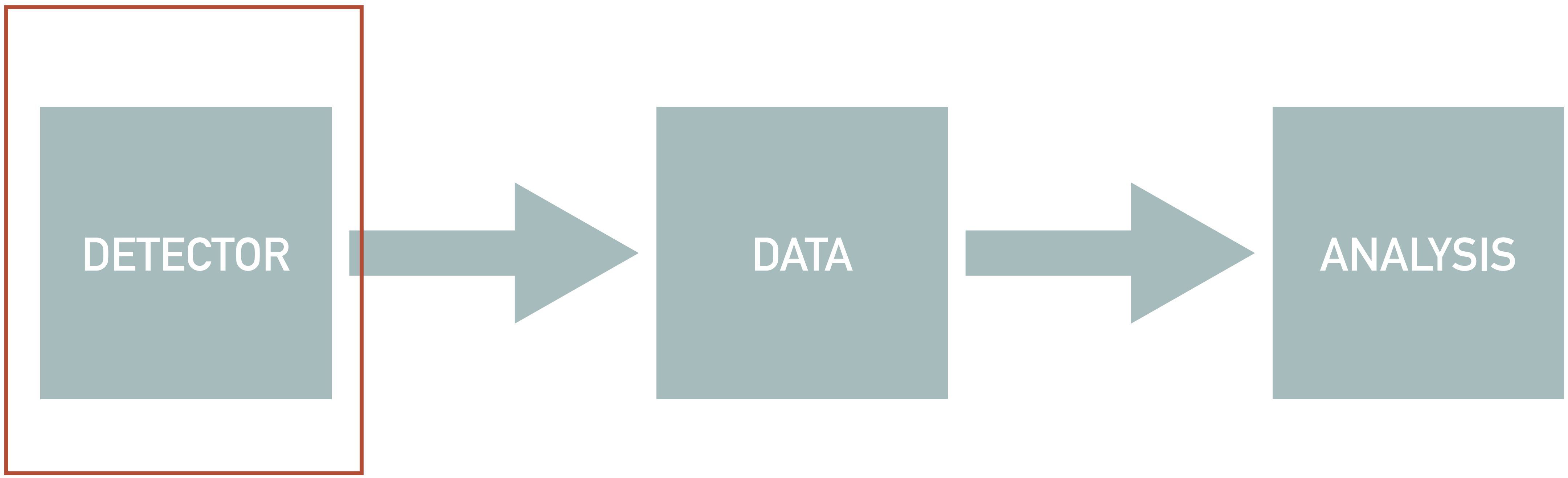
Machine Learning in physics experiment

Machine Learning in physics experiment



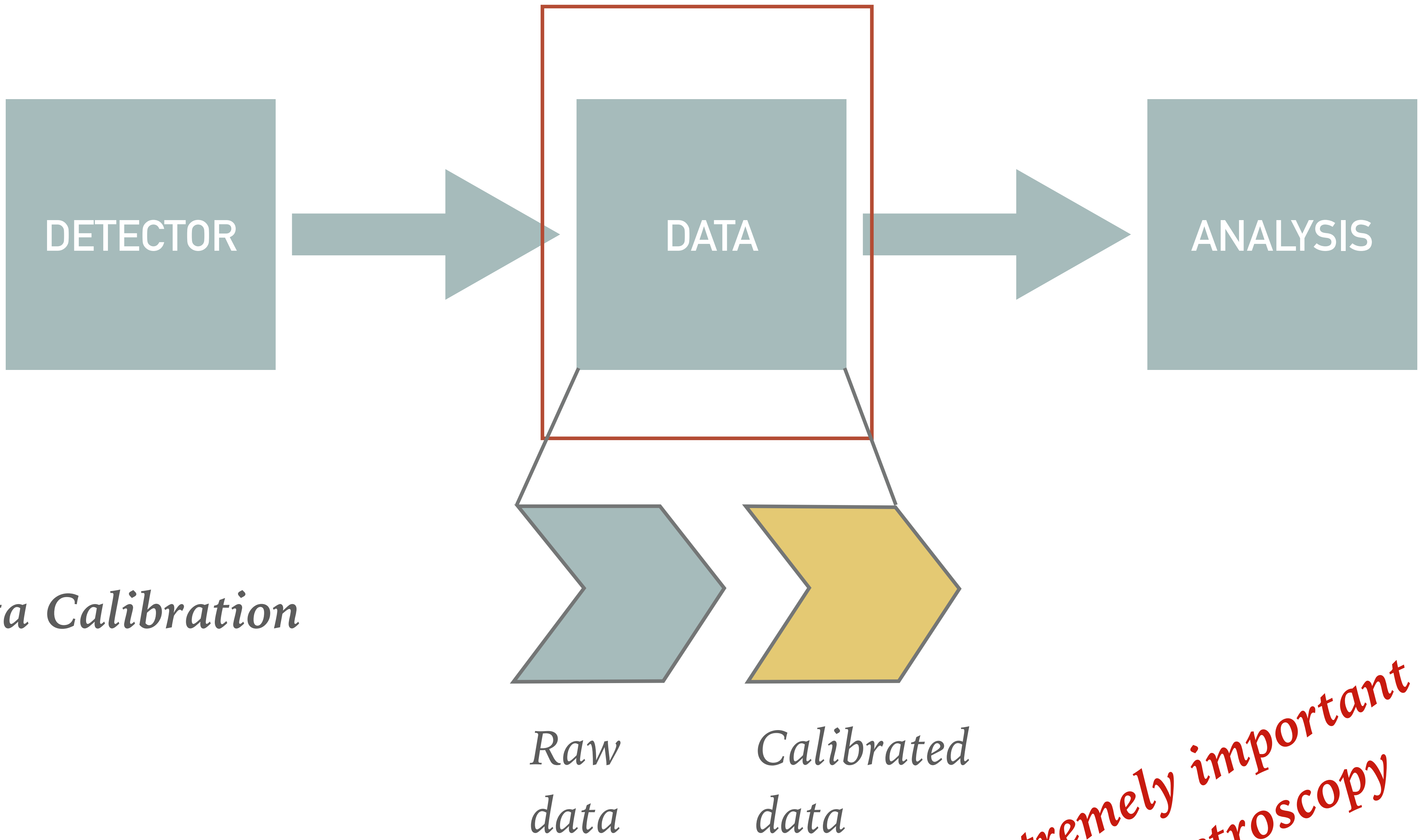
ML improving the analysis:
S vs B classification,
object reconstruction,
data simulation
etc

Machine Learning in physics experiment



e.g. optimization of detector geometry

Machine Learning in physics experiment



Data Calibration

Extremely important in x-ray spectroscopy https://www.sciencedirect.com/science/article/pii/S0370269311001225

Using the correction term ε in Eq. (2), the absolute energy of the kaonic ^{3}He $3d \to 2p$ transition was then determined to be:

$$E_{\mathrm{exp}} = E_{\mathrm{fit}} + \varepsilon = 6223.0 \pm 2.4(\mathrm{stat}) \pm 3.5(\mathrm{syst})\mathrm{eV}, \quad (4)$$

where the second term is the statistical error, and the third term is the systematic error. The latter was evaluated from the accuracy of the energy determination (±3.5 eV). Other contributions to the systematic error (e.g. effects of timing region selection and contributions of the kaonic oxygen line at 6.0 keV) are negligible.

Kaonic He-3/4 Measurement @ DAFNE

As a result, the 1s-level shift ϵ_{1s} and width Γ_{1s} of kaonic hydrogen were determined by SIDDHARTA to be

$$\epsilon_{1s} = -283 \pm 36(\mathrm{stat}) \pm 6(\mathrm{syst})\ \mathrm{eV} \quad \text{and}$$
$$\Gamma_{1s} = 541 \pm 89(\mathrm{stat}) \pm 22(\mathrm{syst})\ \mathrm{eV},$$

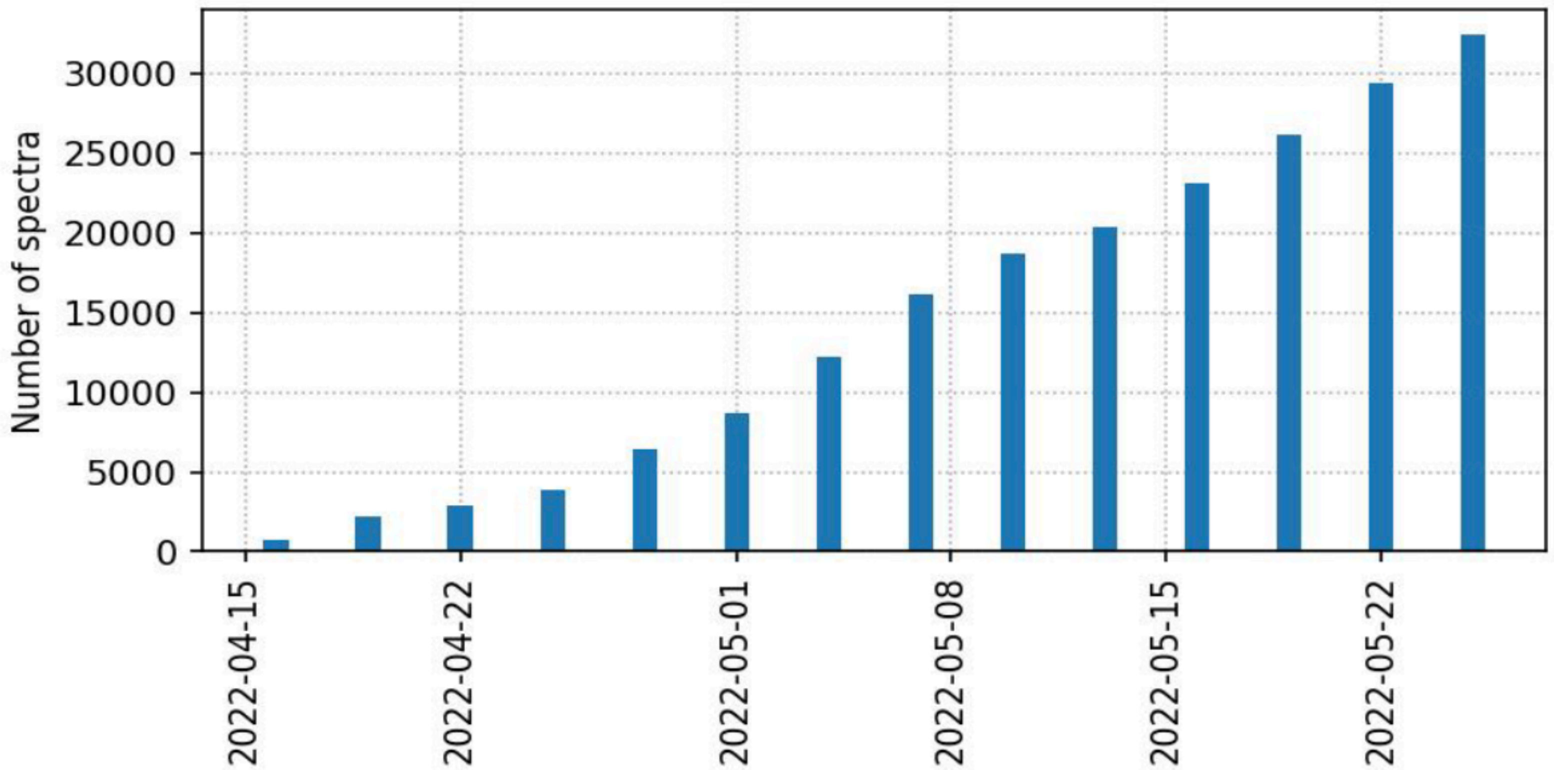
E62 J-PARC measurement

PRL 128, 112503 (2022)

TABLE I. Measured x-ray energies and widths of the kaonic ^{3}He and ^{4}He $3d \to 2p$ transitions, together with the summary of the statistical and systematic errors. Electromagnetic calculated energies are also tabulated. All the values are in units of eV.

	$K^-{}^3$He		$K^-{}^4$He	
	Energy	Width	Energy	Width
Measured ($E^{\mathrm{exp}}_{3d\to 2p}$, Γ_{2p})	6224.48	2.5	6463.69	1.0
Statistical error	0.40	1.0	0.27	0.6
Systematical error: total	0.18	0.4	0.11	0.3
(a) Absolute energy scale	0.17	...	0.09	...
(b) Detector resolution	0.01	0.2	0.01	0.1
(c) Low-energy tail	0.03	0.1	0.03	0.1
(d) Fitting robustness	0.05	0.3	0.05	0.3

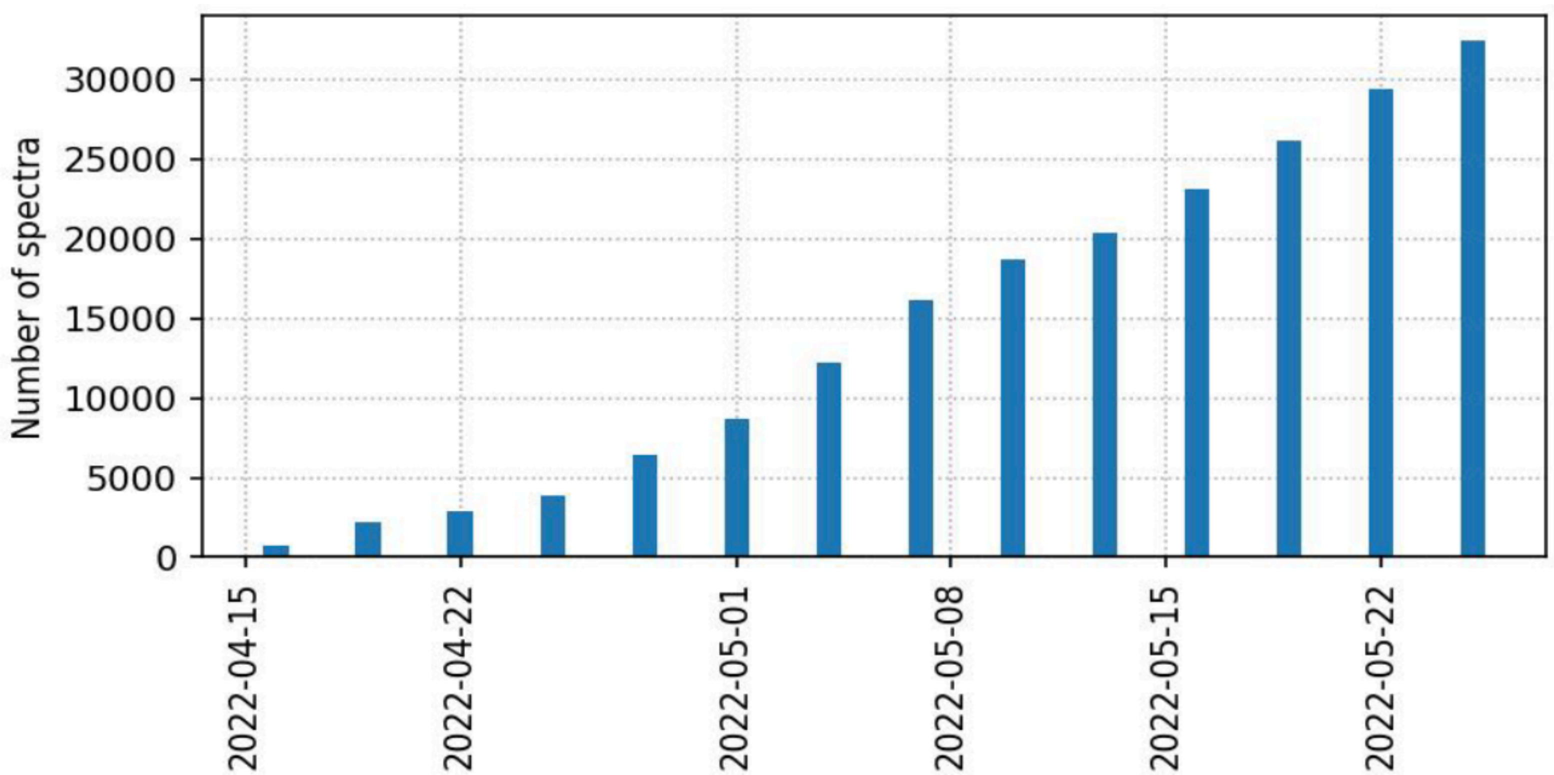
Number of spectra - SIDDHARTA-2 2022 Run



Calibration of SDD depends on:

- *Availability of calibration runs (depends on machine conditions/availability) (Ideally one wants the beam on 24/7)*
- *Dependence on temperature/pressure @ the detectors*
  - *Not precisely know (vacuum, cryo conditions, setup constraints)*
- *Dependence on beam background*

Number of spectra - SIDDHARTA-2 2022 Run



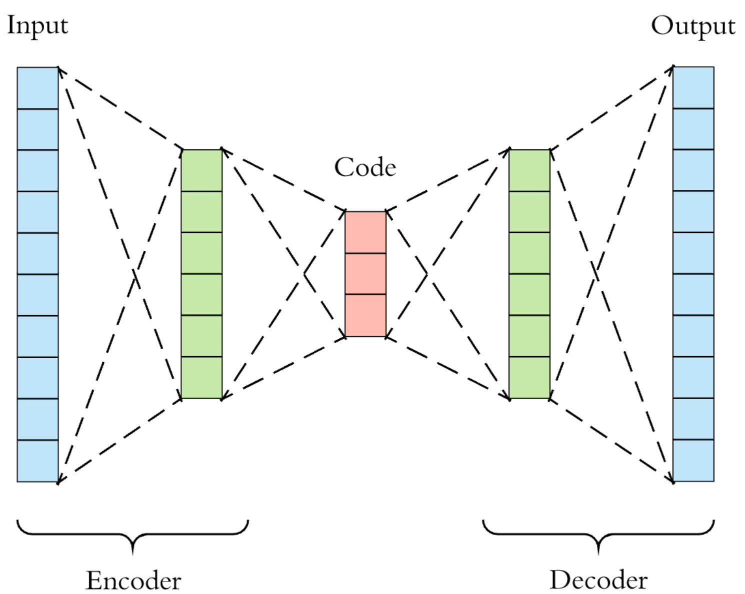
Calibration of SDD depends on:

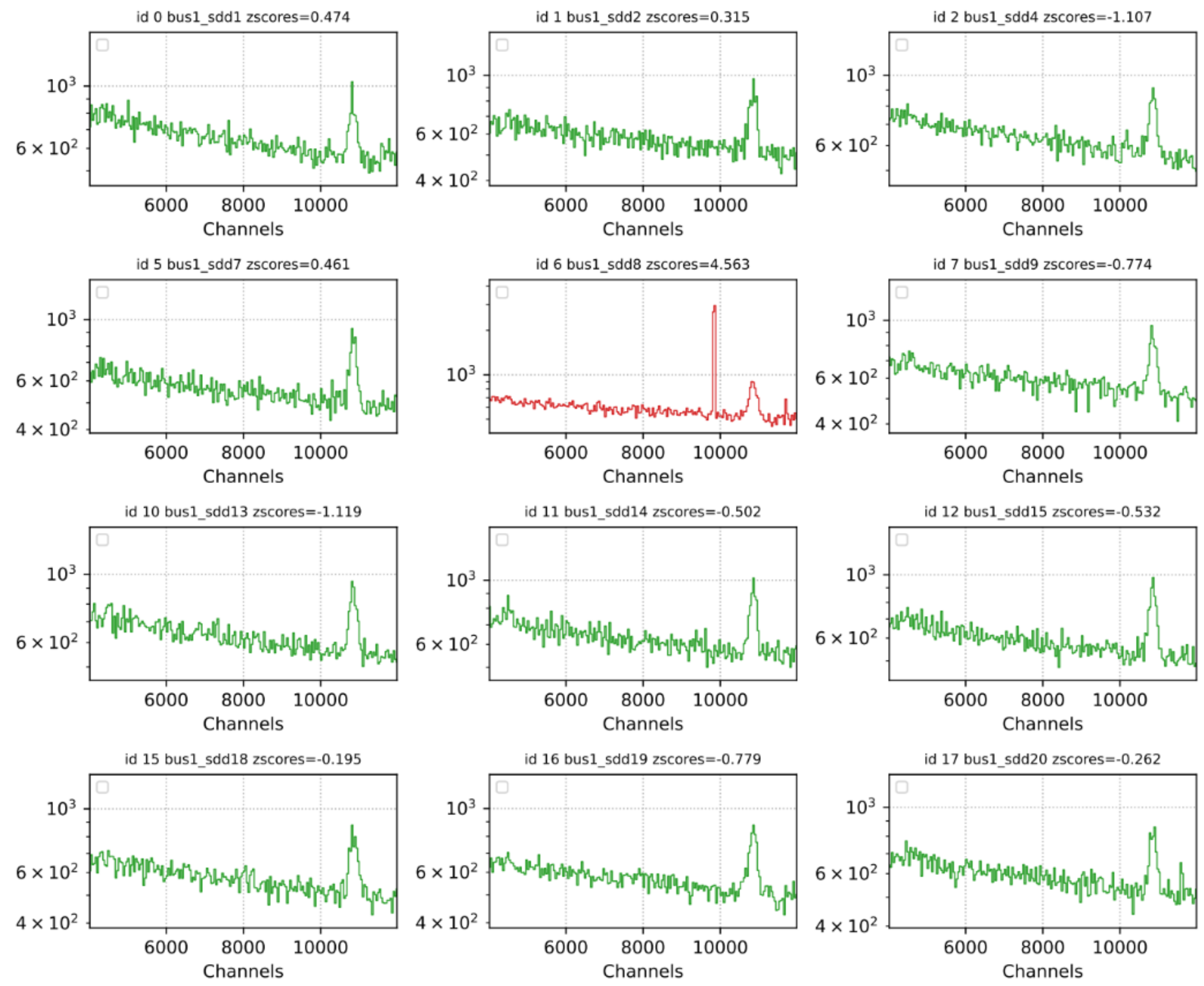
- *Availability of calibration runs (depends on … e conditions/availability) (Ideally one wants the beam on 24/7)*
- *Dependence on temperatur… @ the detectors*
  - *Not precisely know… …uum, cryo conditions, setup constraints)*
- *Dependence on beam background*

0th step: channel selection

The Monitoring Challenge in SIDDHARTA-2

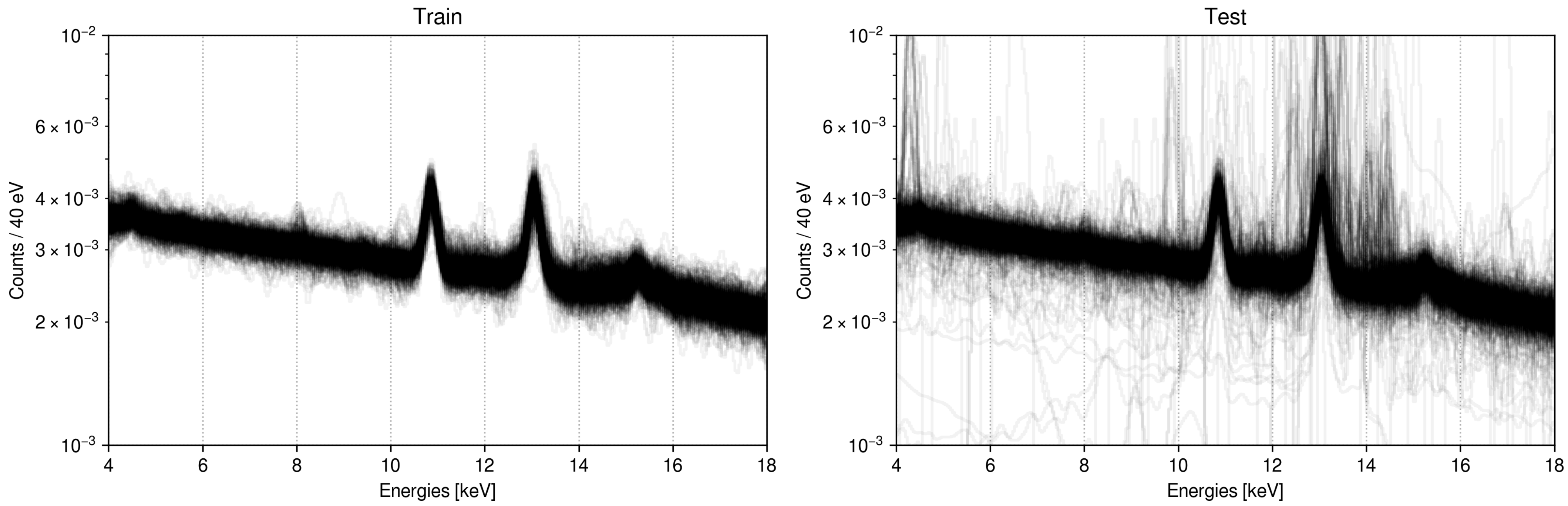
Autoencoder to detect anomalies



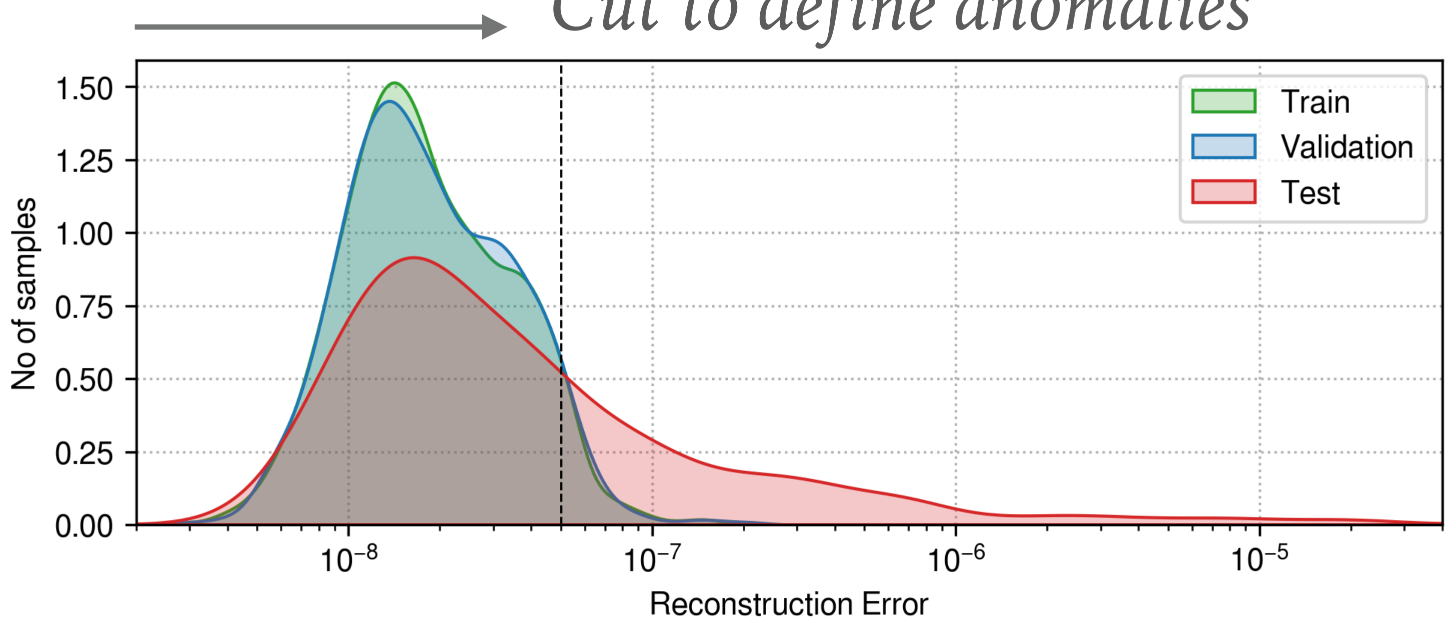


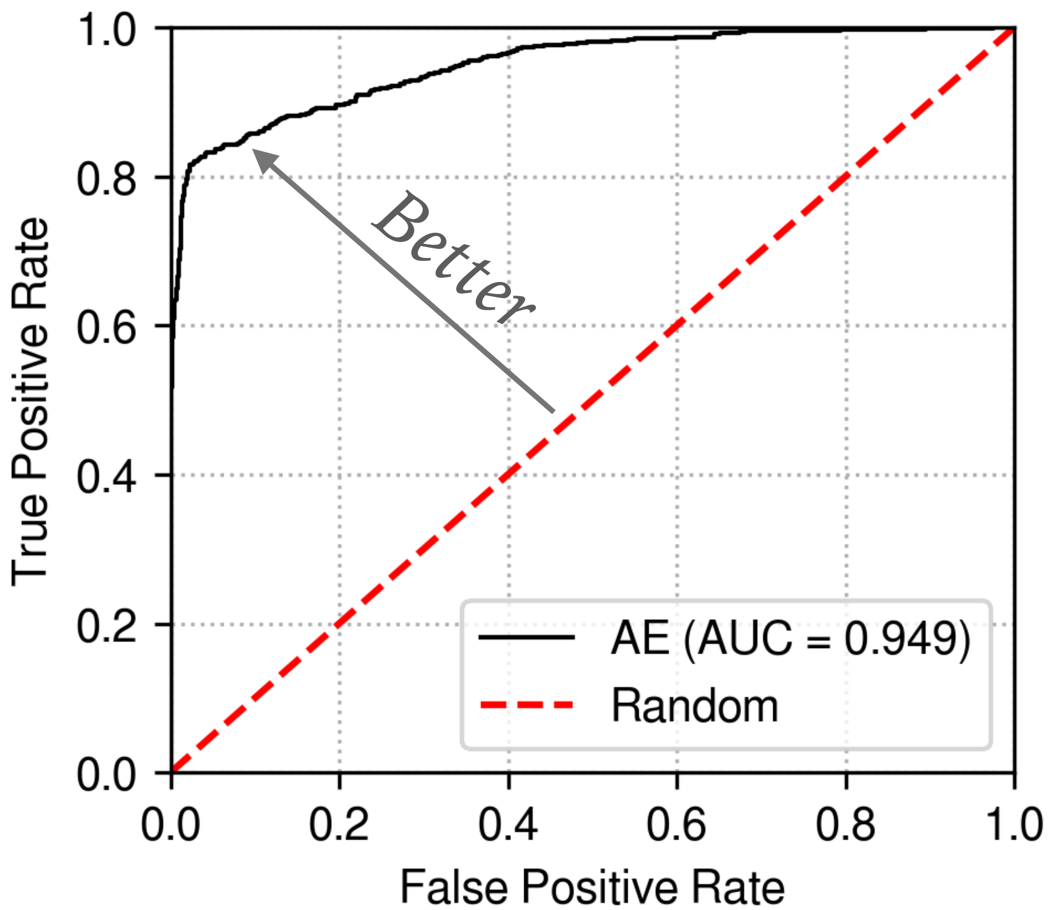
Anomalies due to various effects, physical and electronical

Automatize task -> gain in statistics of the final observable.



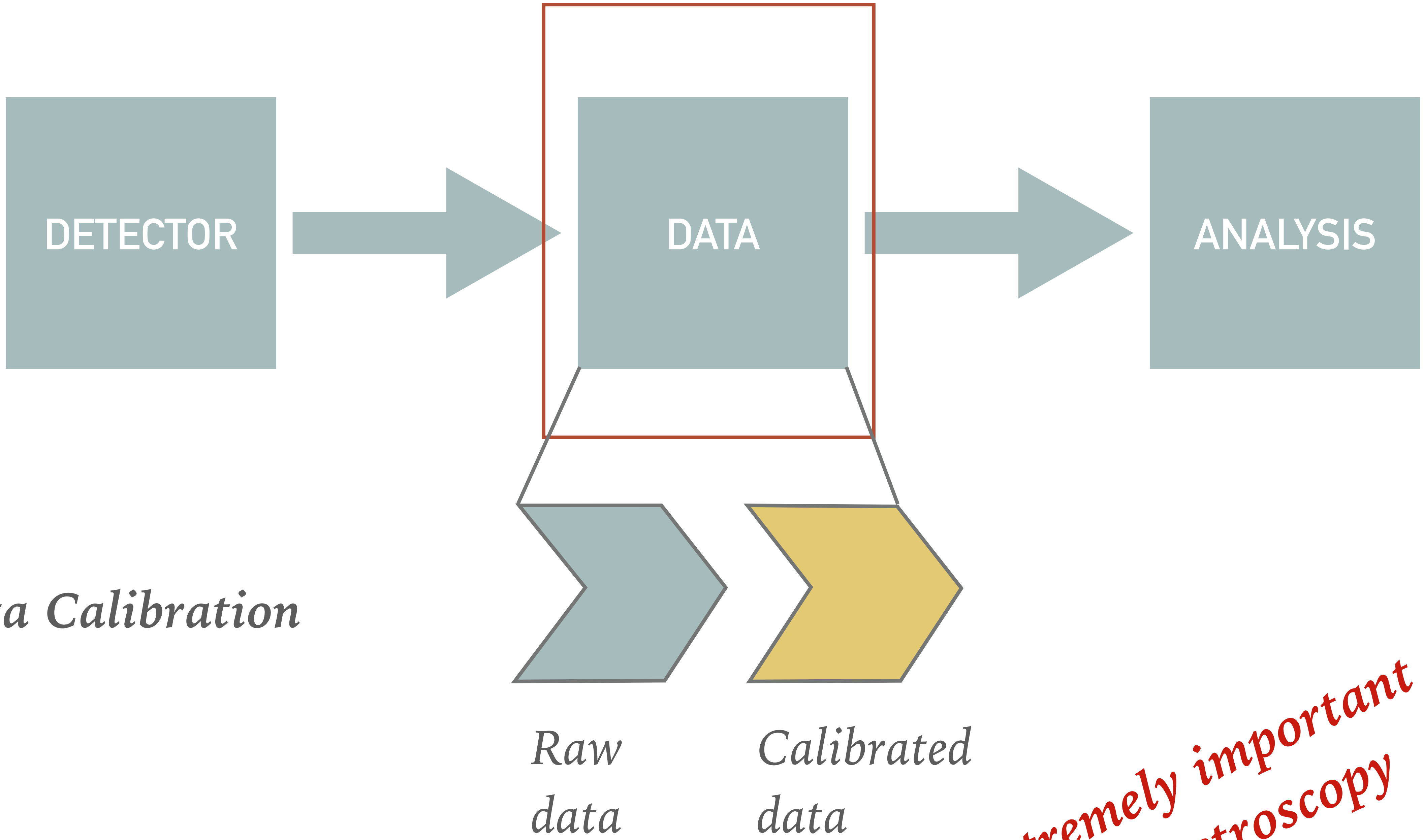
Cut to define anomalies





WIP - courtesy of S.Manti

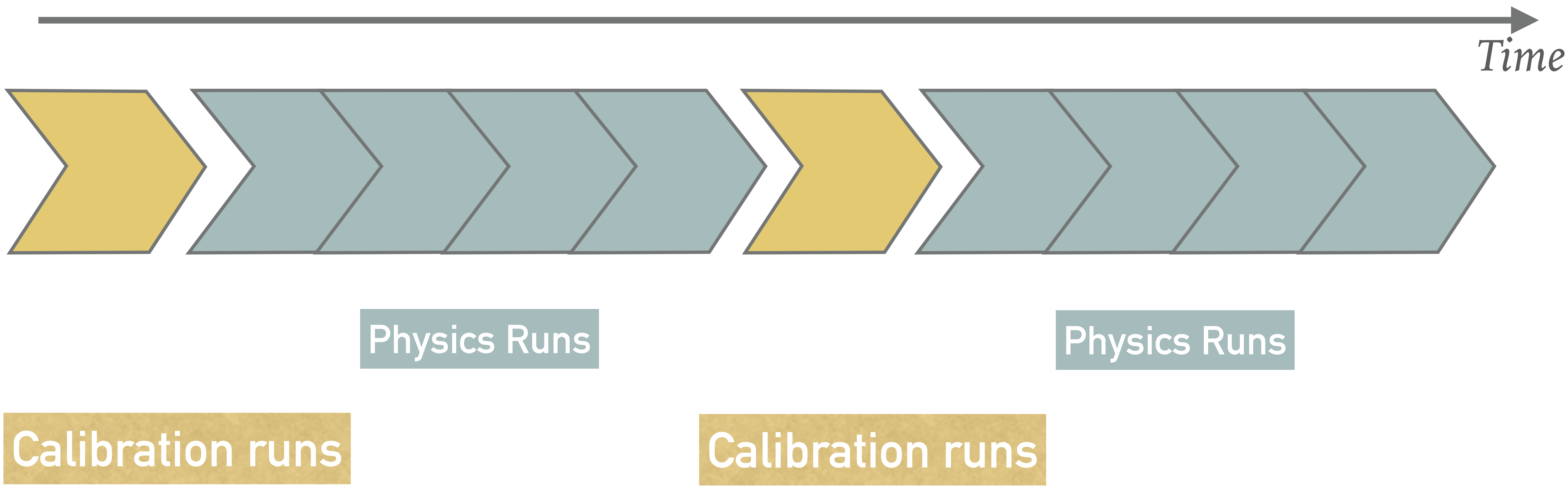
Machine Learning in physics experiment



Data Calibration

Extremely important in x-ray spectroscopy

Typical calibration strategy

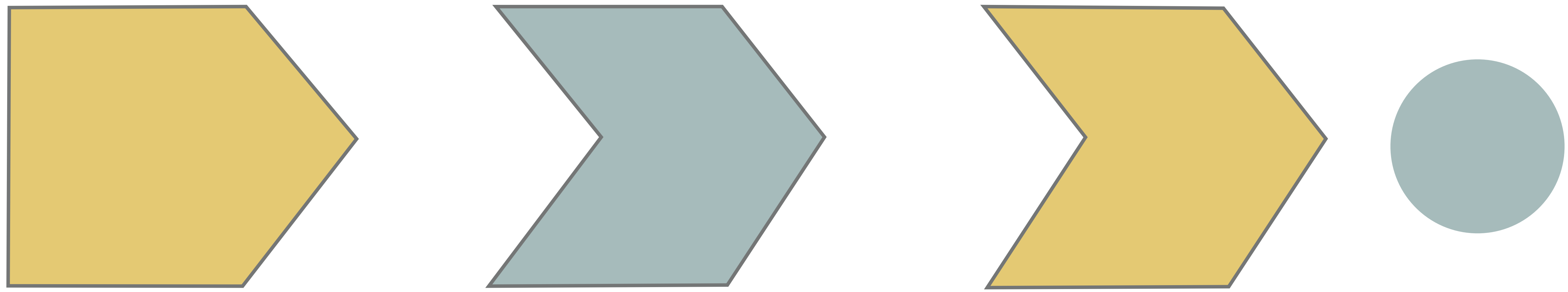


Need to balance - trade off!

Want more calibration points
but also more physics data

Enhance calibrations to the limit with differentiable programming

Enhance calibrations to the limit with differentiable programming



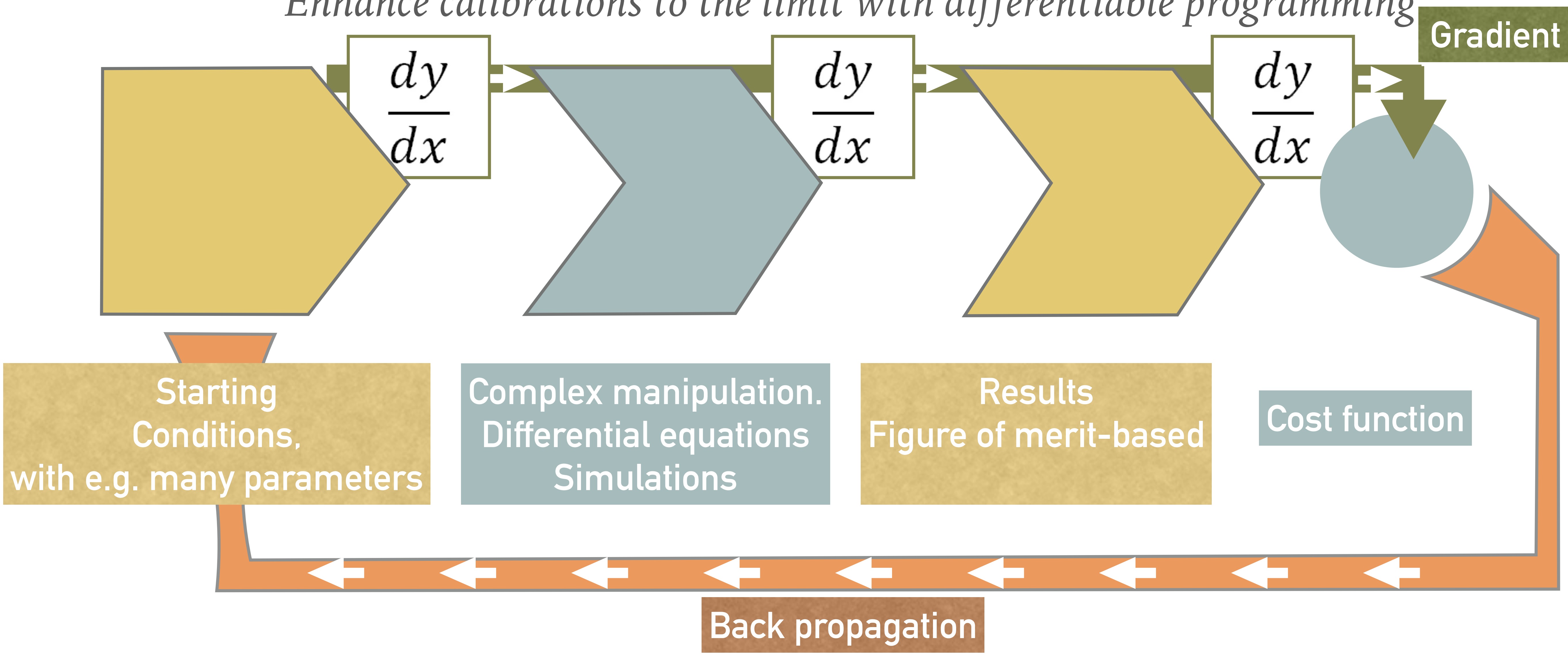
Starting
Conditions,
with e.g. many parameters

Complex manipulation.
Differential equations
Simulations

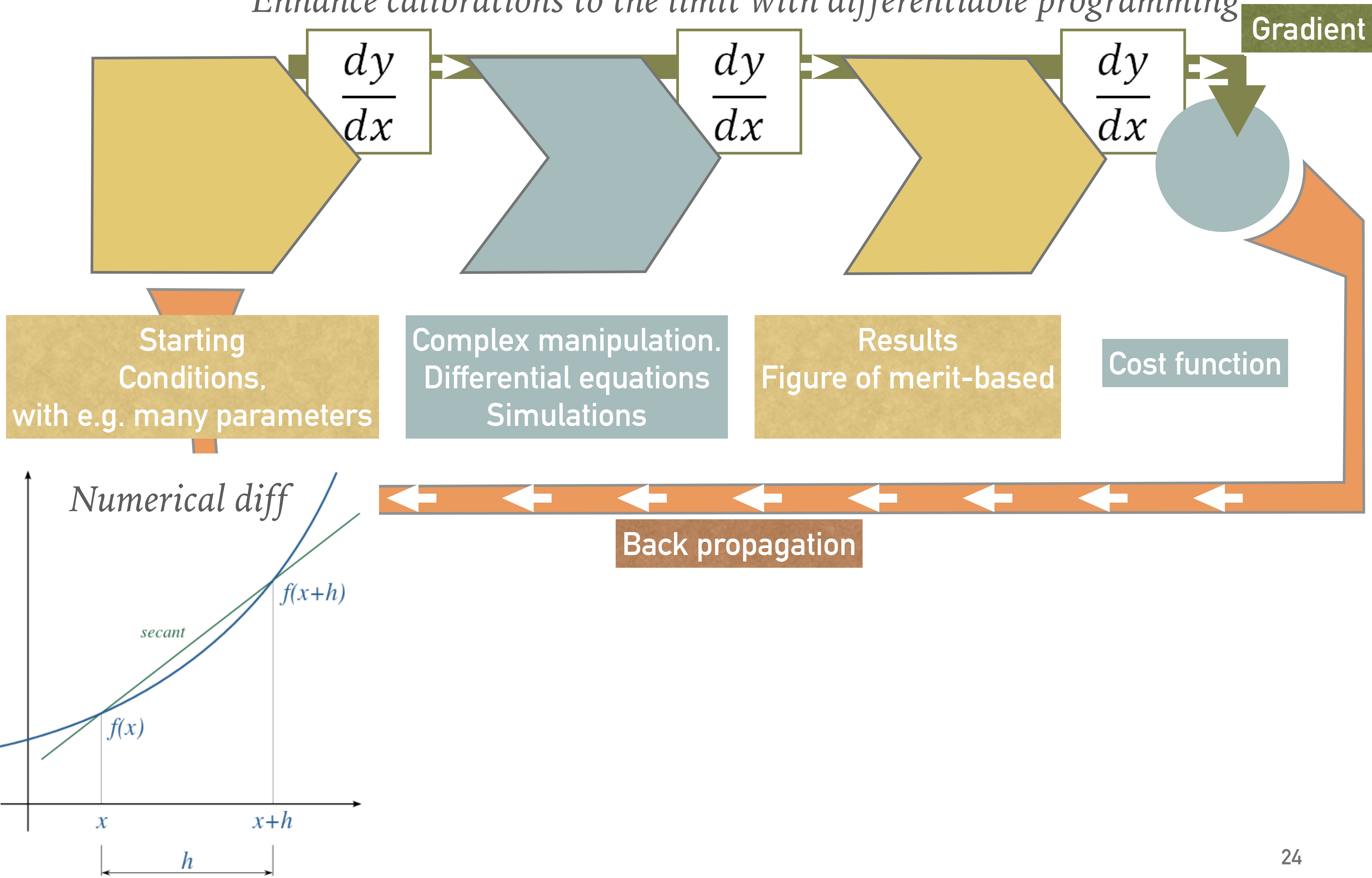
Results
Figure of merit-based

Cost function

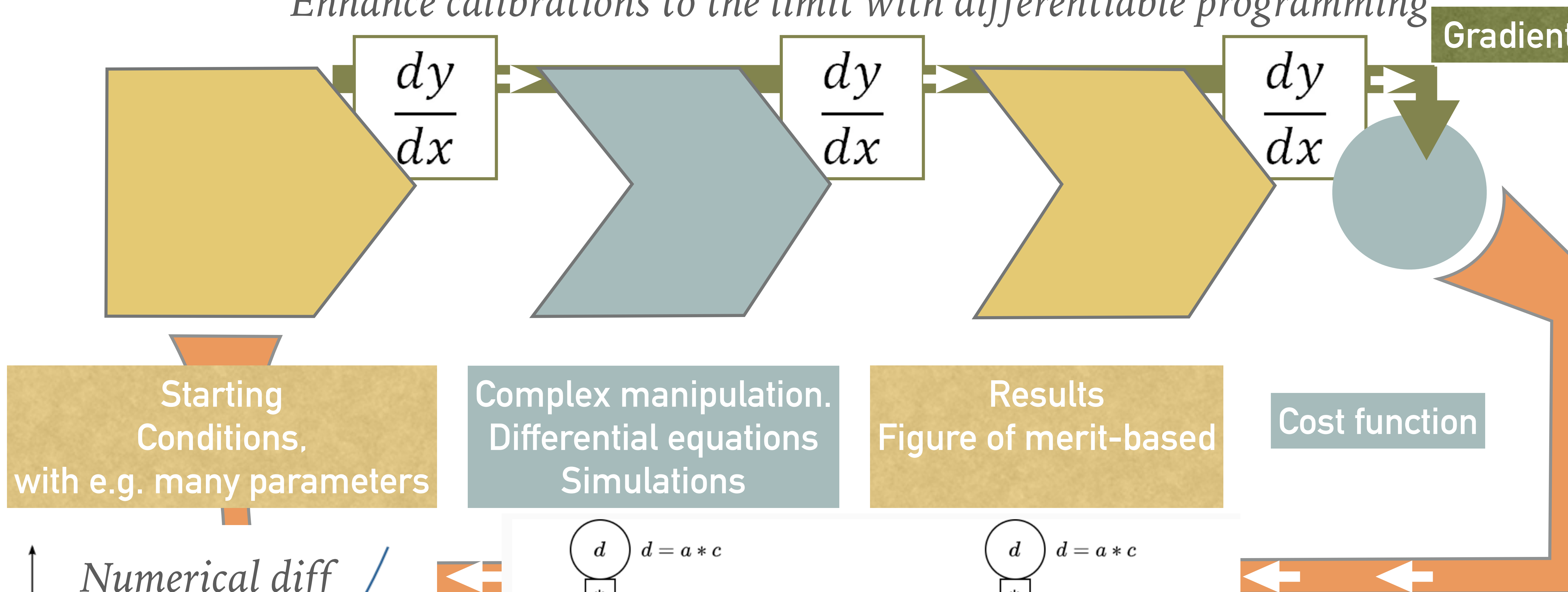
Enhance calibrations to the limit with differentiable programming

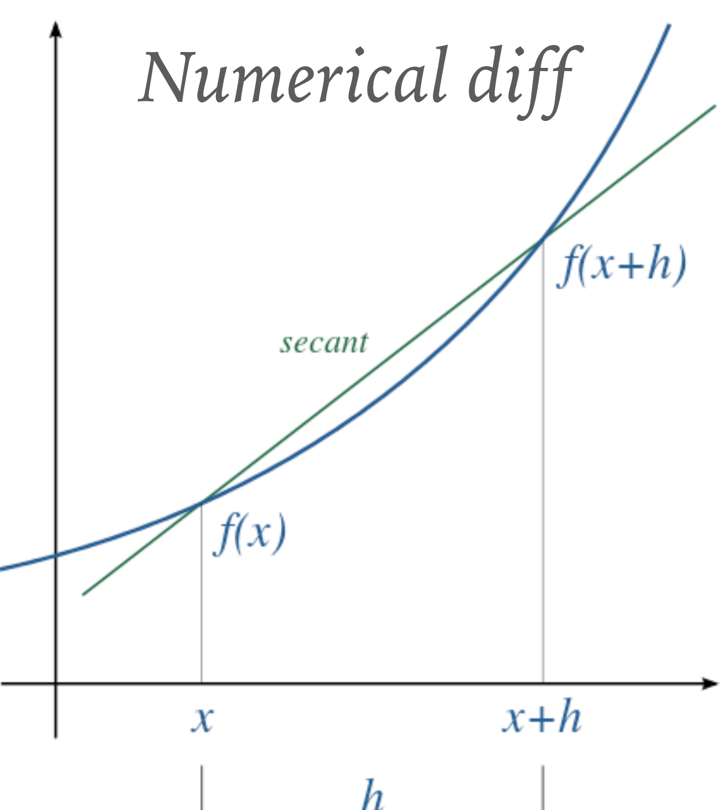


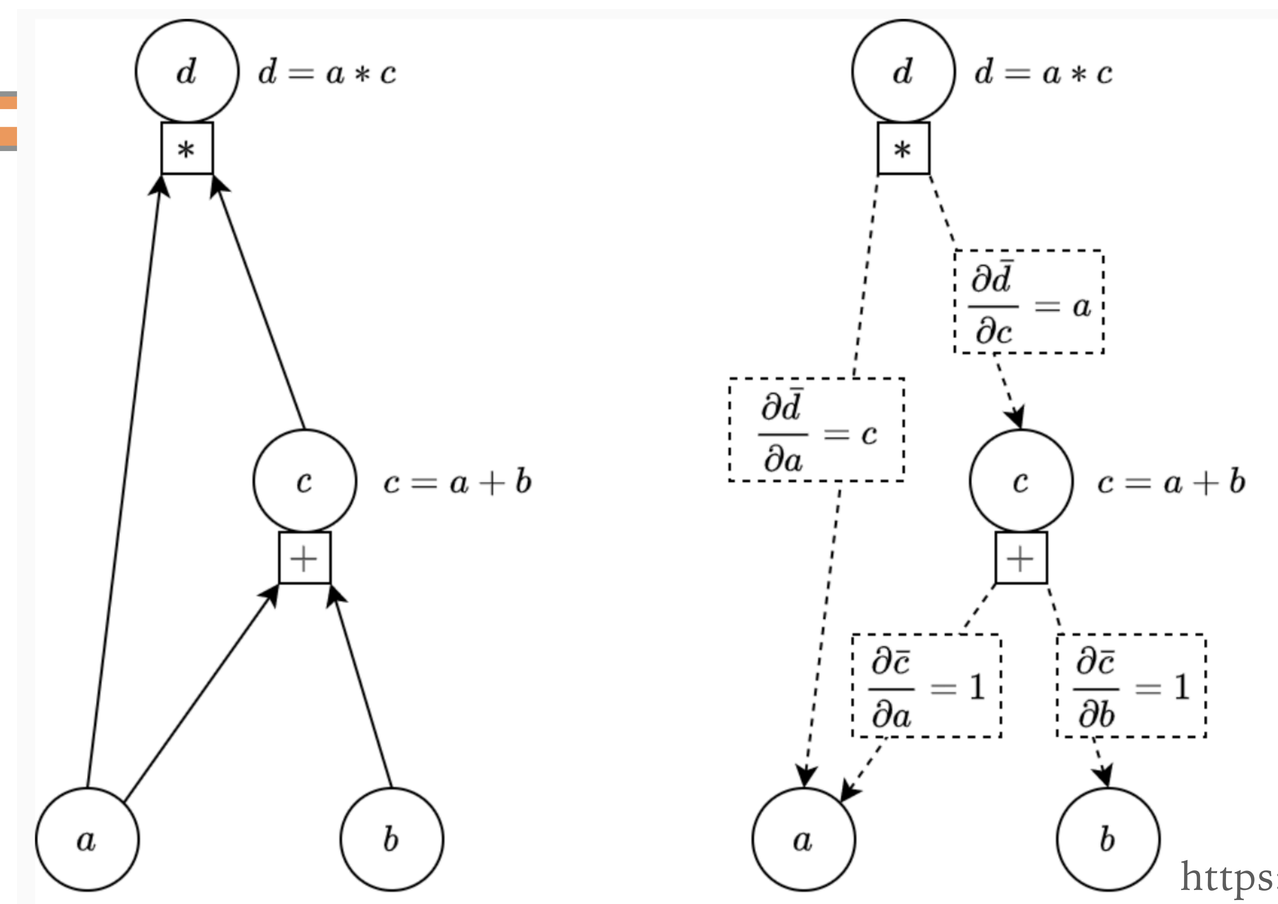
Enhance calibrations to the limit with differentiable programming



Enhance calibrations to the limit with differentiable programming







$$\frac{\partial d}{\partial a} = \frac{\partial \bar{d}}{\partial a} + \frac{\partial \bar{d}}{\partial c} * \frac{\partial \bar{c}}{\partial a}$$

$$\frac{\partial d}{\partial a} = c + a * 1$$

$$\frac{\partial d}{\partial a} = a + b + a$$

$$\frac{\partial d}{\partial a} = 2a + b$$

https://sidsite.com/posts/autodiff/

Enhance calibrations to the limit with differentiable programming

Target and environment variables

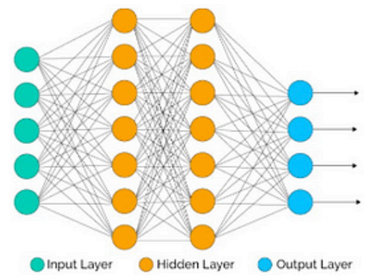
Control Parameters

Loss

Neural Network

ODE Solver

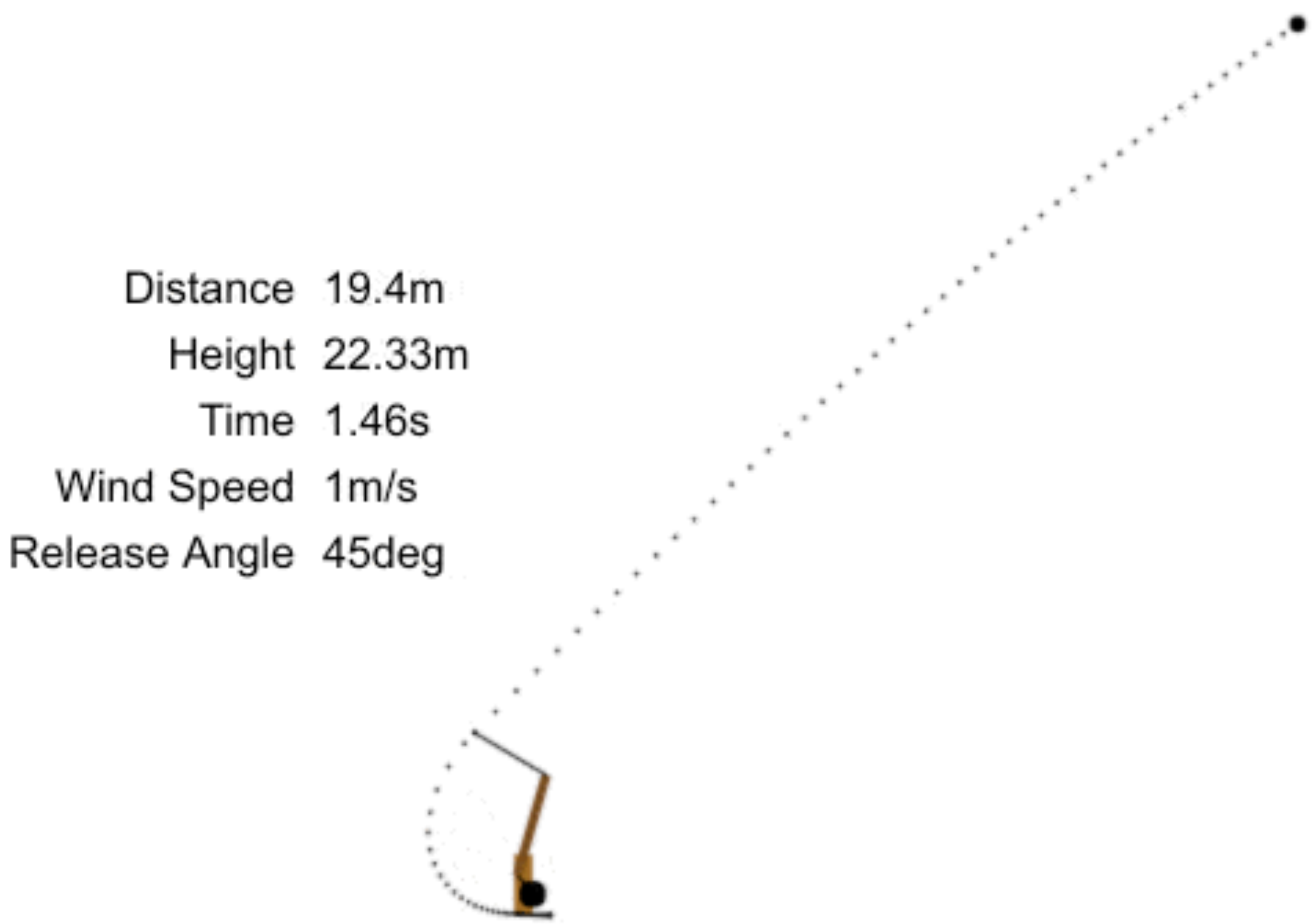
wind = -10m/s
target = 50m



angle = 25°
weight = 200kg

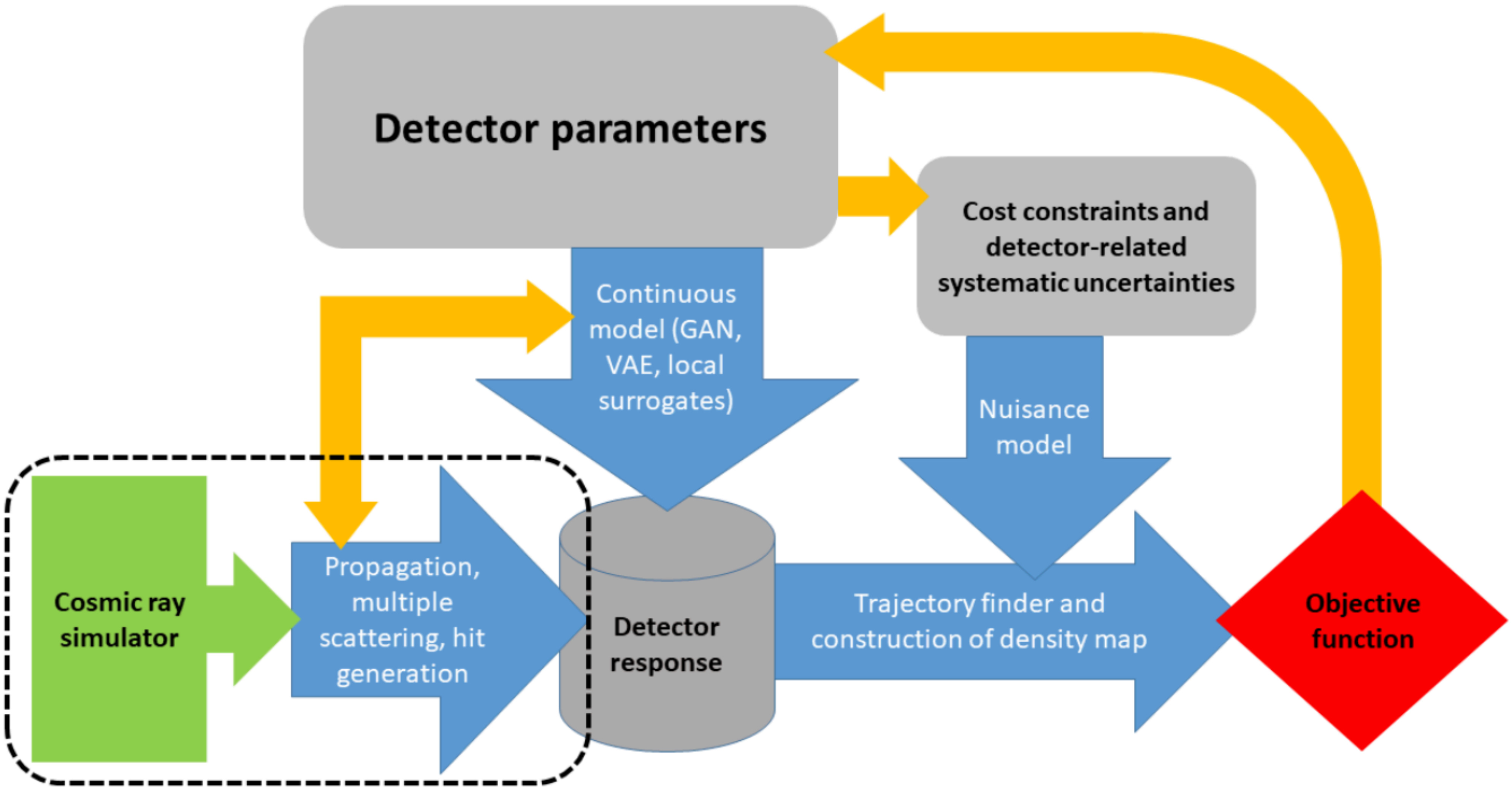
(target_distance – actual_distance)²

Backpropagation



https://github.com/JuliaComputing/ODSC2019

Enhance calibrations to the limit with differentiable programming

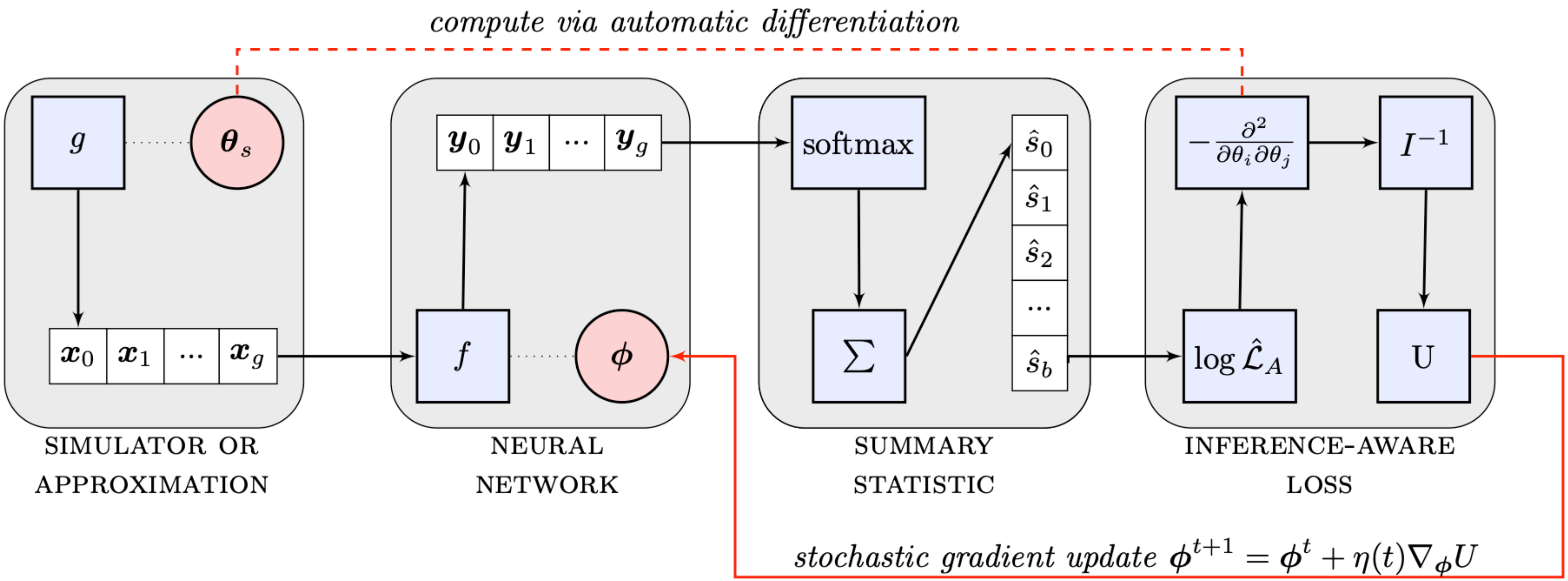


Machine-learning Optimized Design of Experiments

Optimization of detector design and operation

https://inspirehep.net/files/8f6784eac6a7f06c184cfc50757de656

Enhance calibrations to the limit with differentiable programming

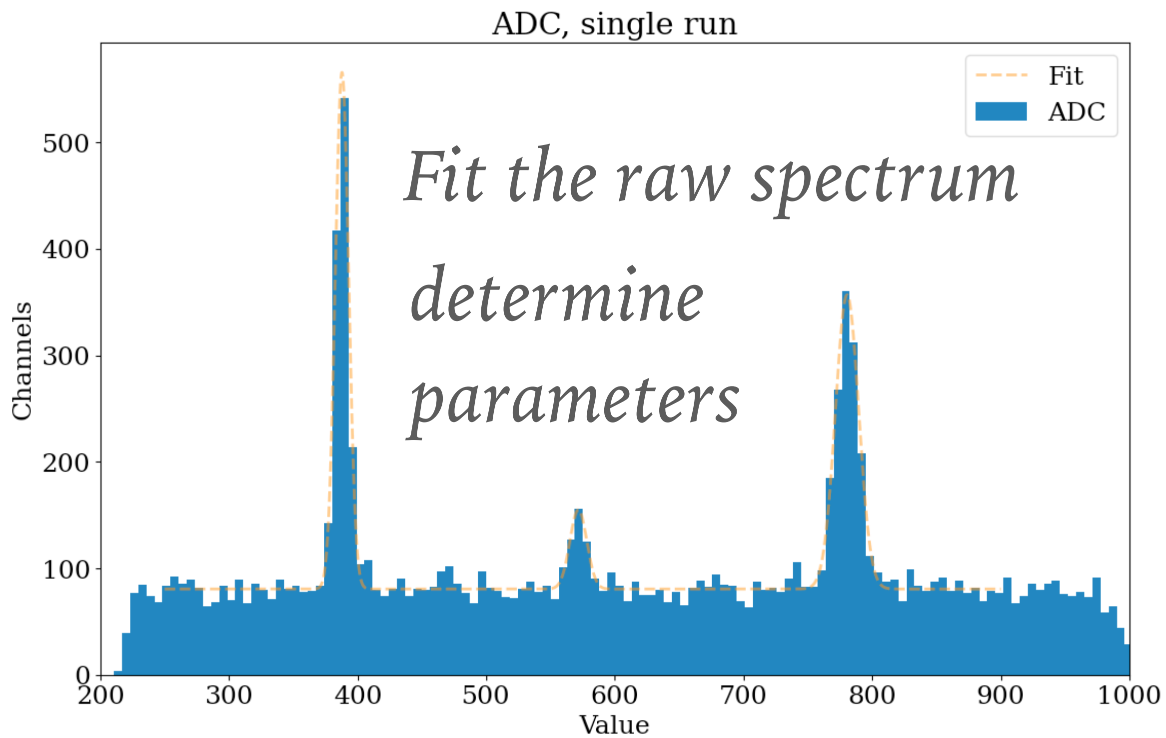


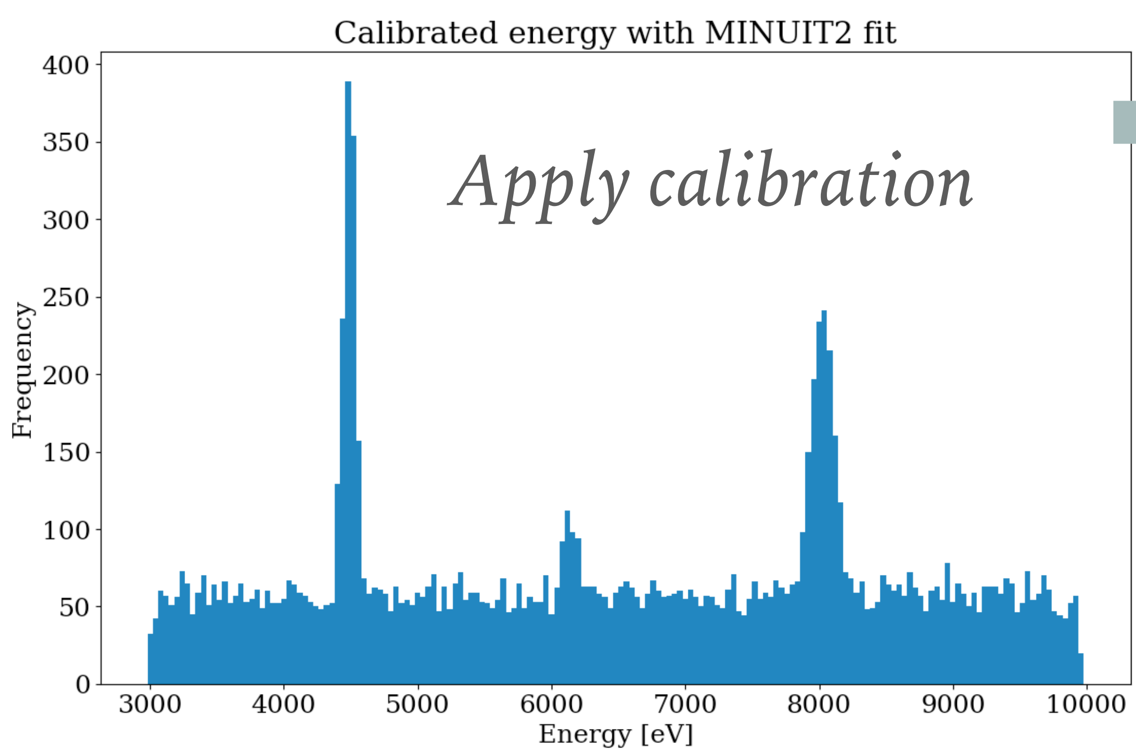
Sketch of the INFERNO algorithm. Batches from a simulator are passed through a neural network and a differentiable summary statistic is constructed that allows to calculate the variance of the POI. The parameters of the network are then updated by stochastic gradient descent.

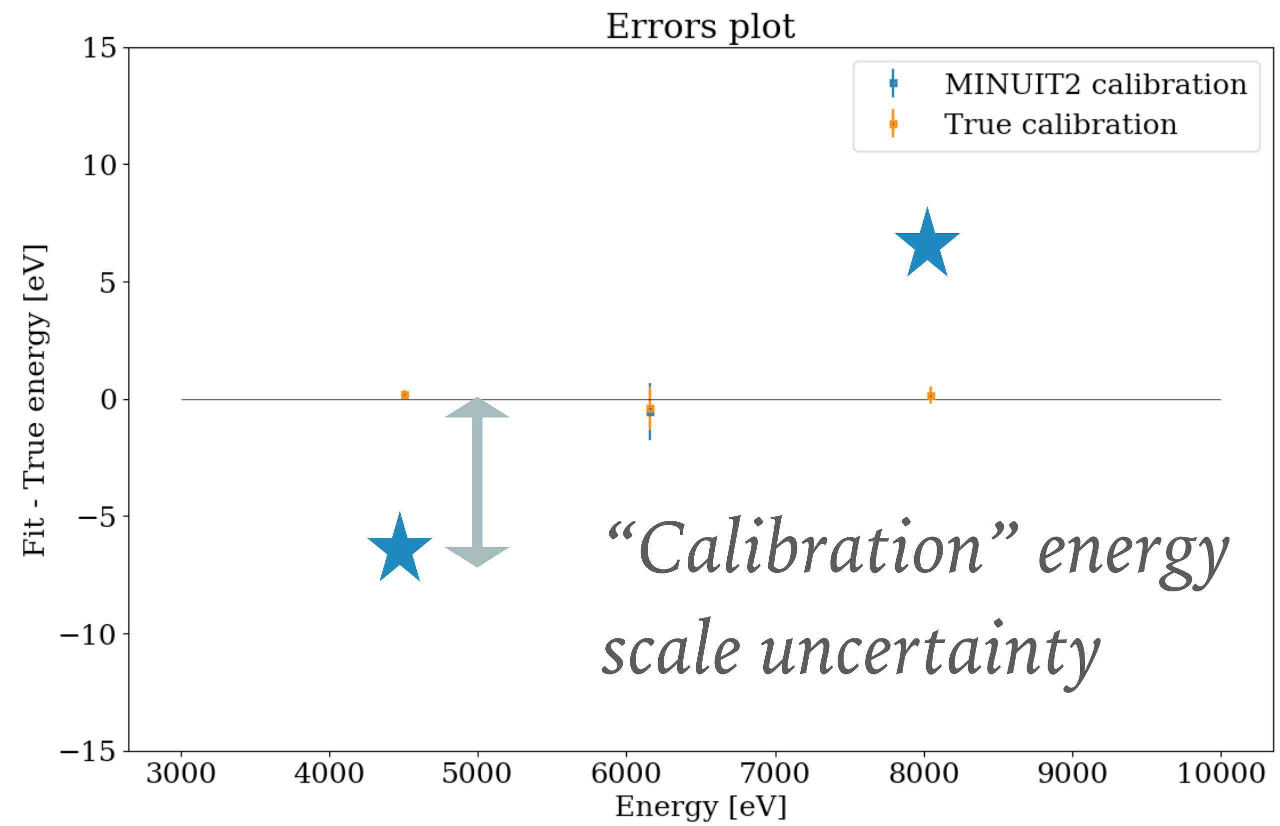
https://cds.cern.ch/record/2807001/plots

A test optimisation setup

The "Standard Approach"
to spectroscopic calibration







A test optimisation setup

The differentiable programming approach

Loss function for the gradient: loss

Derive calibrated dataset

$$\mathbf{E} = \mathscr{C}(\mathbf{ADC} \,|\, \mathbf{P})$$

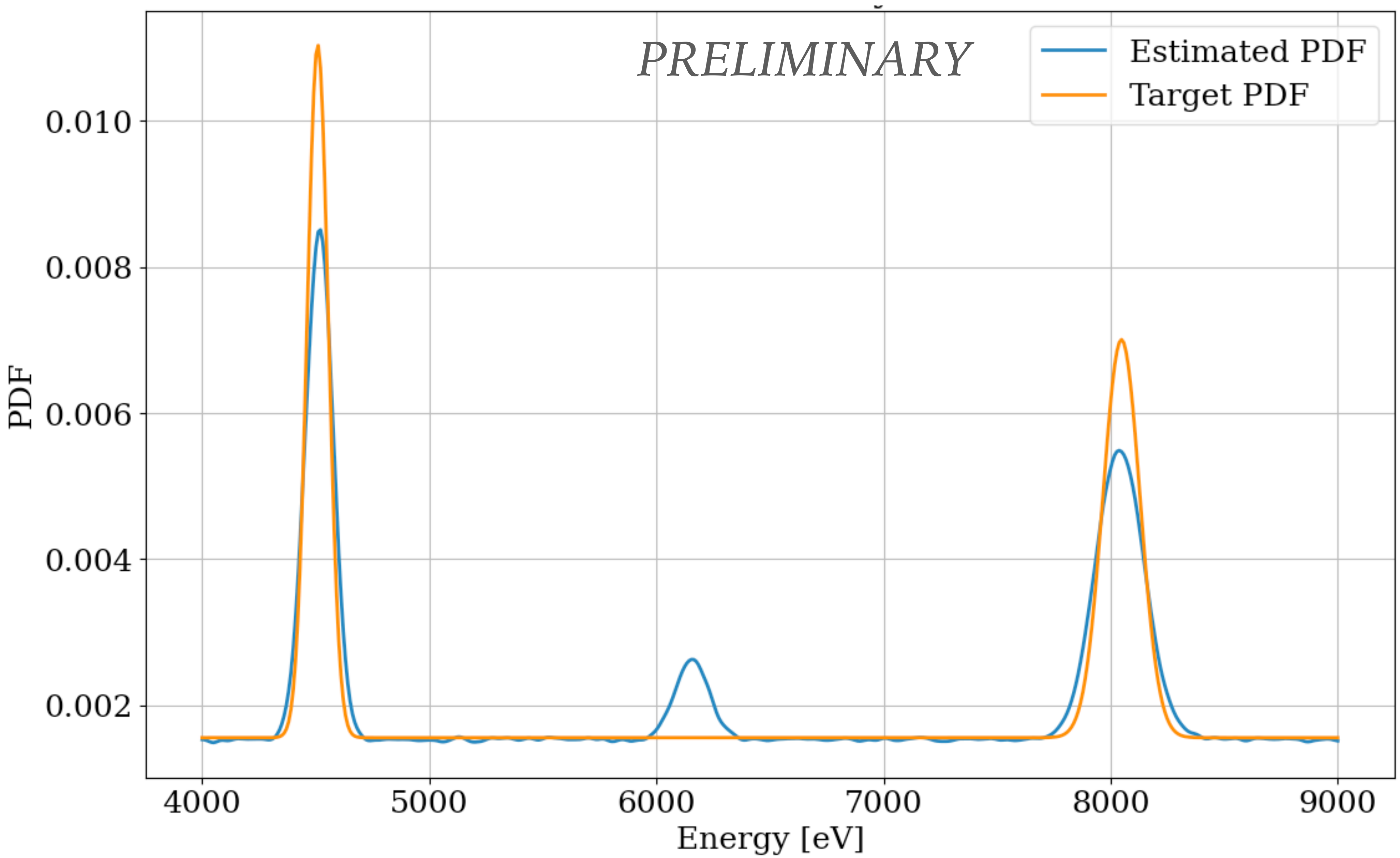
Compute the loss

$$\mathscr{L}(\mathbf{E})$$

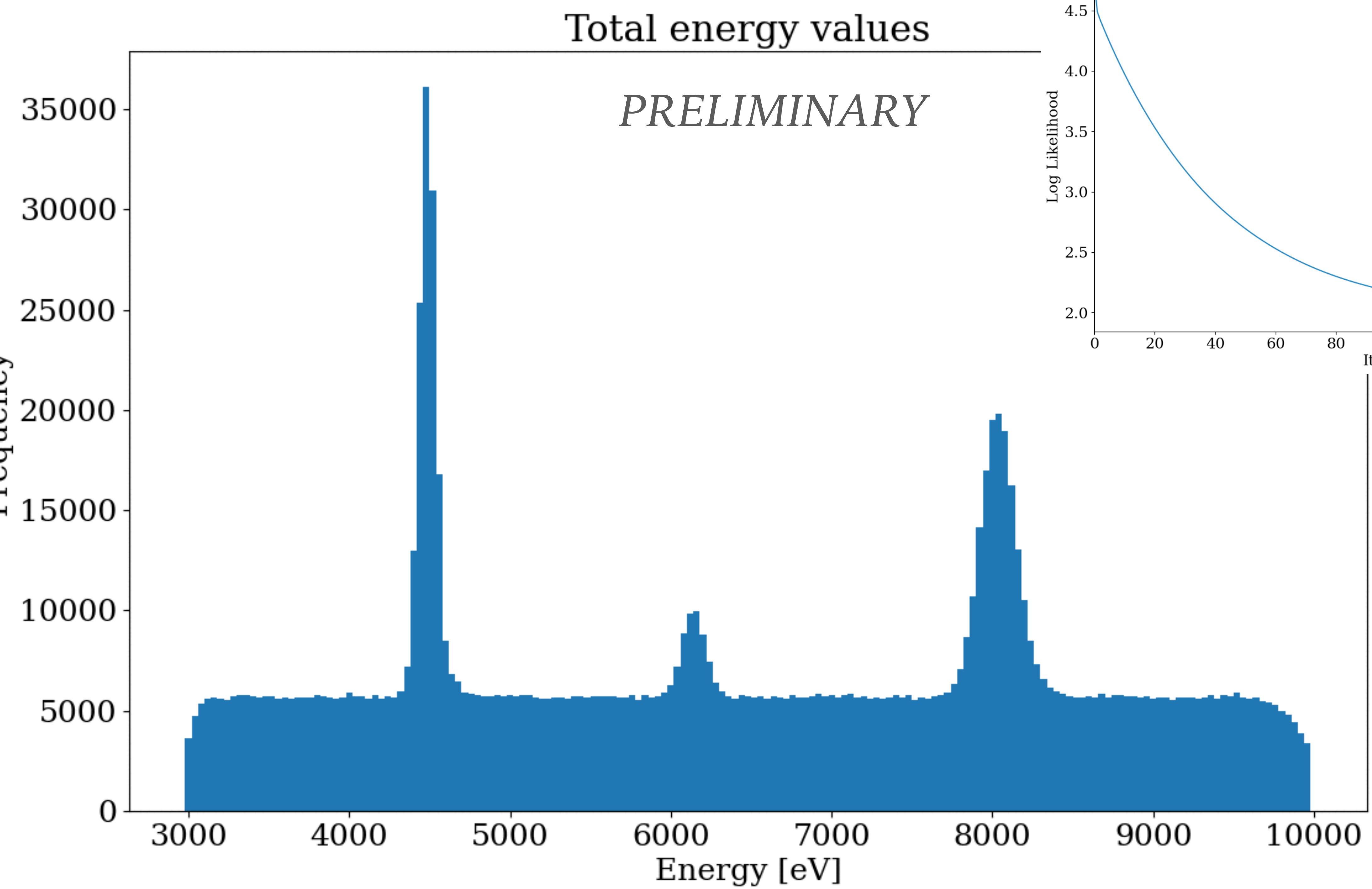
Update P to minimize the loss

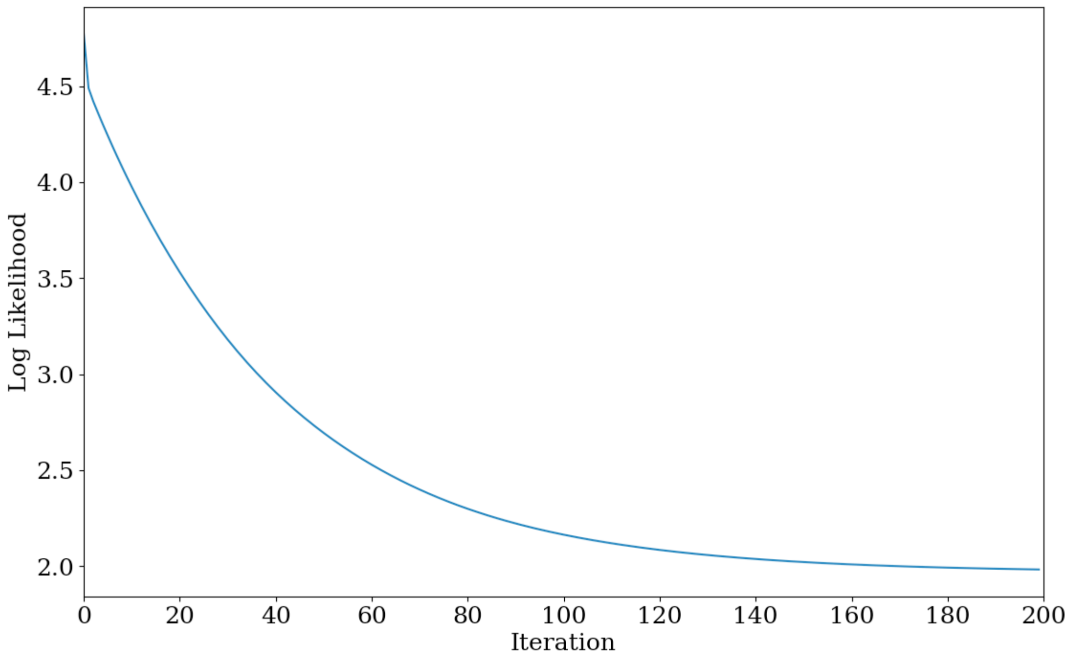
$$\mathbf{P} = \begin{bmatrix} p_{0,1} & p_{1,1} & p_{2,1} \\ \vdots & \vdots & \vdots \\ p_{0,i} & p_{1,i} & p_{2,i} \\ \vdots & \vdots & \vdots \\ p_{0,N} & p_{1,N} & p_{2,N} \end{bmatrix}$$

Take new calibration constants

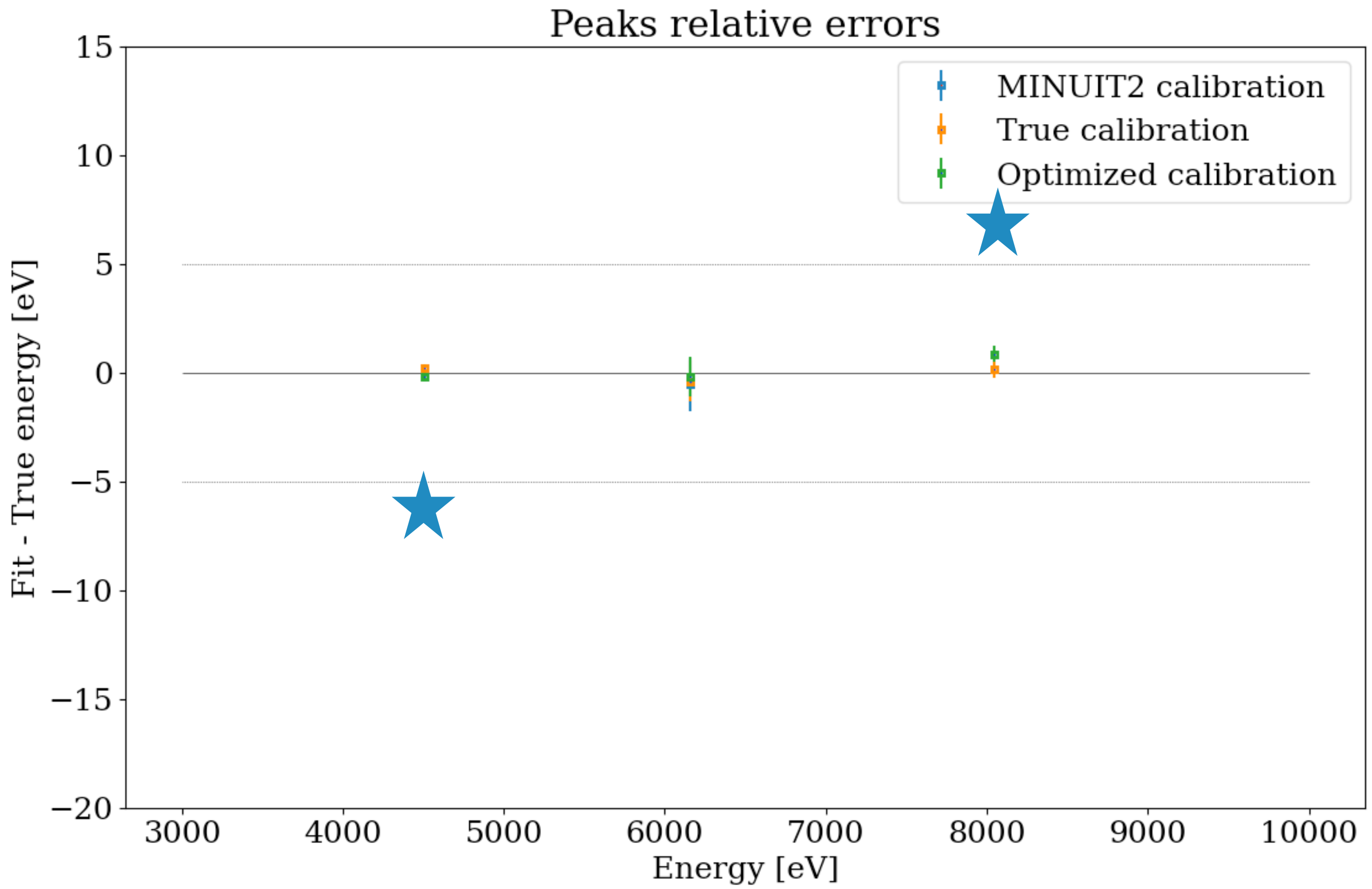


Distance to target PDF estimated via KDE approximation


Total energy values
PRELIMINARY
35000
30000
25000
20000
15000
10000
5000
0
3000
4000
5000
6000
7000
8000
9000
10000
Energy [eV]

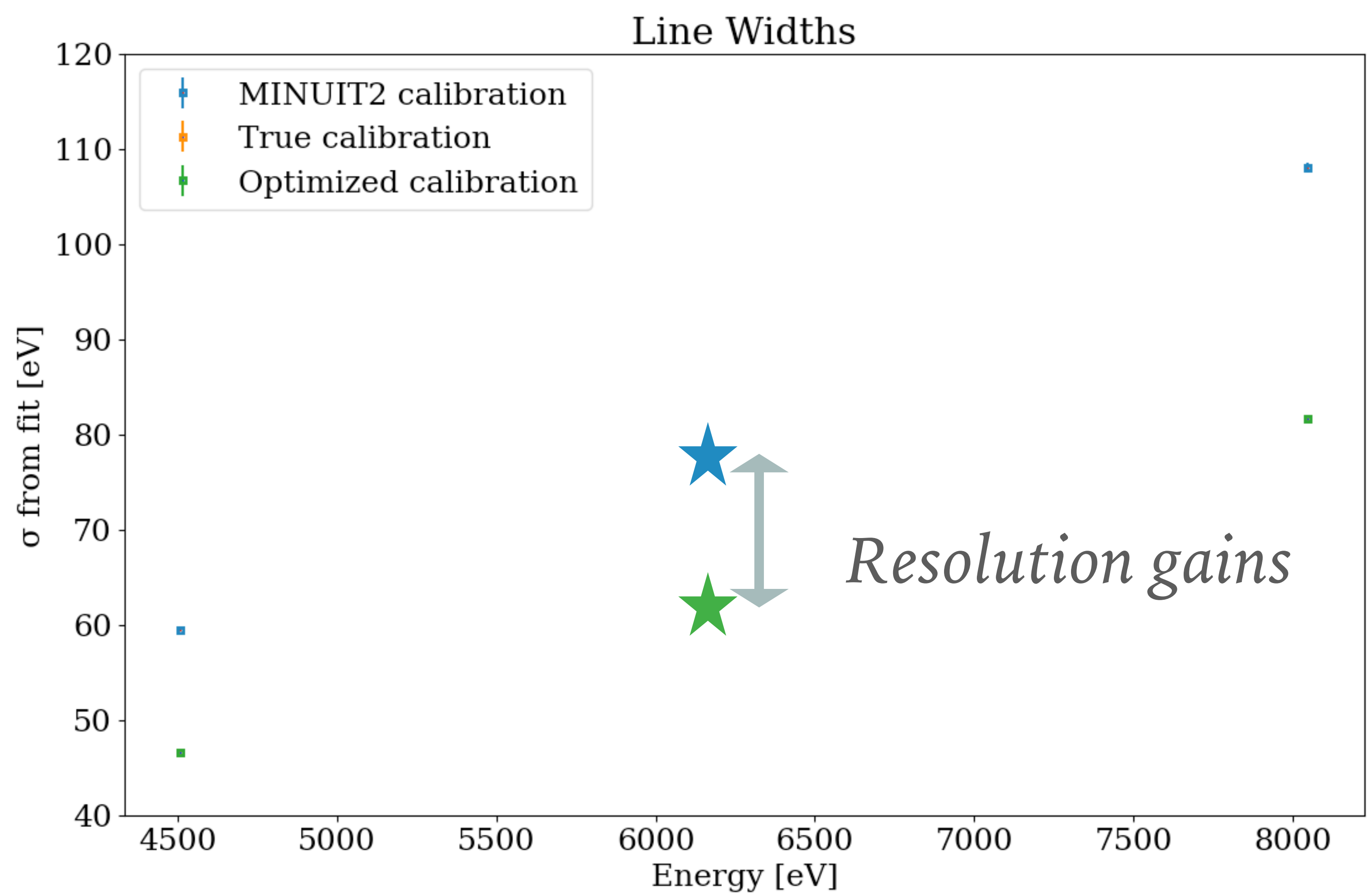

Log Likelihood
4.5
4.0
3.5
3.0
2.5
2.0
0
20
40
60
80
100
120
140
160
180
200
Iteration

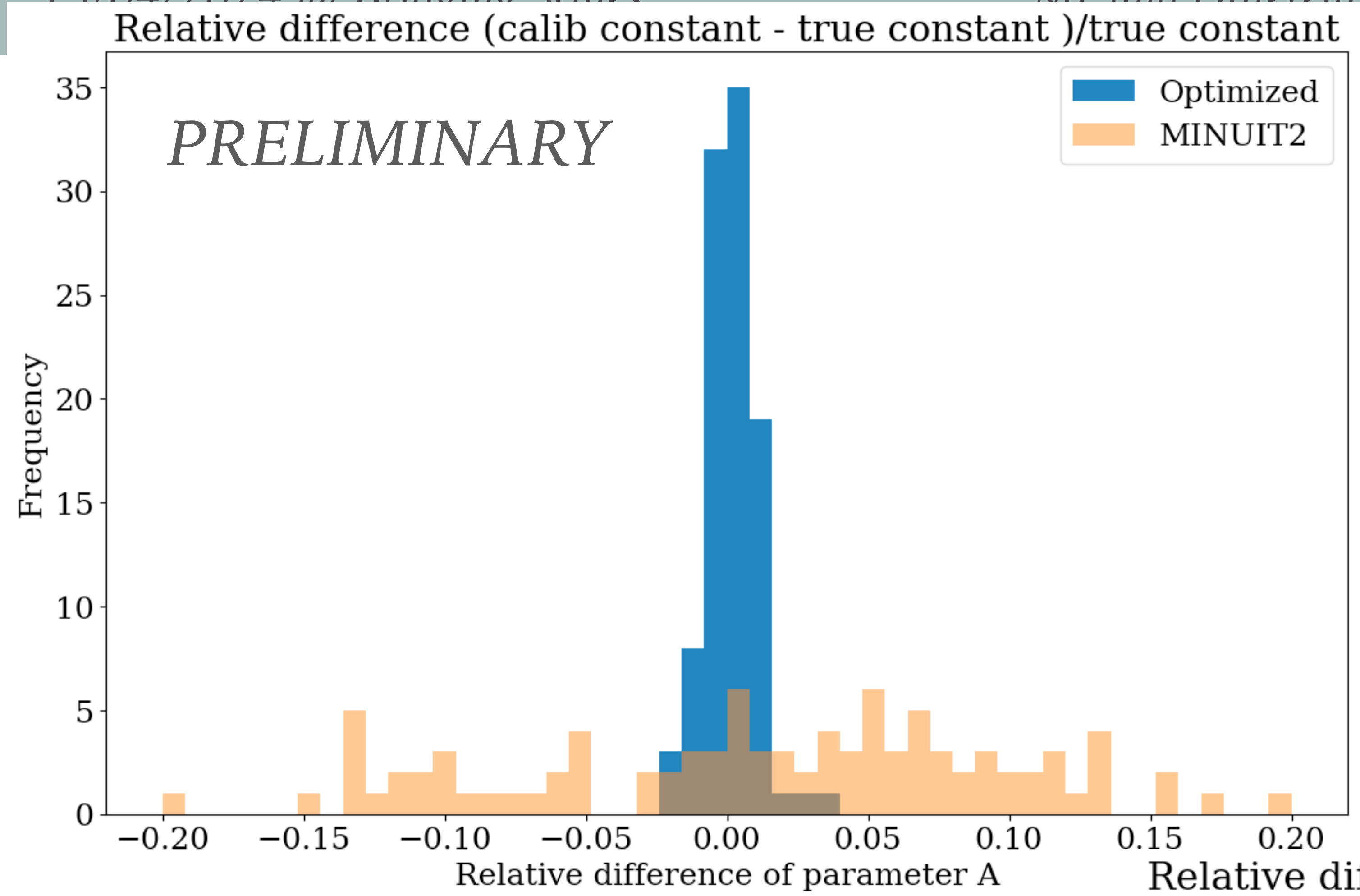
PRELIMINARY

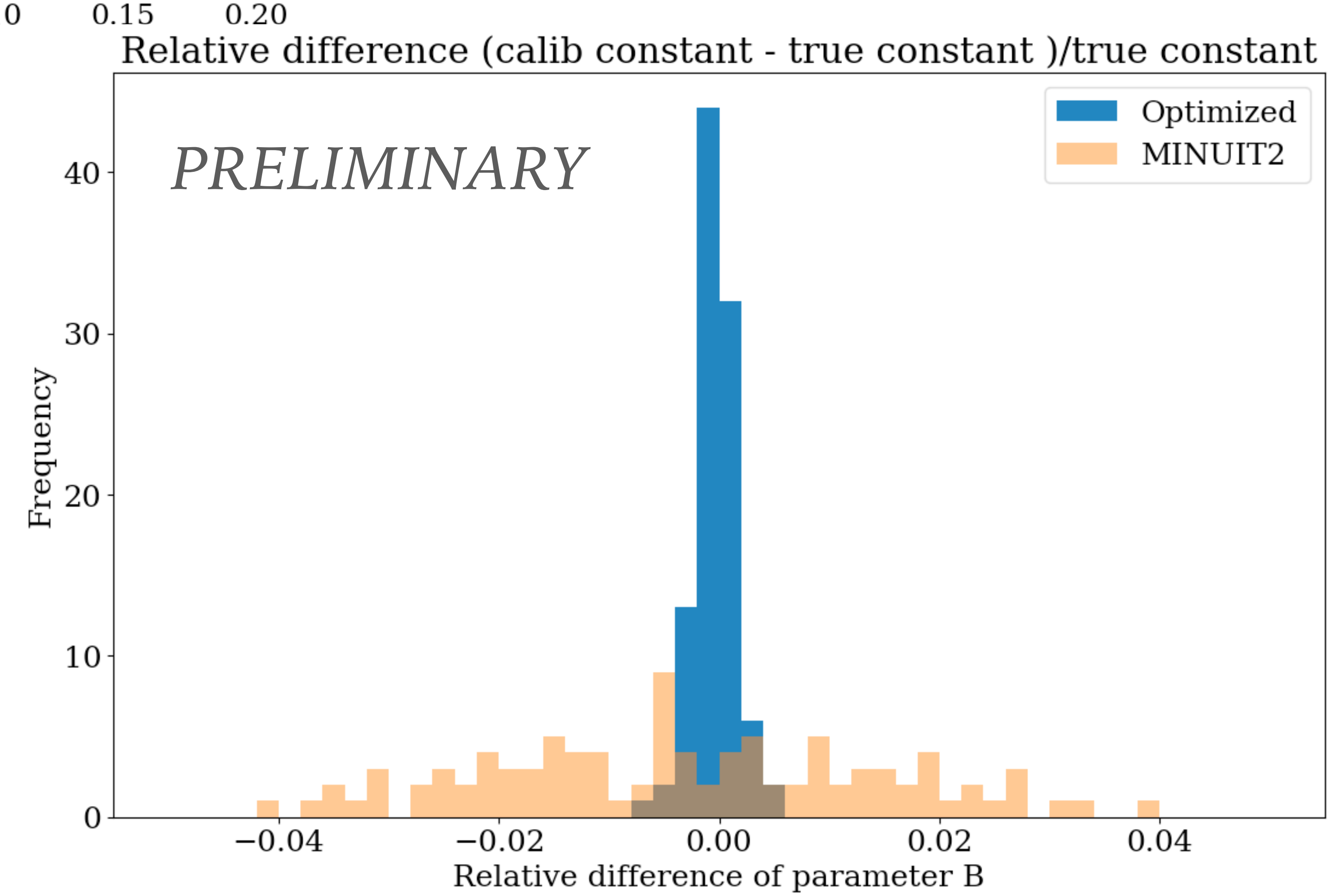


Minimize the uncertainty to the statistical level

Nice gains in spectroscopic peaks resolution

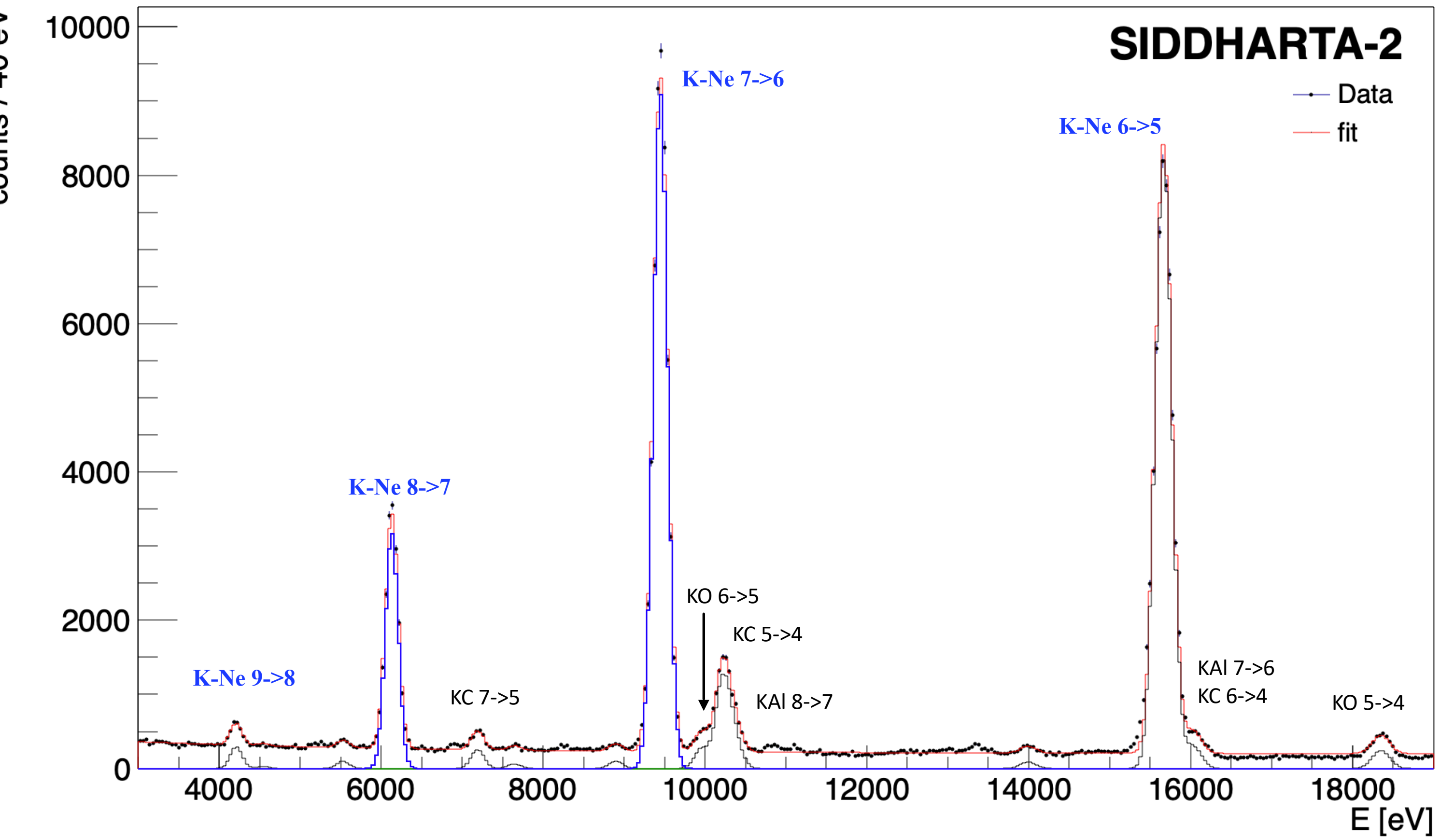


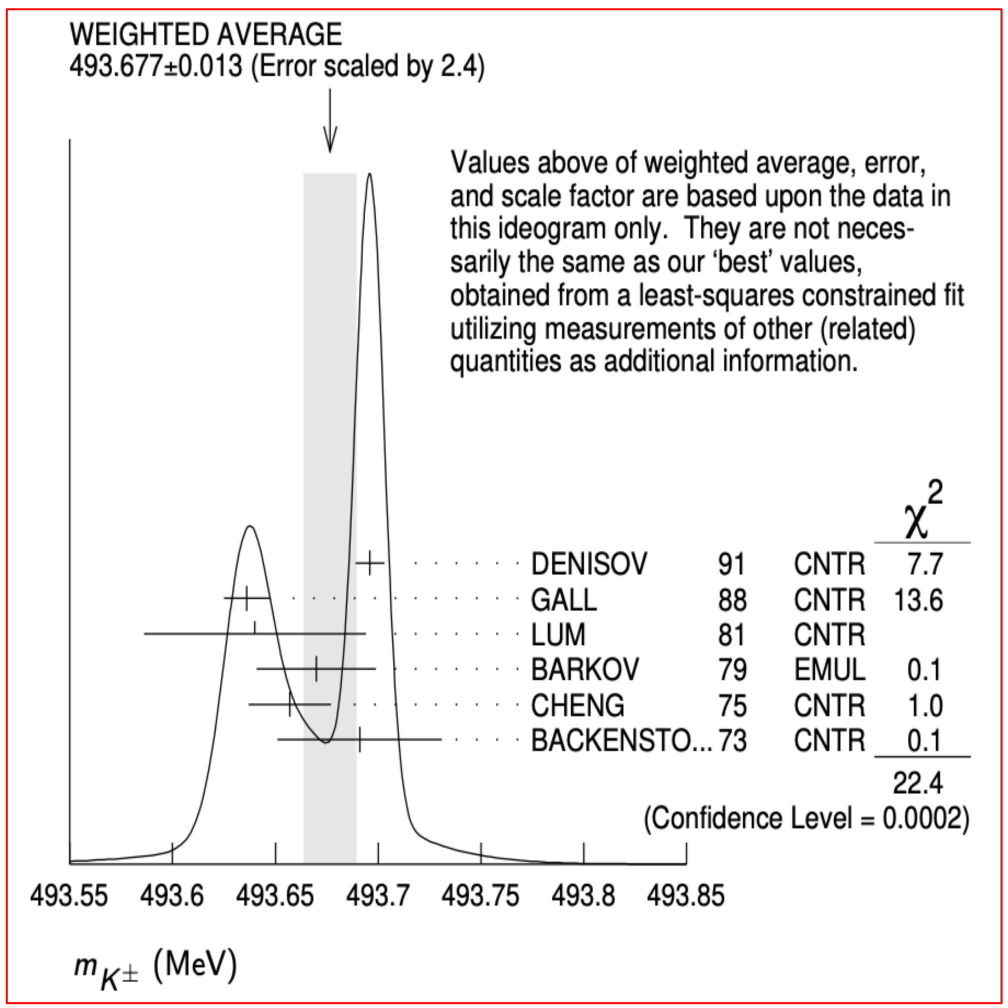

Relative difference (calib constant - true constant)/true constant
PRELIMINARY
Optimized
MINUIT2
Frequency
0
5
10
15
20
25
30
35
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
Relative difference of parameter A

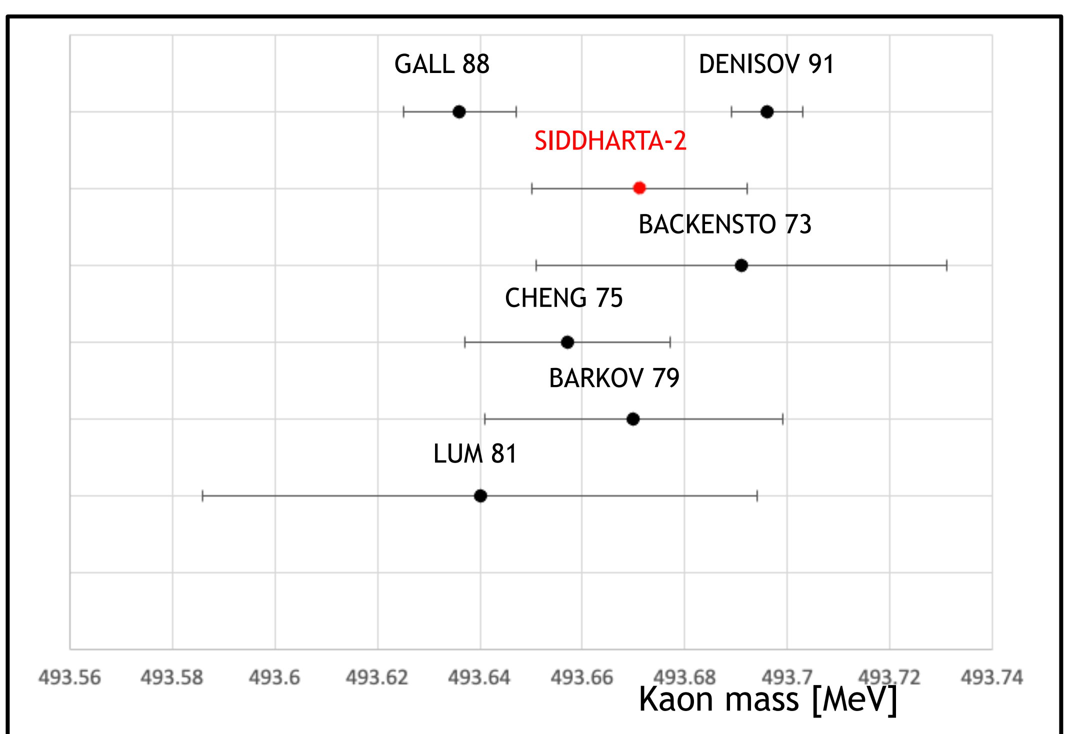

Relative difference (calib constant - true constant)/true constant
PRELIMINARY
Optimized
MINUIT2
Frequency
0
10
20
30
40
−0.04
−0.02
0.00
0.02
0.04
Relative difference of parameter B

Line	Energy [eV]
K-Ne 98	4206.35 3.75 (stat)
K-Ne 8	6130.86 0.71 (stat) 2 (sys.)
K-Ne 7	9450.08 0.41 (stat) 2 (sys.)
K-Ne 6	15673.30 0.52 (stat) 5 (sys.)

Outlook: kaon mass measurement @ SIDDHARTA-2







Conclusions

- *Calibration is a critical source of uncertainty in spectroscopy x-ray experiment*
- *Differentiable prog. approach has the potential to minimize*
  - *Can enhance existing data*
  - *Can calibrate “intermediately” or at each step*
- *In principle applicable to spectroscopic experiments*
- *Plan to test the method on real data for kaon mass meas*

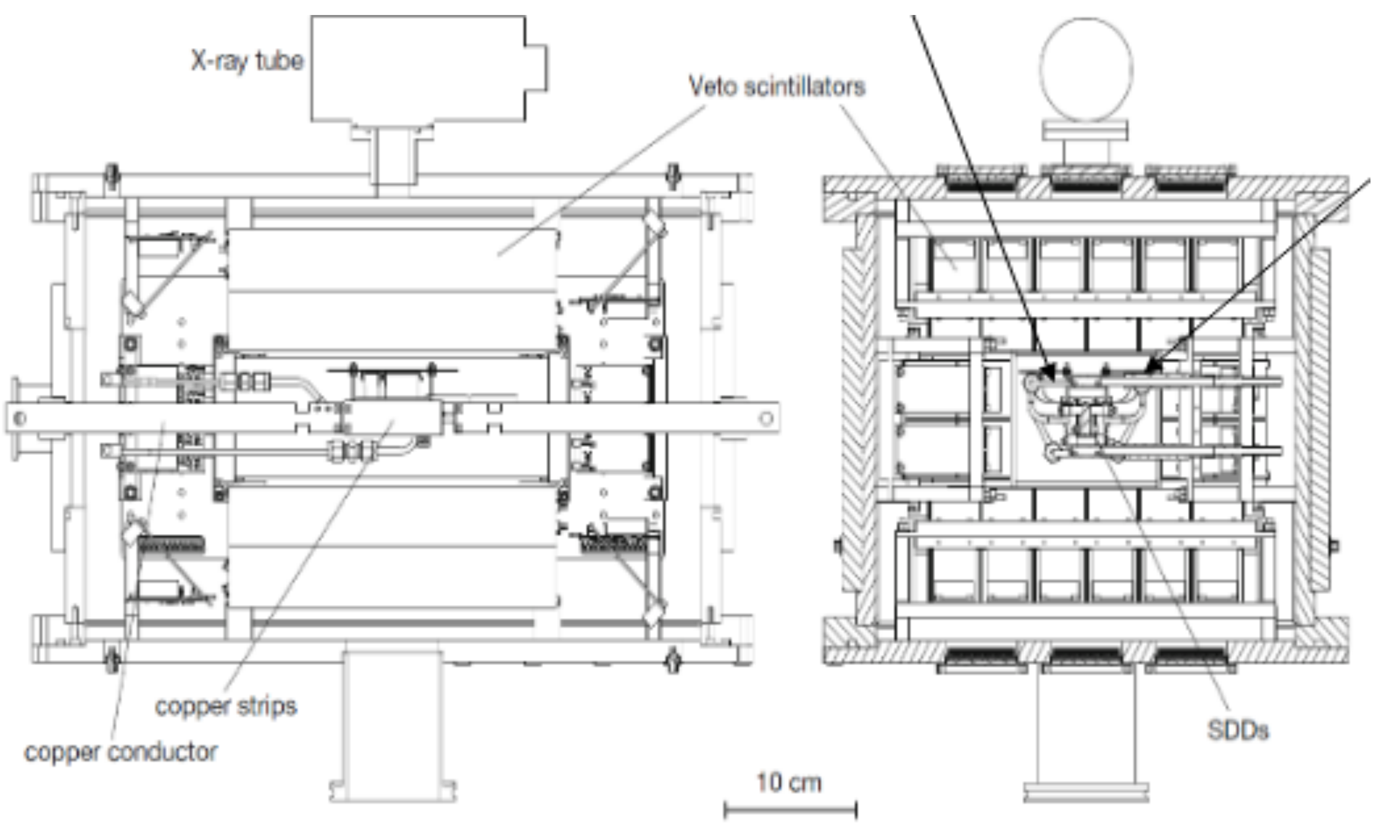
- *Method based on:* *https://iopscience.iop.org/article/10.1088/1361-6501/ad080a/meta*

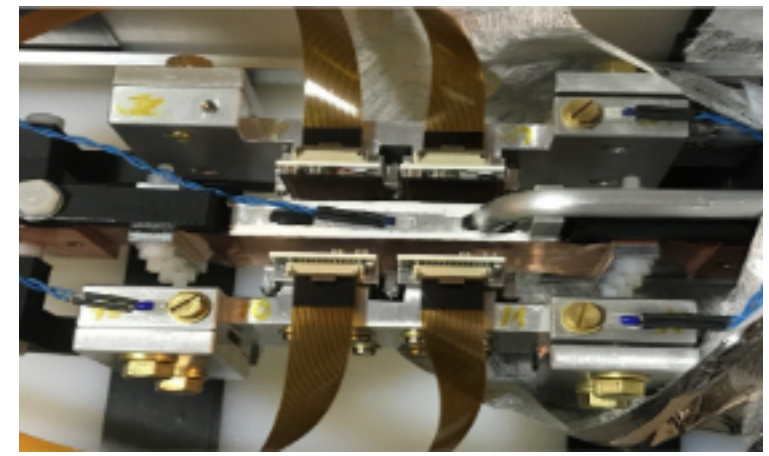
Thank you for your attention!

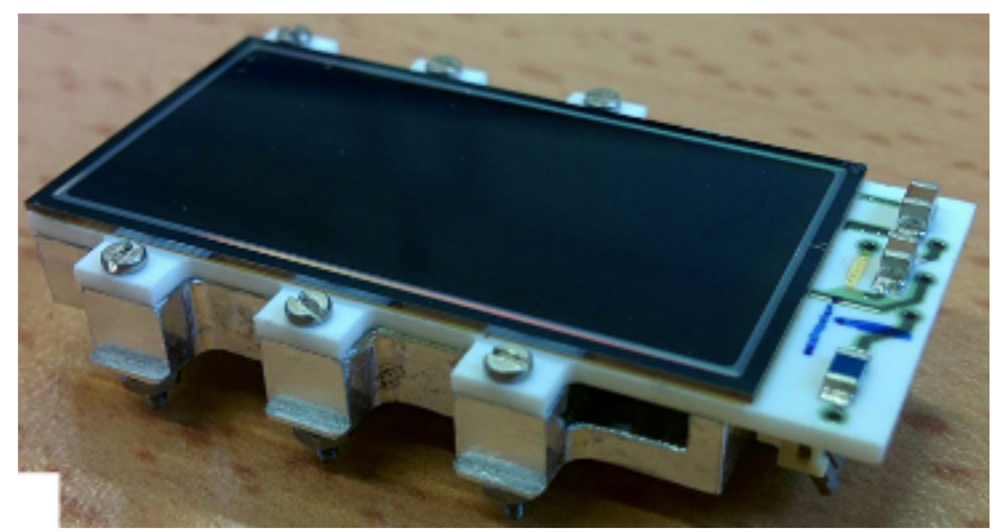
Questions?

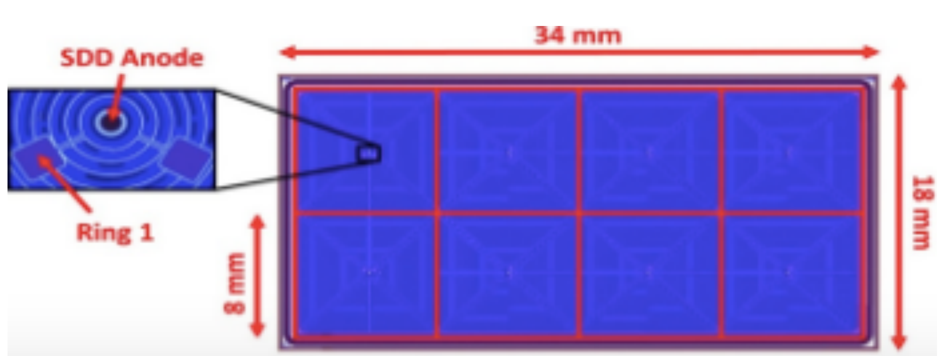
The VIP-2 Experiment

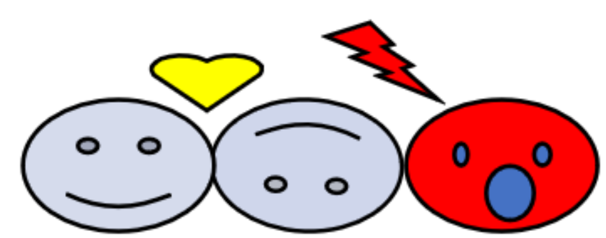
*Silicon Drift Detectors (**SDDs**) higher resolution (190 eV FWHM at 8.0 →keV), faster (triggerable) detectors. 4 arrays of 2 x 4 SDDs 8mm x 8mm each, liquid argon closed circuit cooling 170 °C*

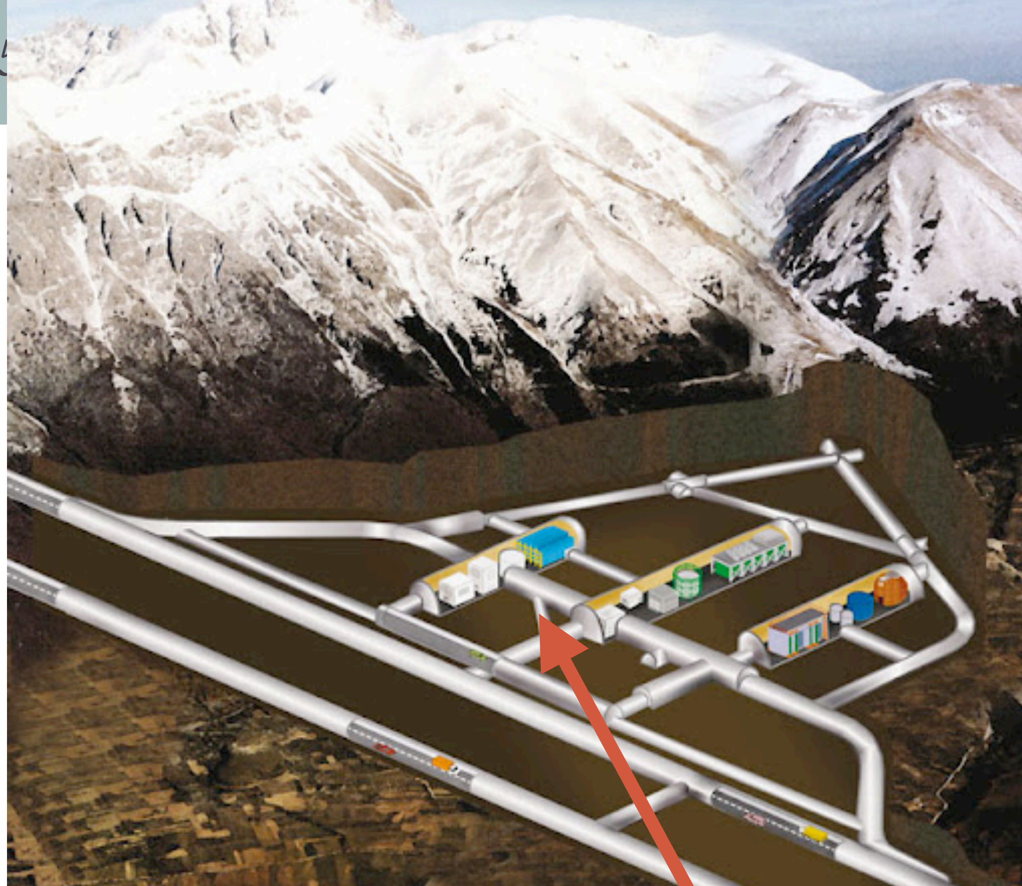






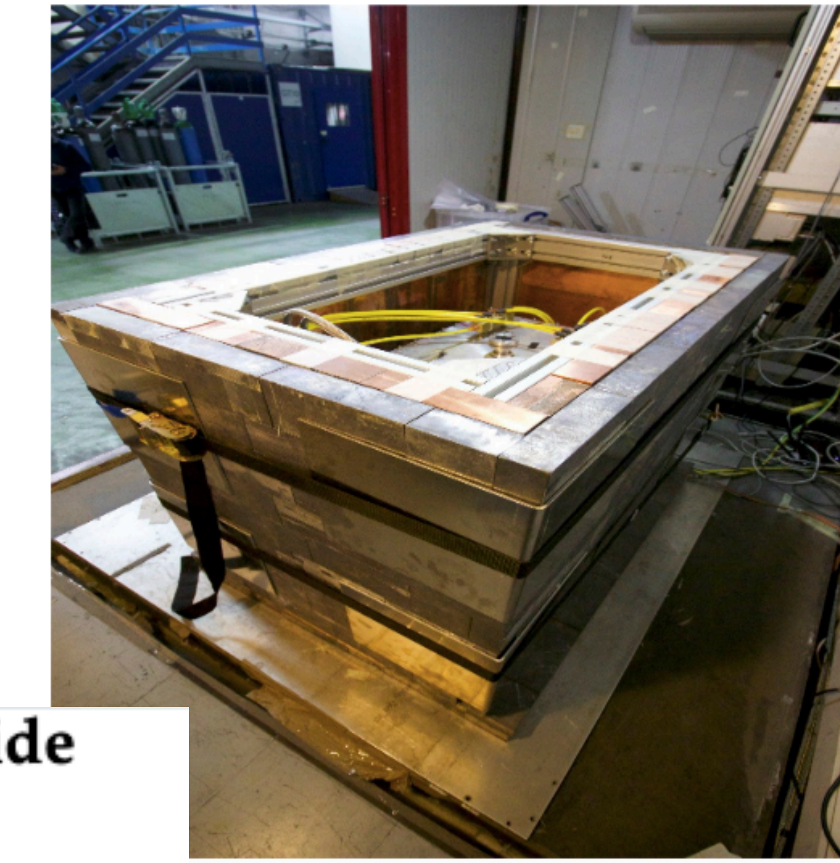




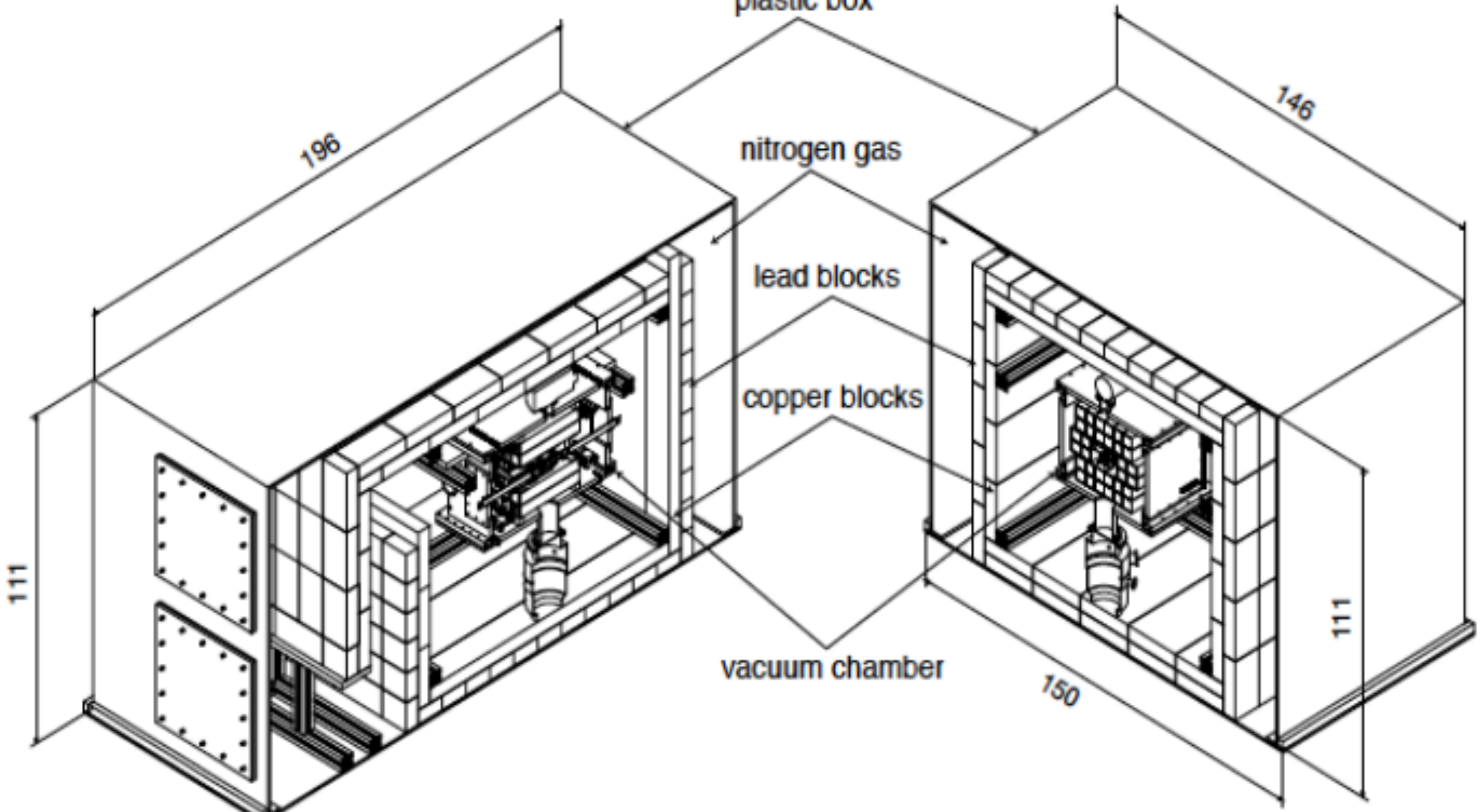


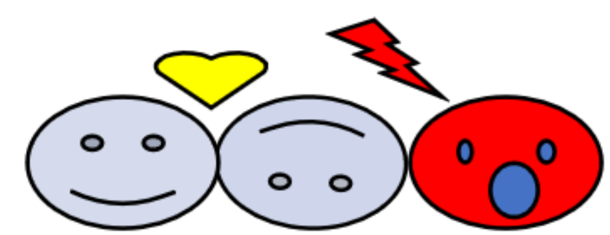
1400 m rock coverage

Upgrade concluded in April 2019:



Passive scielding → two layers, copper inside lead outside





Upgrade concluded in April 2019:

1400 m rock overburden

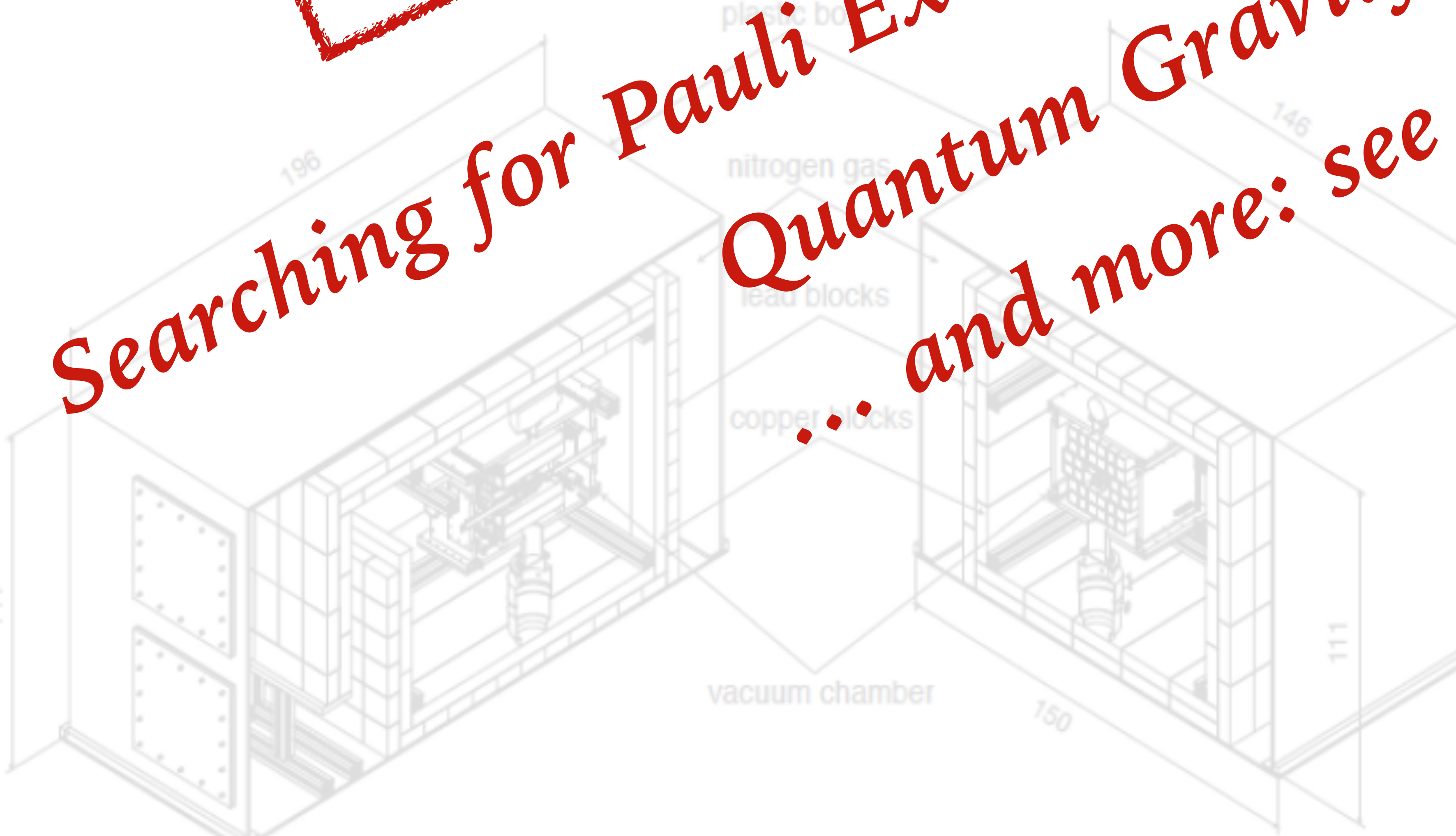
Ideal place for precision X-ray spectroscopy

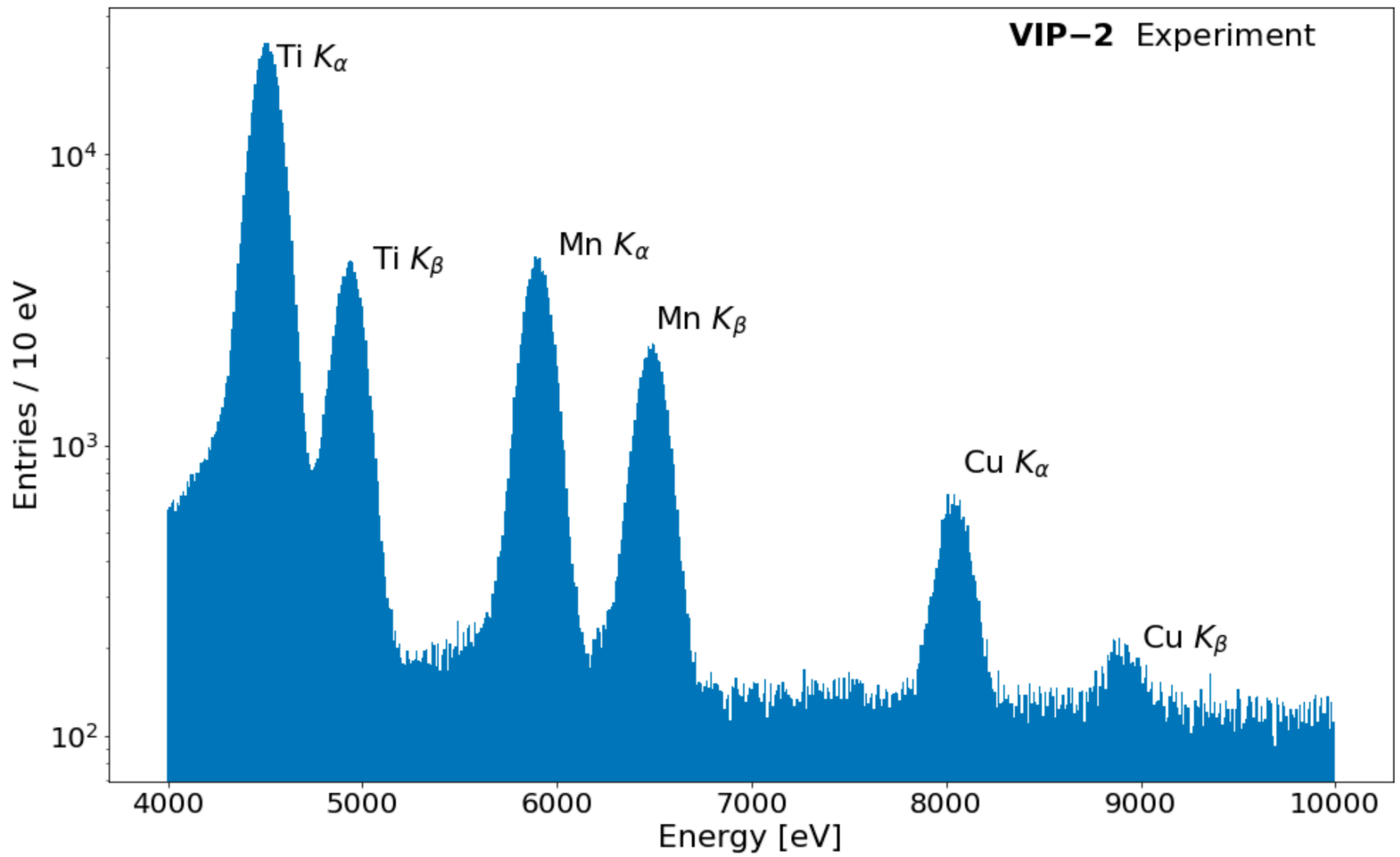
Searching for Pauli Exclusion Principle Violations

Quantum Gravity

… and more: see next talks!

Passive shielding → two layers, copper inside lead outside





Calibrated spectrum of 4 SDD arrays.

Not easy to calibrate because:

- *Copper line at orders of magnitude smaller than Ti and Mn*
- *Tiny distortions of FEE*

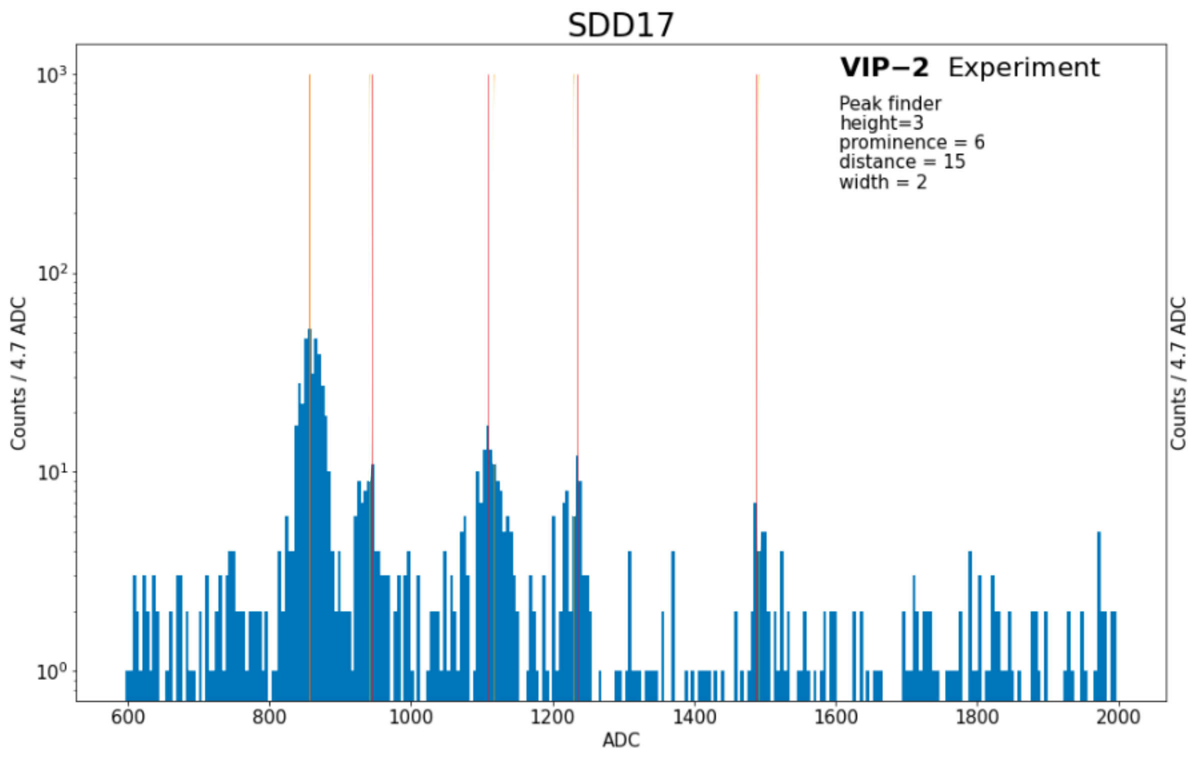
Calibration can be done in big or small batches

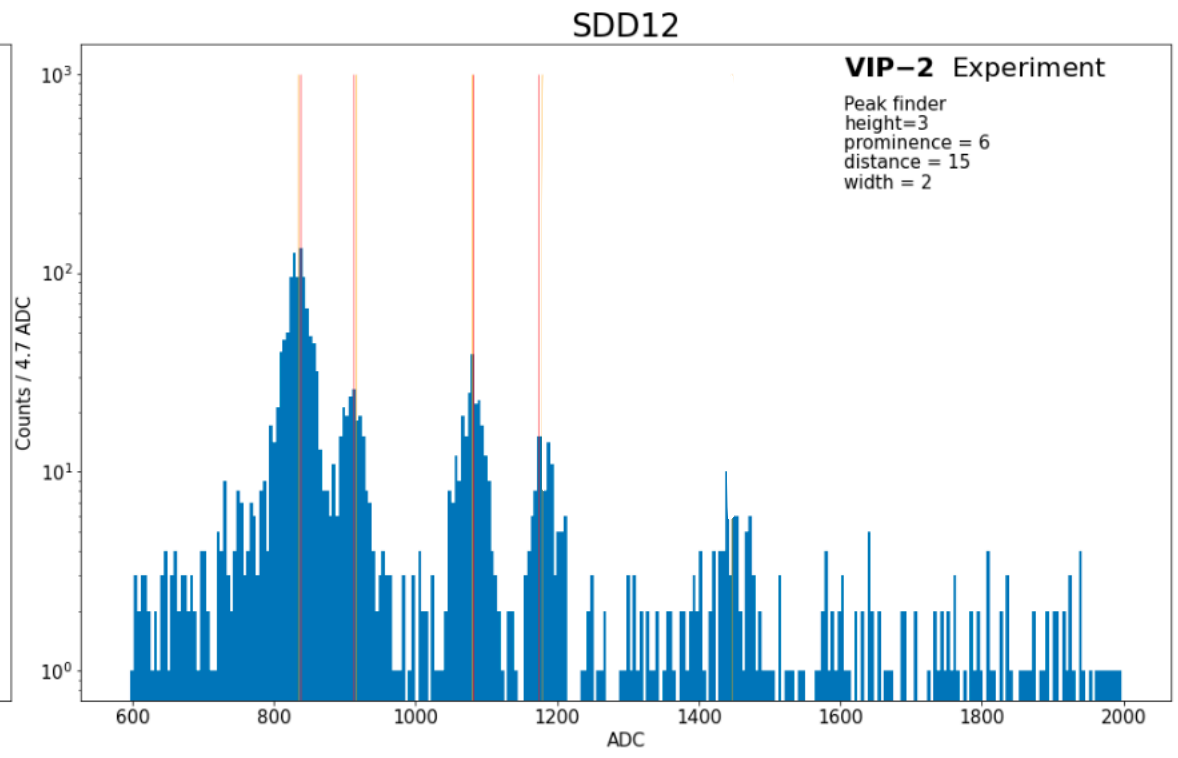
Big batches

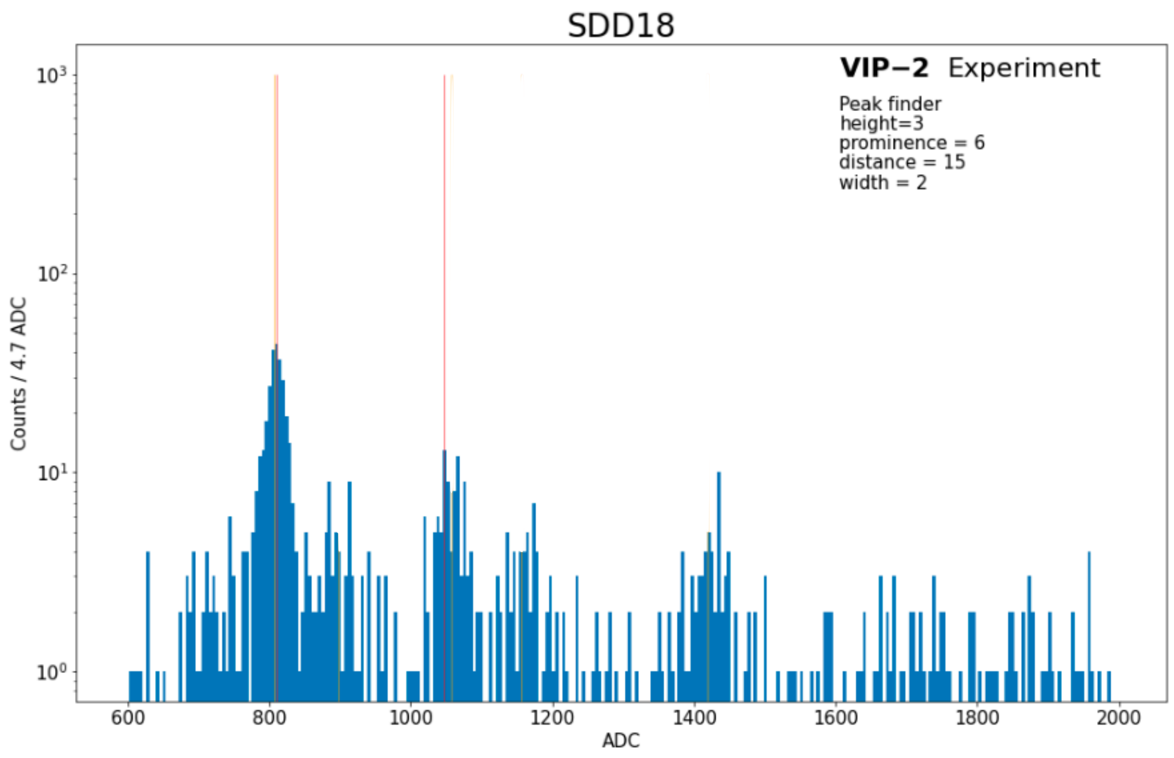
Can determine better the Copper position but cannot capture fluctuations

Small batches

Can capture fluctuations but cannot determine the Copper position well

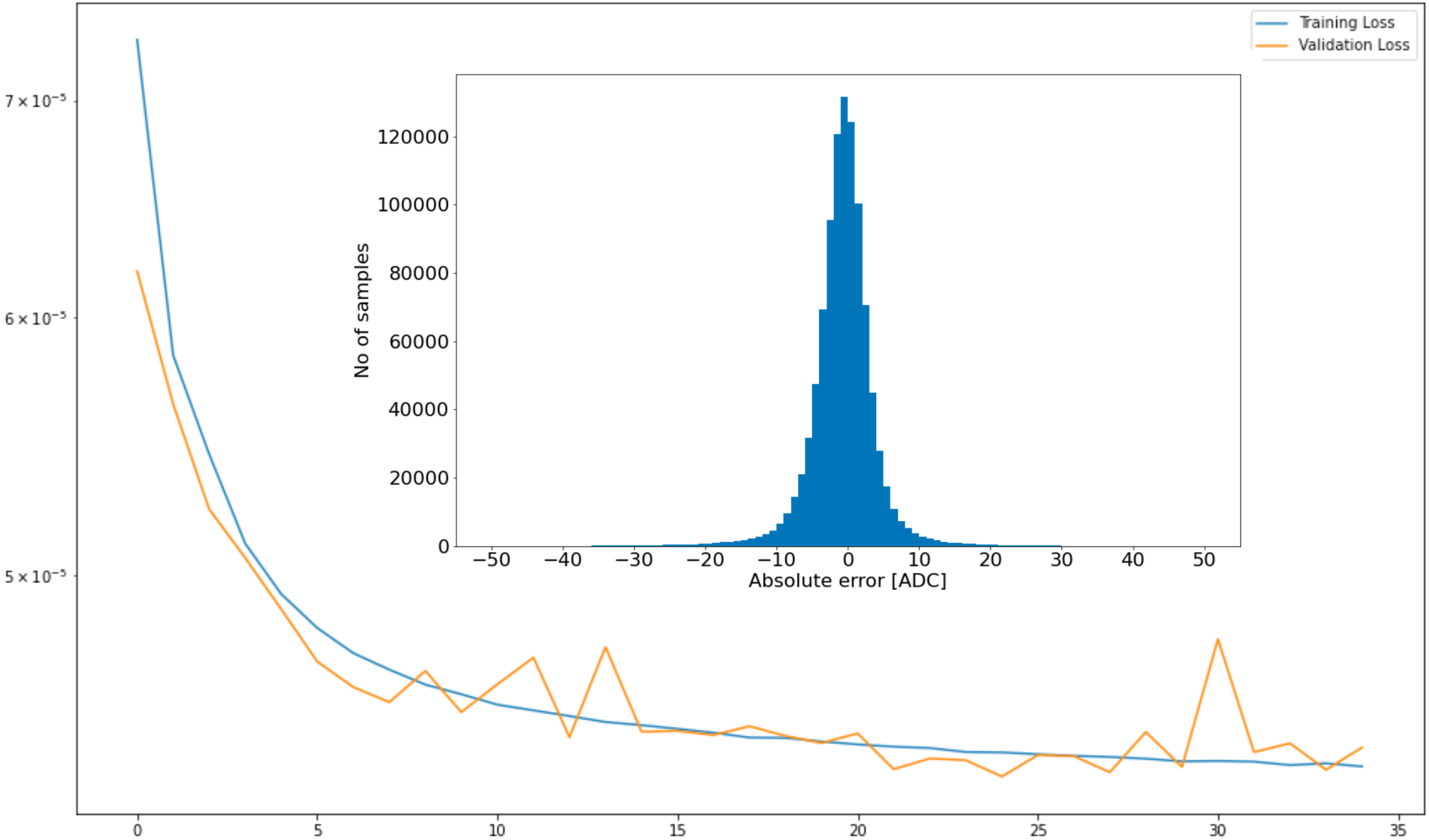


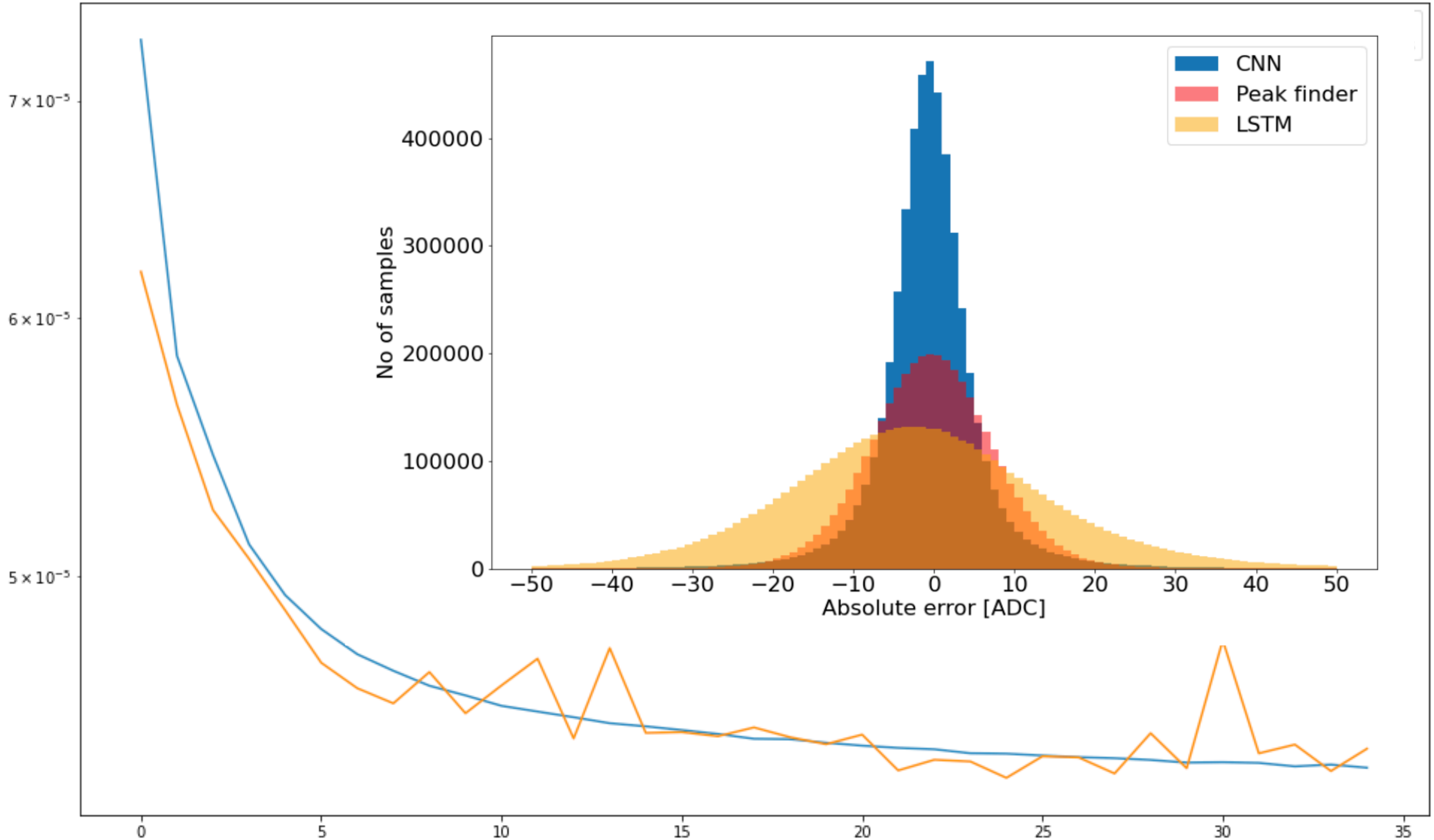




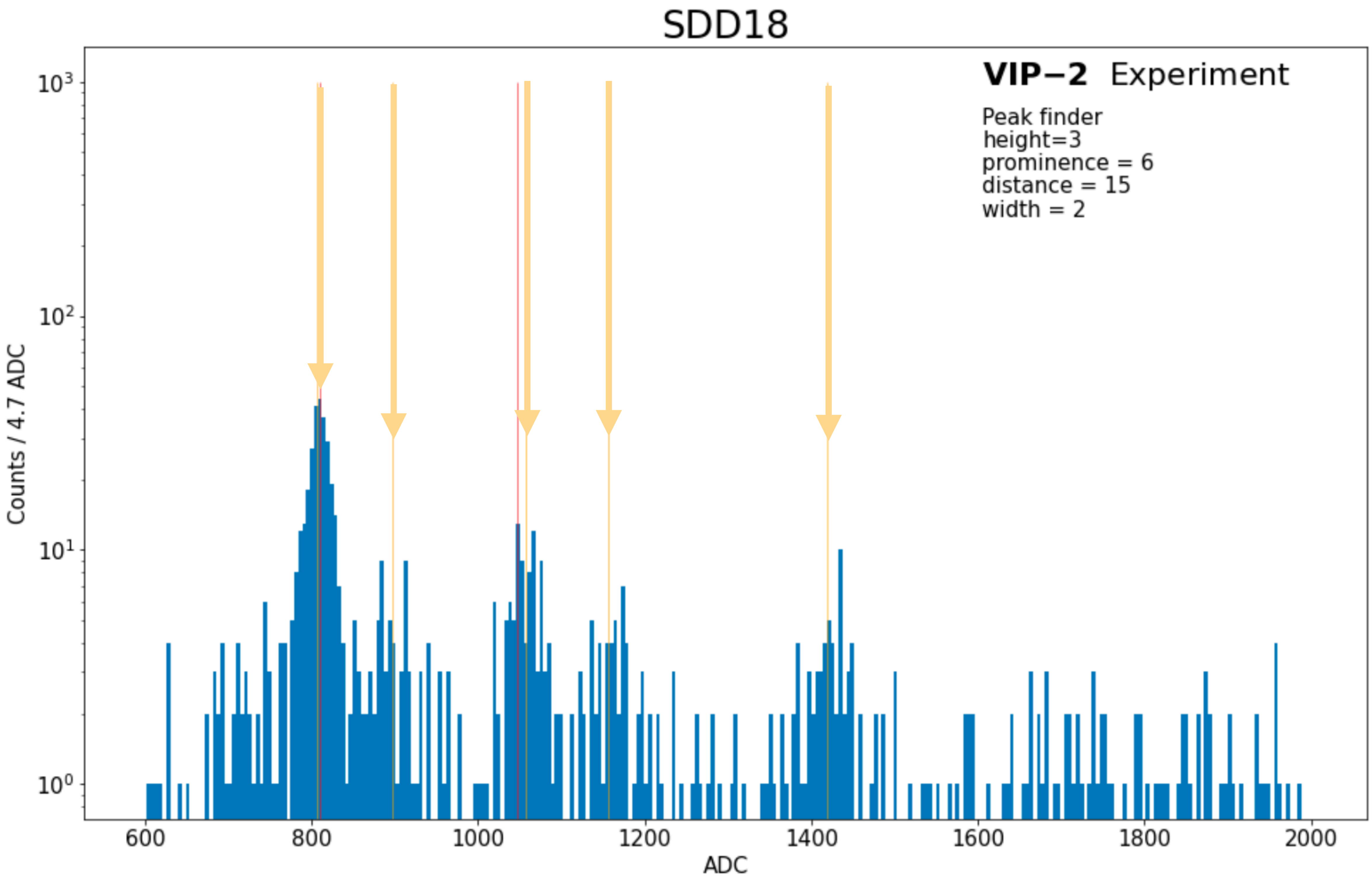
Statistical fluctuations at low counts can make the use of peak finder algorithm tricky to setup needs constant care calibrated to be resilient algo params need to be tuned

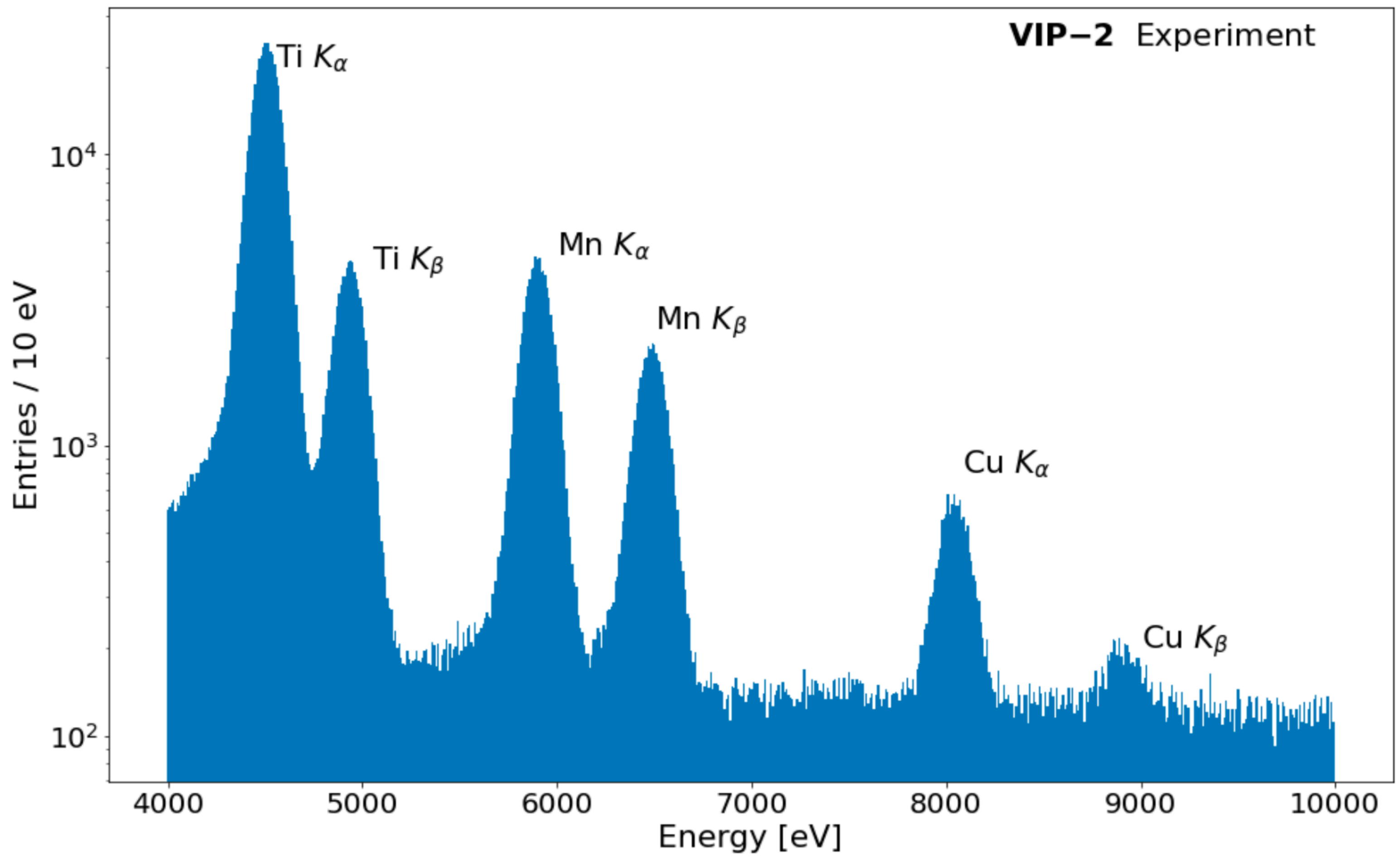
Use two step approach - 1st: convolutional neural network as peak finder




7 × 10⁻⁵
6 × 10⁻⁵
5 × 10⁻⁵
0
5
10
15
20
25
30
35
CNN
Peak finder
LSTM
No of samples
400000
300000
200000
100000
0
−50
−40
−30
−20
−10
0
10
20
30
40
50
Absolute error [ADC]

Use two step approach - 1st: convolutional neural network as peak finder





Calibrated spectrum of 4 SDD arrays.

Not easy to calibrate because:

- *Copper line at orders of magnitude smaller than Ti and Mn*
- *Tiny distortions of FEE*

Calibration can be done in big or small batches

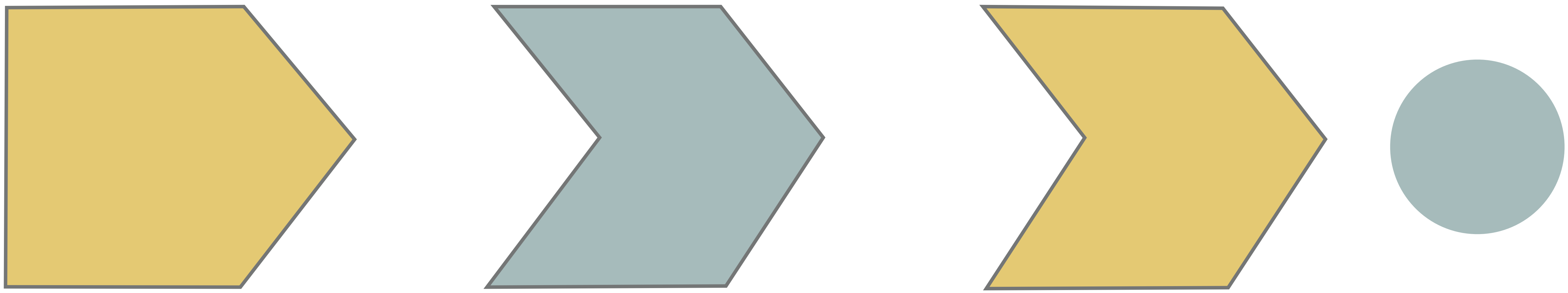
Big batches

Can determine better the Copper position but cannot capture fluctuations

Small batches

Can capture fluctuations but cannot determine the Copper position well

Use two step approach - 2nd: to the limit with differentiable programming



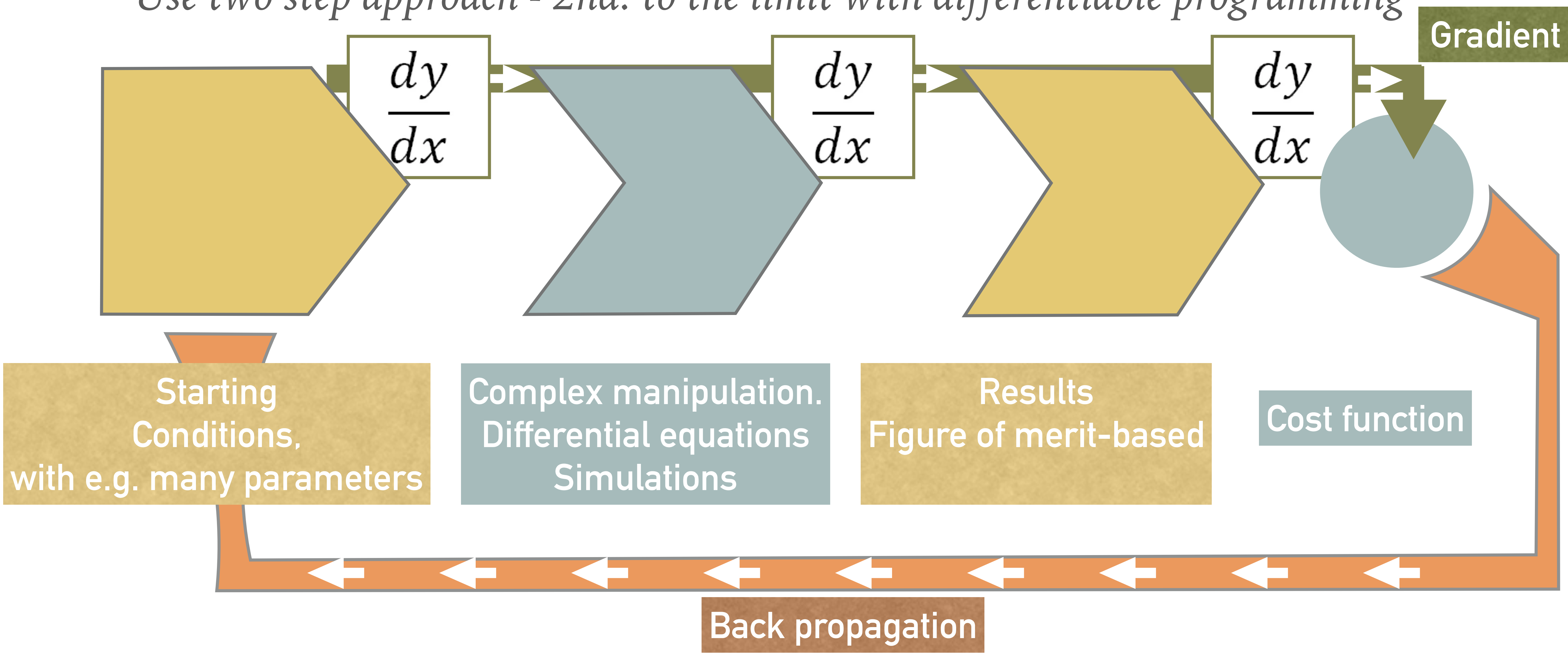
Starting
Conditions,
with e.g. many parameters

Complex manipulation.
Differential equations
Simulations

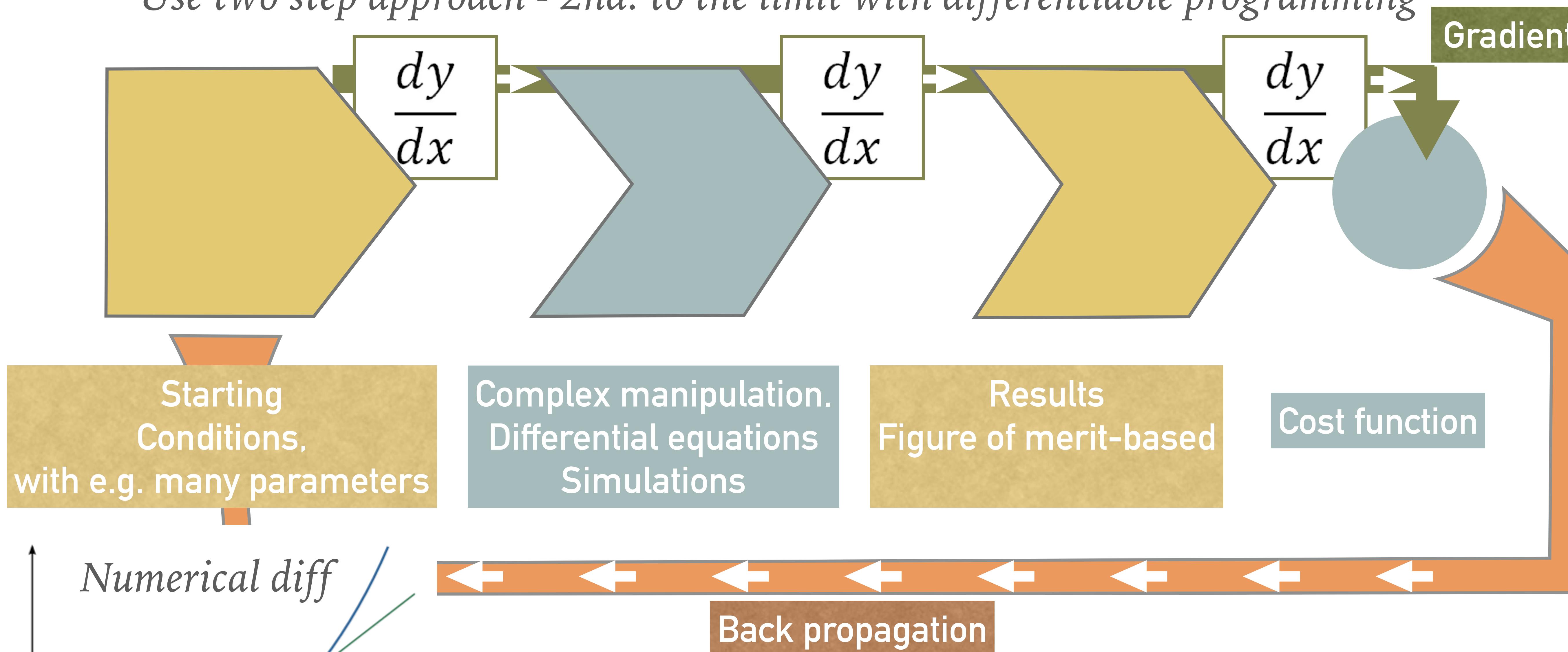
Results
Figure of merit-based

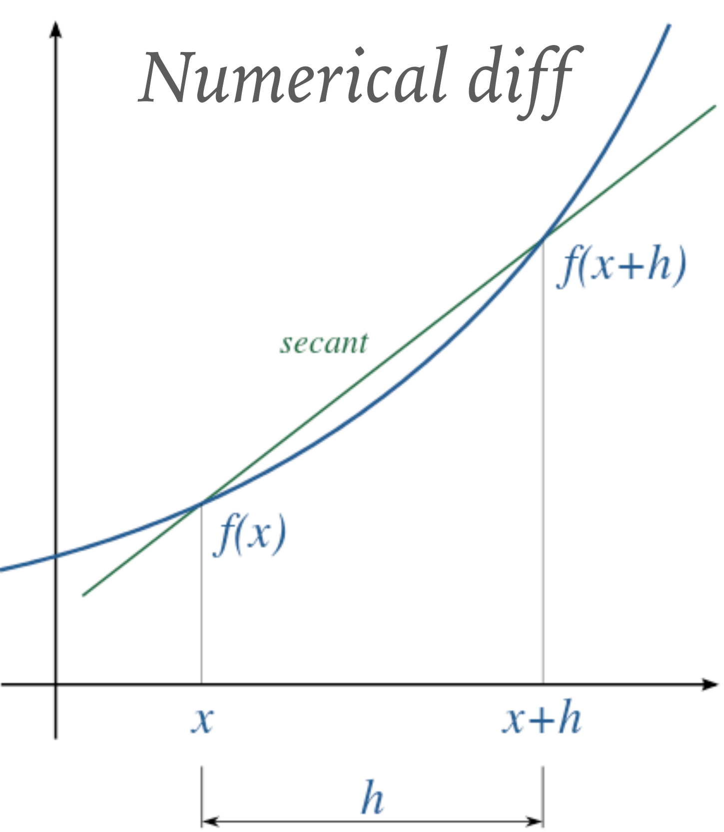
Cost function

Use two step approach - 2nd: to the limit with differentiable programming

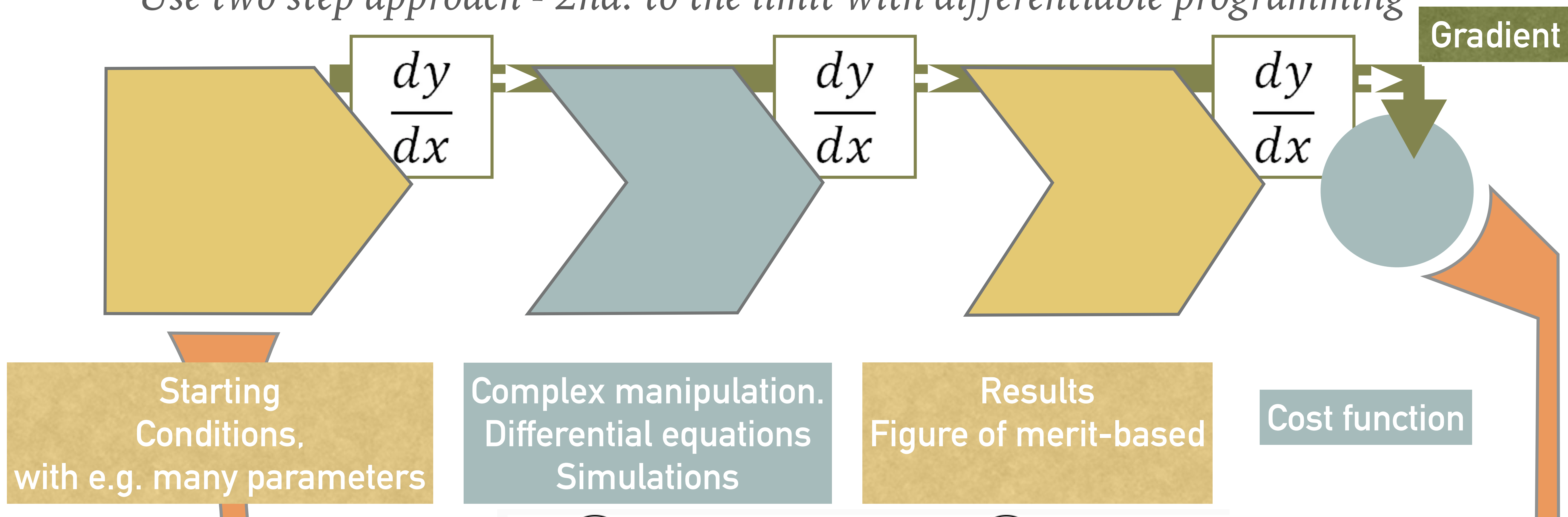


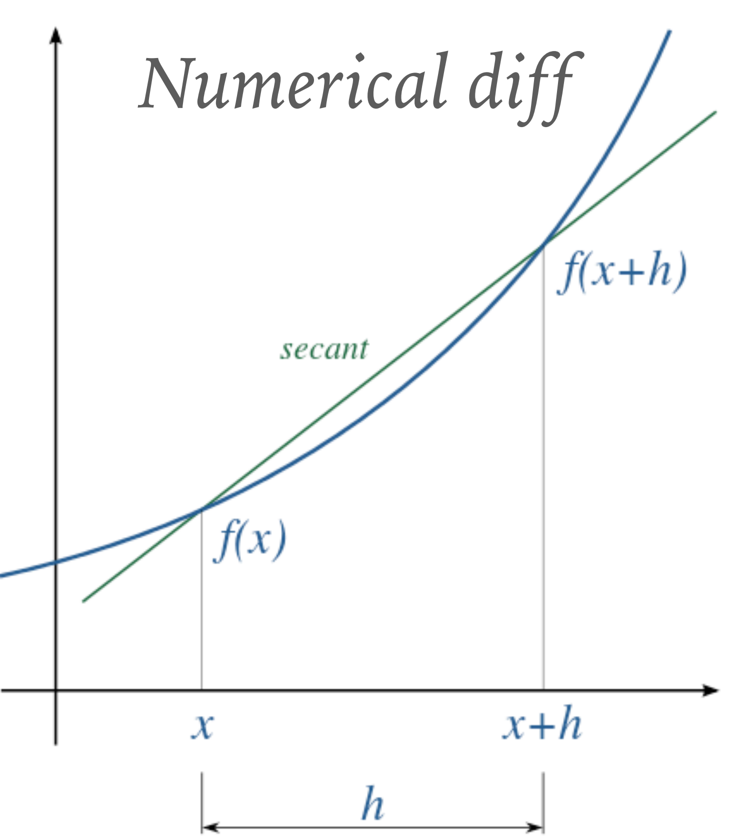
Use two step approach - 2nd: to the limit with differentiable programming

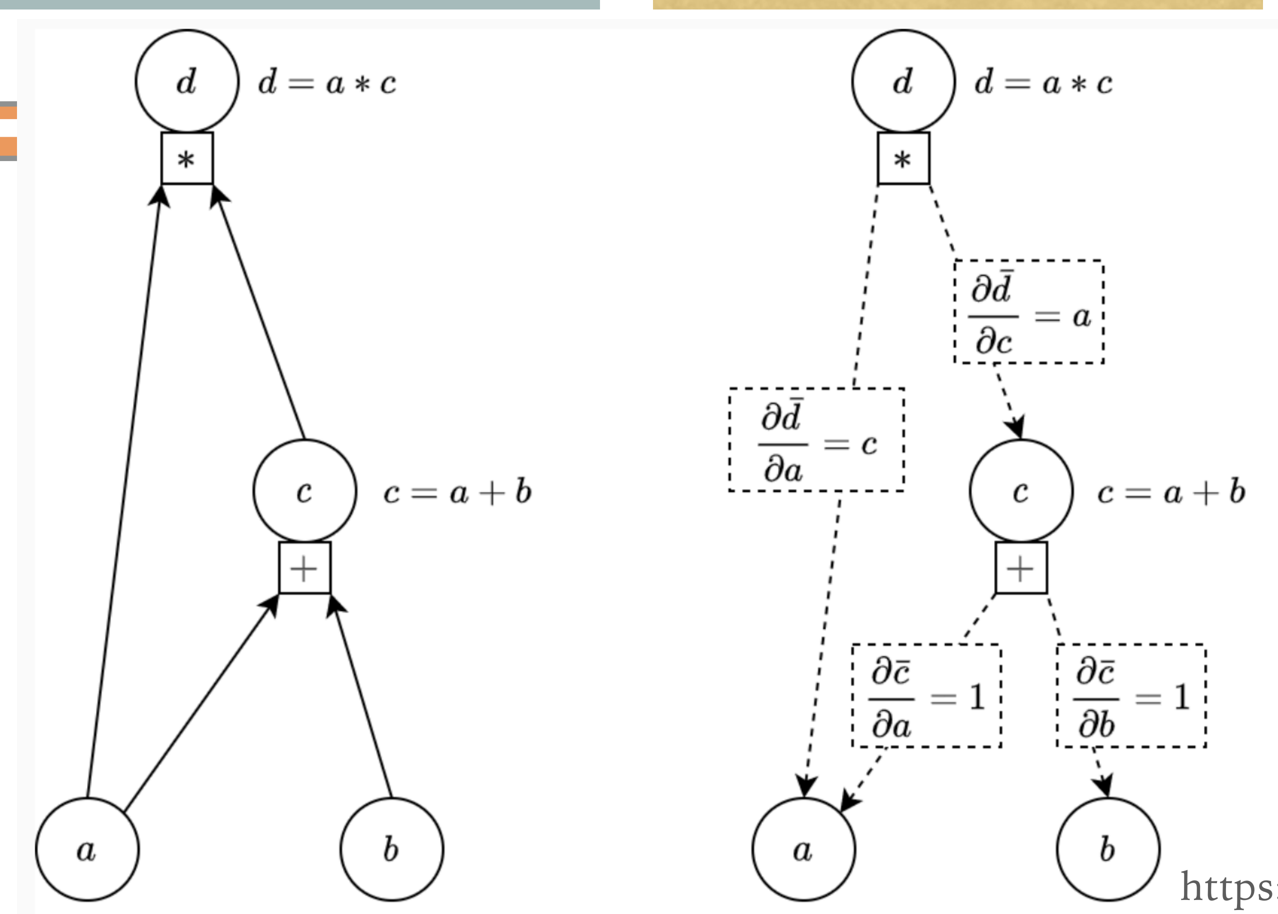




Use two step approach - 2nd: to the limit with differentiable programming







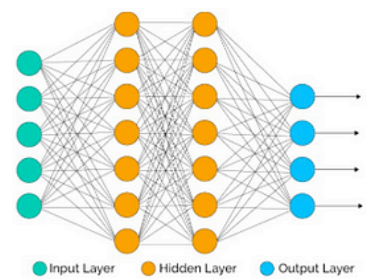
$$\frac{\partial d}{\partial a} = \frac{\partial \bar{d}}{\partial a} + \frac{\partial \bar{d}}{\partial c} * \frac{\partial \bar{c}}{\partial a}$$
$$\frac{\partial d}{\partial a} = c + a * 1$$
$$\frac{\partial d}{\partial a} = a + b + a$$
$$\frac{\partial d}{\partial a} = 2a + b$$

https://sidsite.com/posts/autodiff/

Use two step approach - 2nd: to the limit with differentiable programming

Target and environment variables → Neural Network → Control Parameters → ODE Solver → Loss

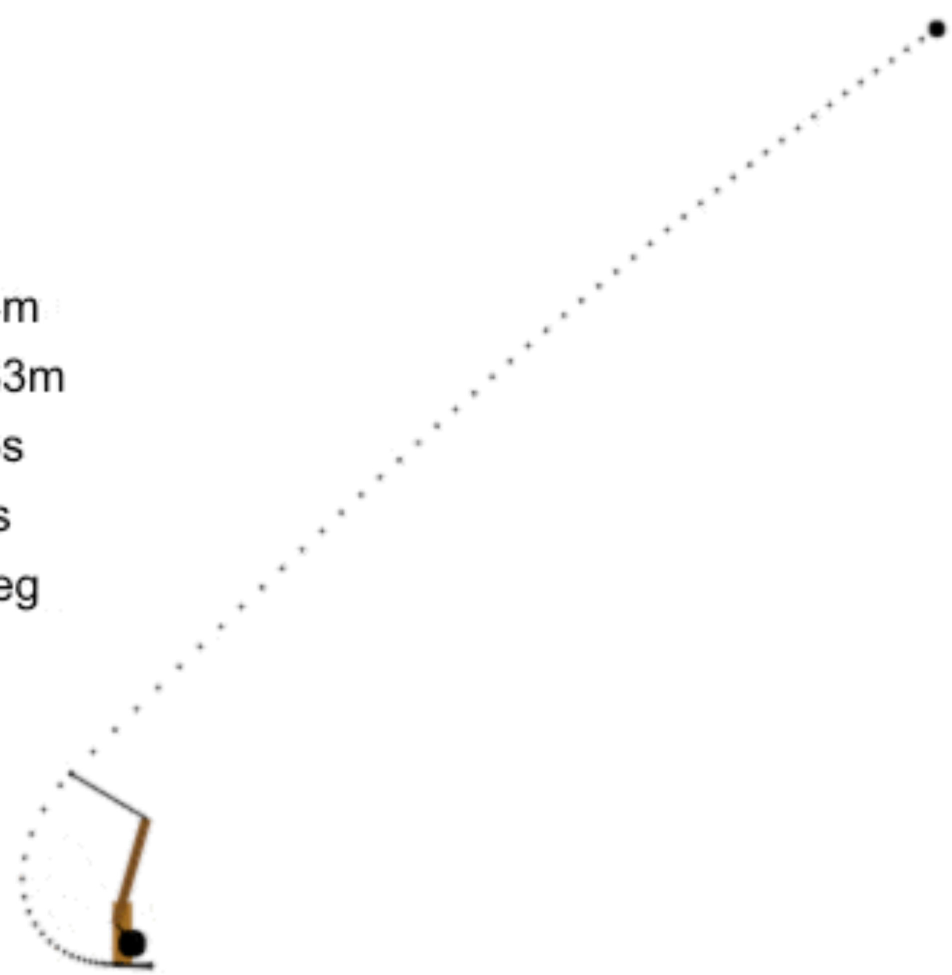
wind = -10m/s
target = 50m



angle = 25°
weight = 200kg

(target_distance – actual_distance)²

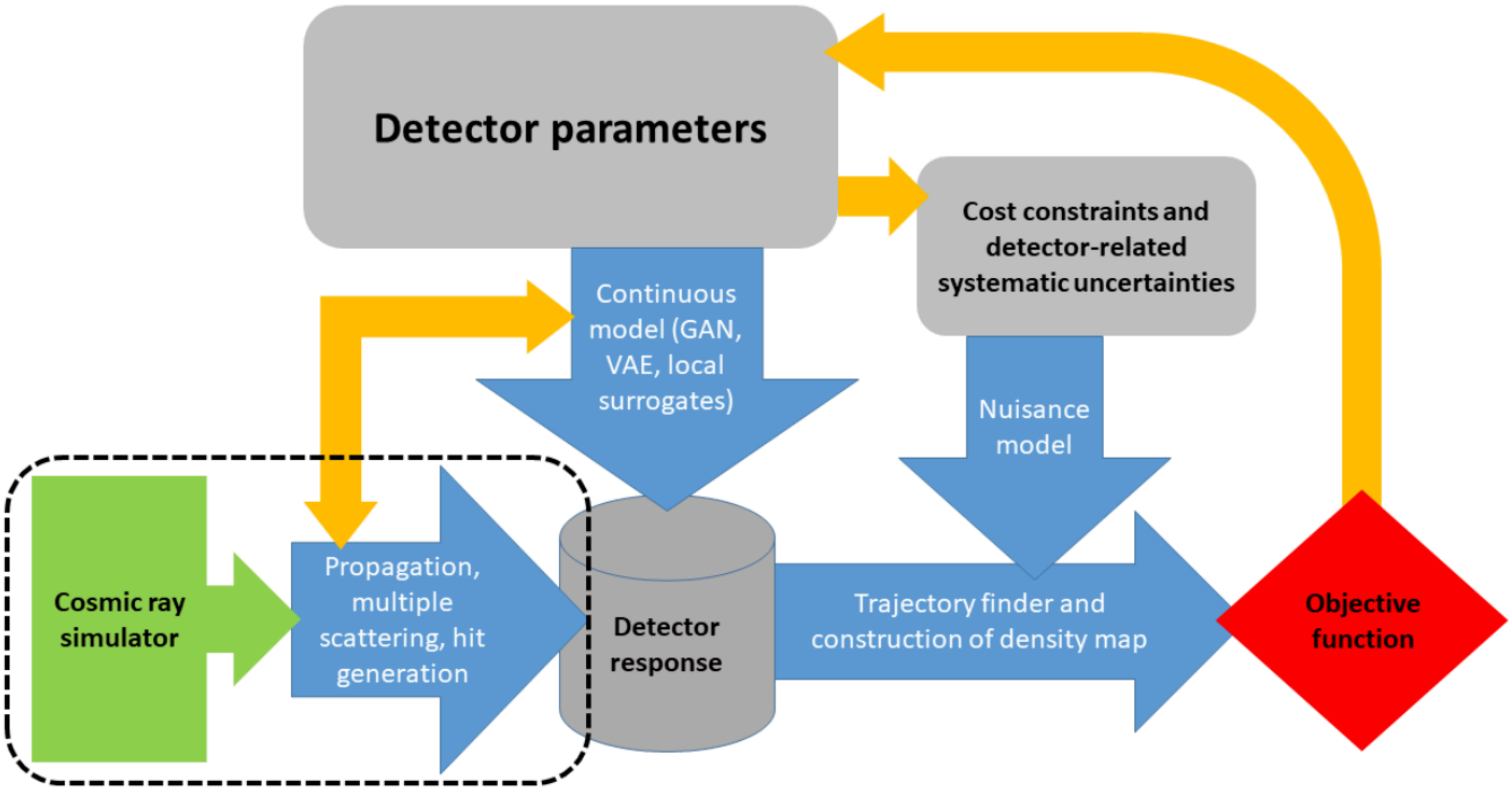
← Backpropagation



Distance	19.4m
Height	22.33m
Time	1.46s
Wind Speed	1m/s
Release Angle	45deg

https://github.com/JuliaComputing/ODSC2019

Use two step approach - 2nd: to the limit with differentiable programming

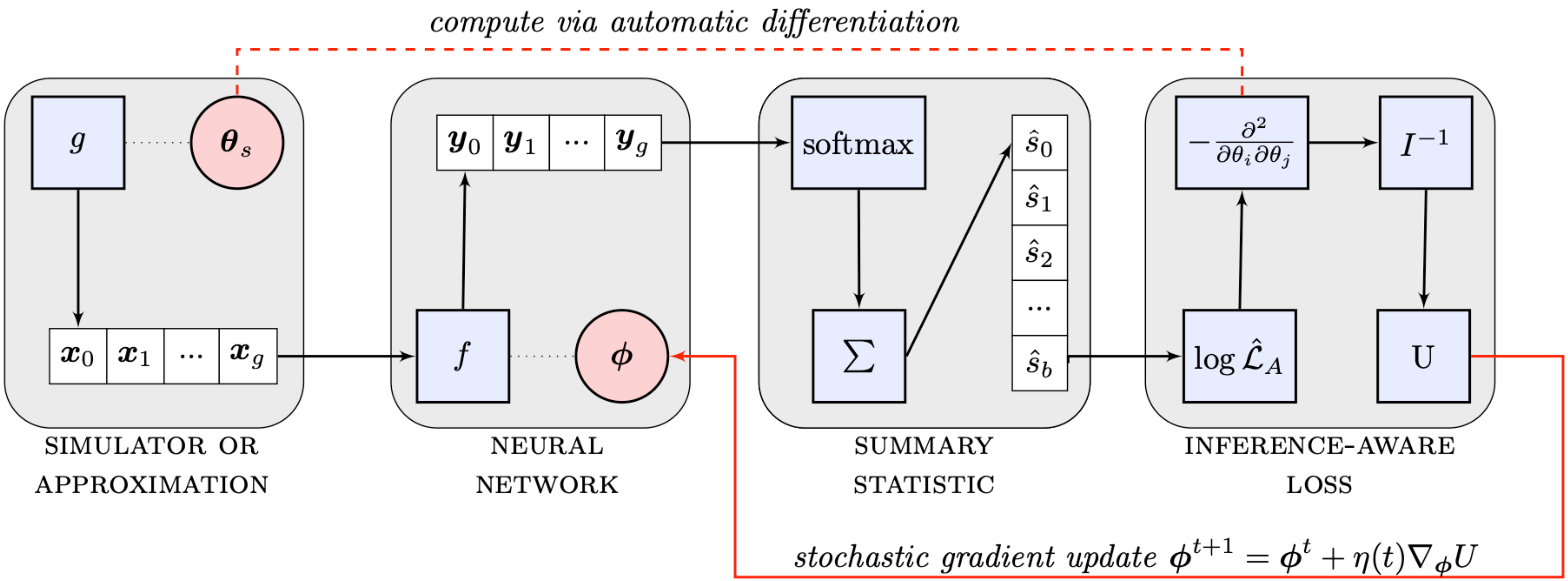


Optimization of detector design and operation

MODE
Machine-learning Optimized Design of Experiments

https://inspirehep.net/files/8f6784eac6a7f06c184cfc50757de656

Use two step approach - 2nd: to the limit with differentiable programming

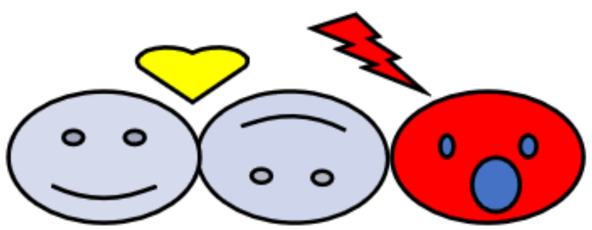


Sketch of the INFERNO algorithm. Batches from a simulator are passed through a neural network and a differentiable summary statistic is constructed that allows to calculate the variance of the POI. The parameters of the network are then updated by stochastic gradient descent.

https://cds.cern.ch/record/2807001/plots

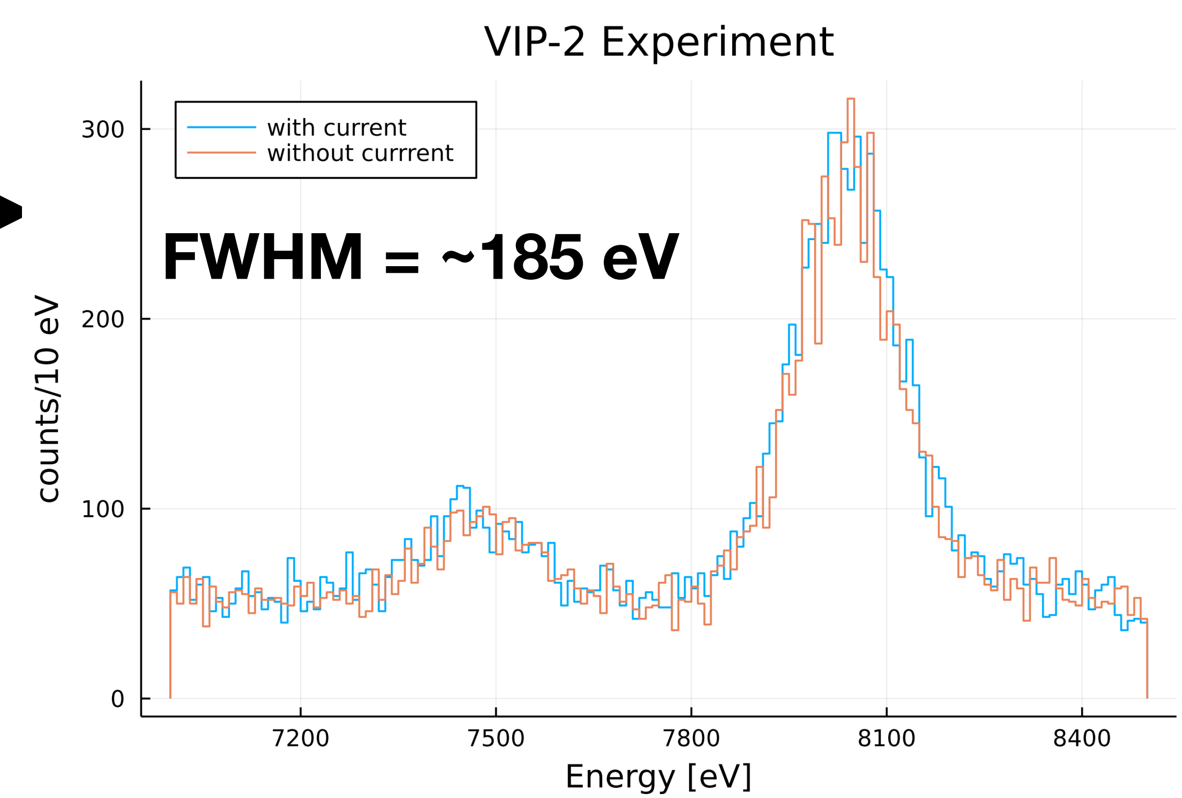
Use two step approach - 2nd: to the limit with differentiable programming

Idea: use automatic differentiation to compute gradients of functions
Use gradients to find global optima



Data → **Get calibrated data** →

Data ↓ **Fit peak positions** → **Derive calibration constants** ↑ **Get calibrated data**

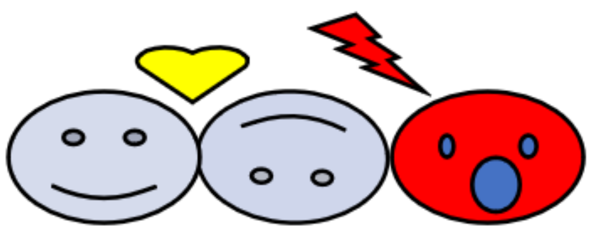


Our Calibration Flow

Use two step approach - 2nd: to the limit with differentiable programming

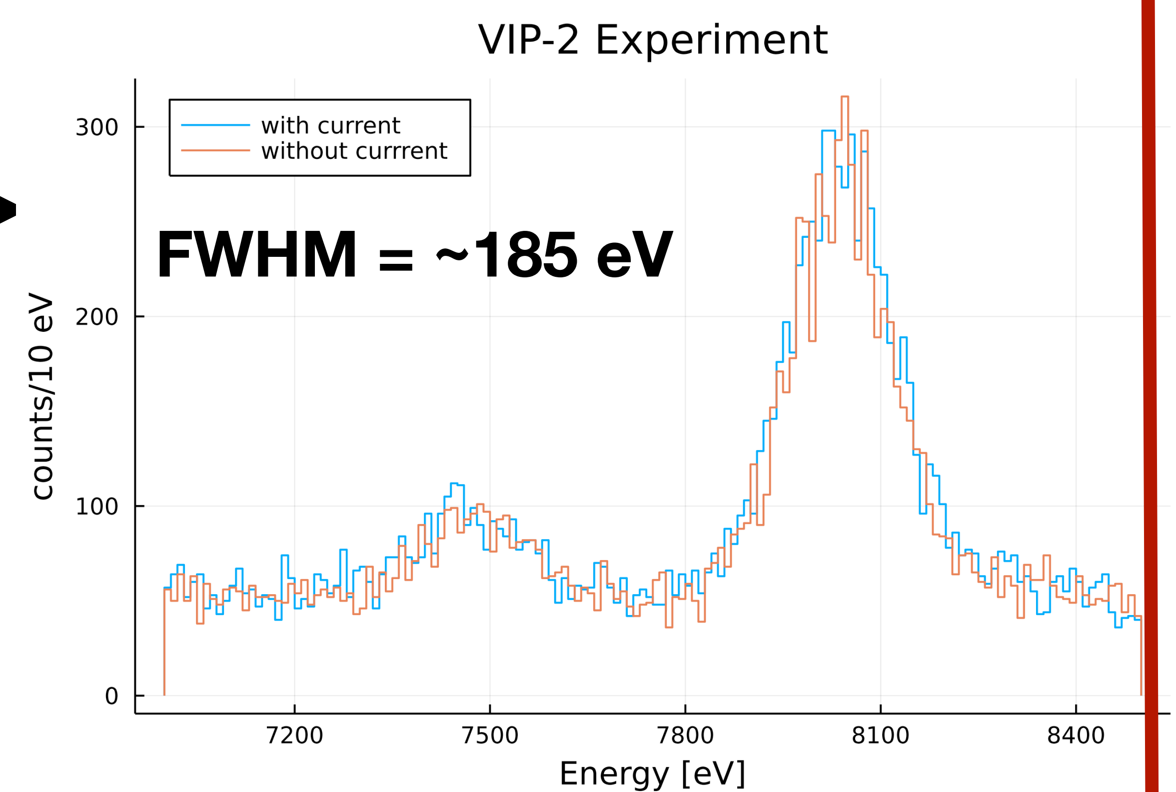
Idea: use automatic differentiation to compute gradients of functions

Use gradients to find global optima



Data

Get calibrated data

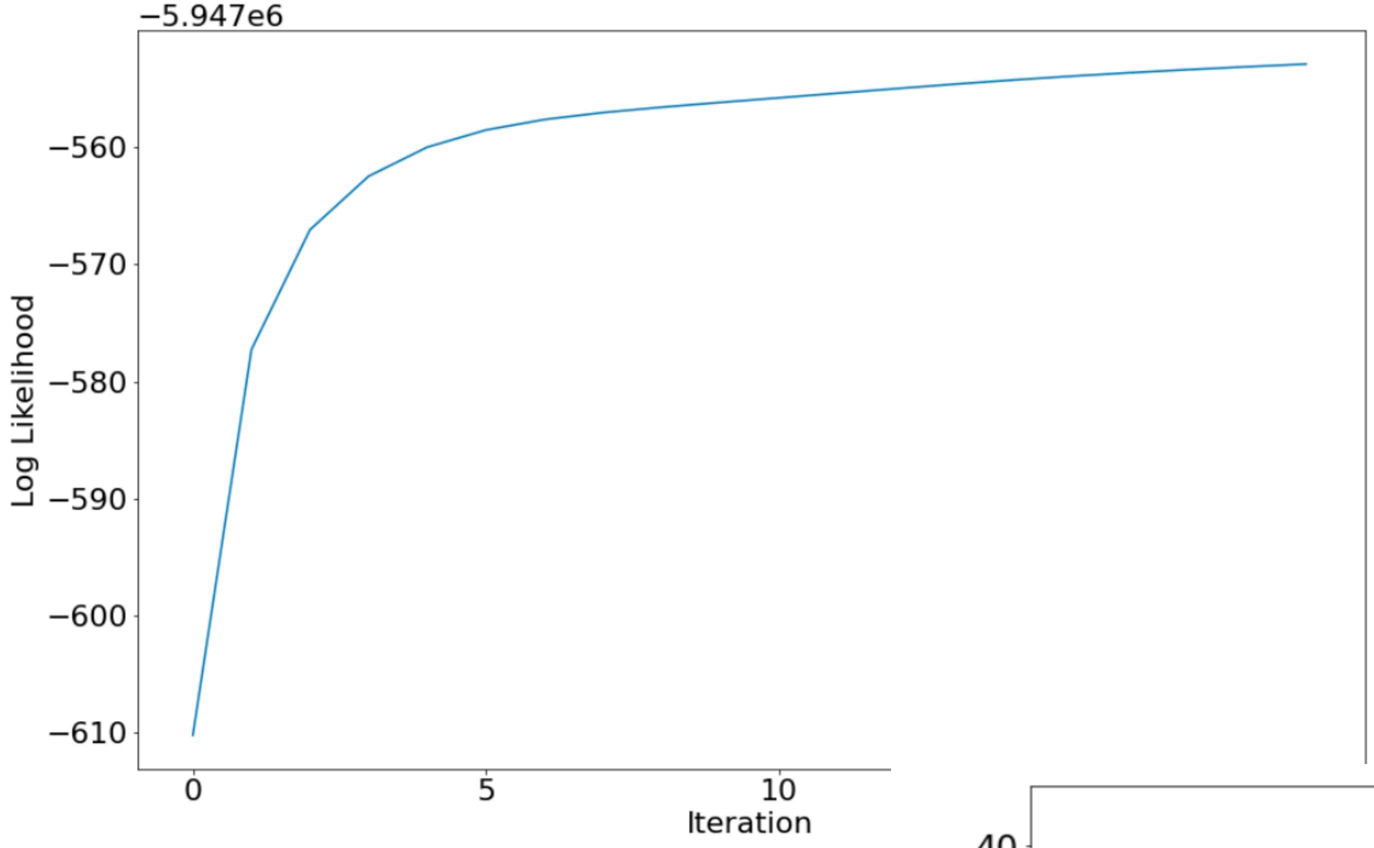


Fit peak positions

Derive calibration constants

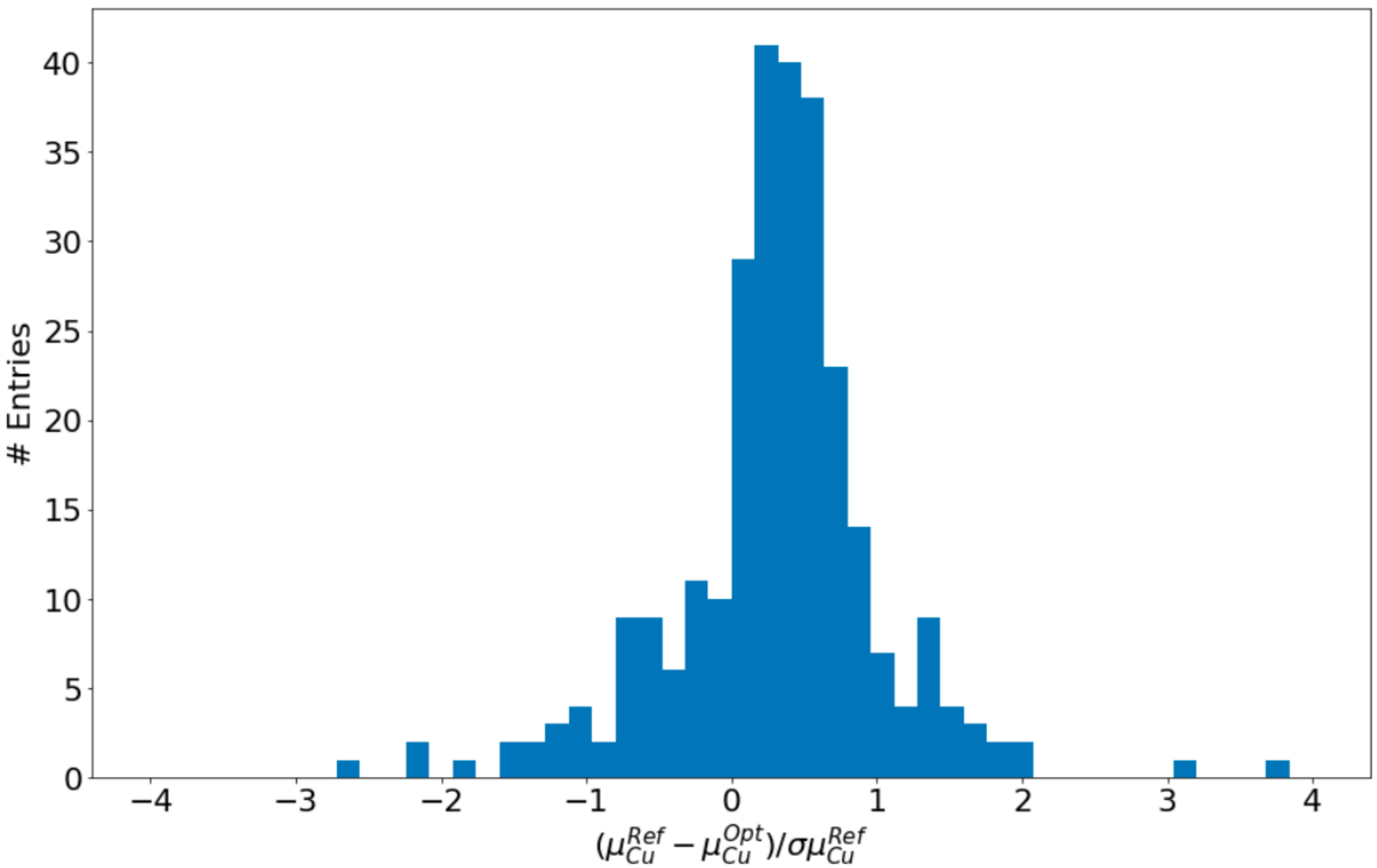
Following the gradient, change the constants to enhance FWHM

Use two step approach - 2nd: to the limit with differentiable programming



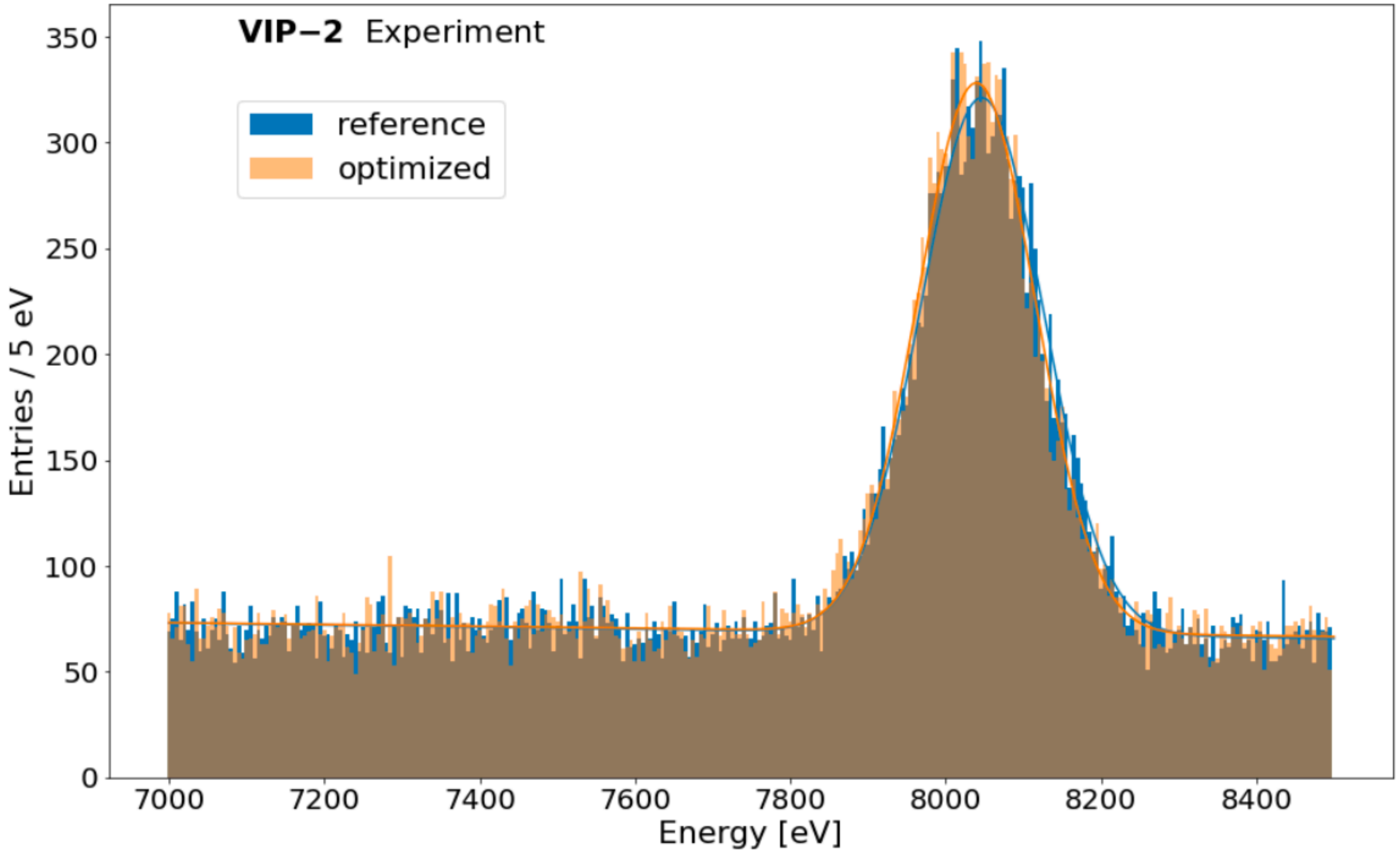
Gradient descended into plateau

Most of the calibration parameters do not change much



Results

https://arxiv.org/abs/2305.17153
Submitted to Meas. Sci. Tech.



	Position [eV]	FWHM [eV]	χ^2/ndf
Reference	8050 ± 1	185 ± 2	1.64
Optimized	8048 ± 1	176 ± 2	1.25

$$f(x, A, \mu, \sigma) = A\times\frac{51}{100}\times\text{Gauss}(x-\mu-20, \sigma)+T_2(x)+A\times\text{Gauss}(x-\mu, \sigma)+T_1(x)+m\times x+C$$

$$T_i(x) = \frac{A_i}{2\beta\sigma}\times e^{\frac{x-\nu}{\beta\sigma}\frac{1}{2\beta^2}}\times erfc\left(\frac{x-\nu}{\sqrt{2}\pi}+\frac{1}{\sqrt{2}\beta}\right)$$