



# Оптимизация электромагнитного калориметра LHCb

XIV Черенковские чтения, 20 апреля 2021

Алексей Болдырев<sup>1</sup>, Денис Деркач<sup>1</sup>, Павел Факанов<sup>1</sup>,  
Фёдор Ратников<sup>1,2</sup>, Андрей Шевелёв<sup>1</sup>

См. также:

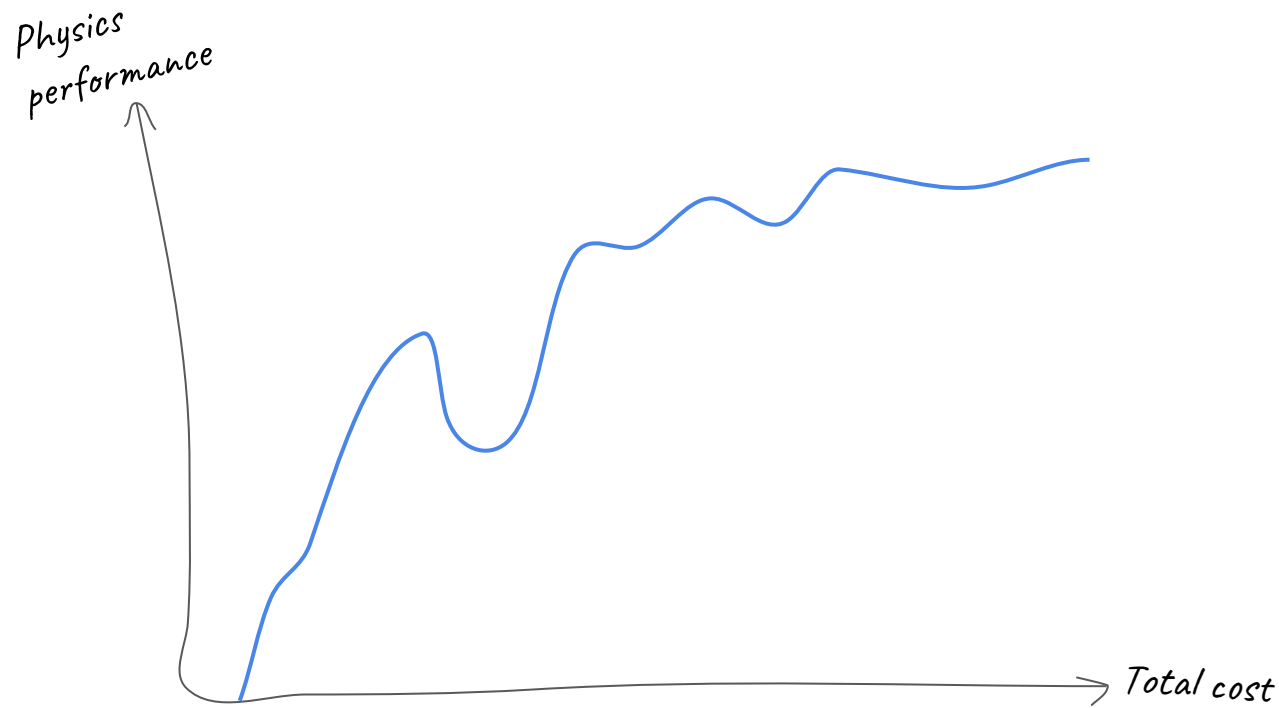
[1] [JINST 15 C05032 \(2020\)](#)

[2] [JINST 15 C09030 \(2020\)](#)

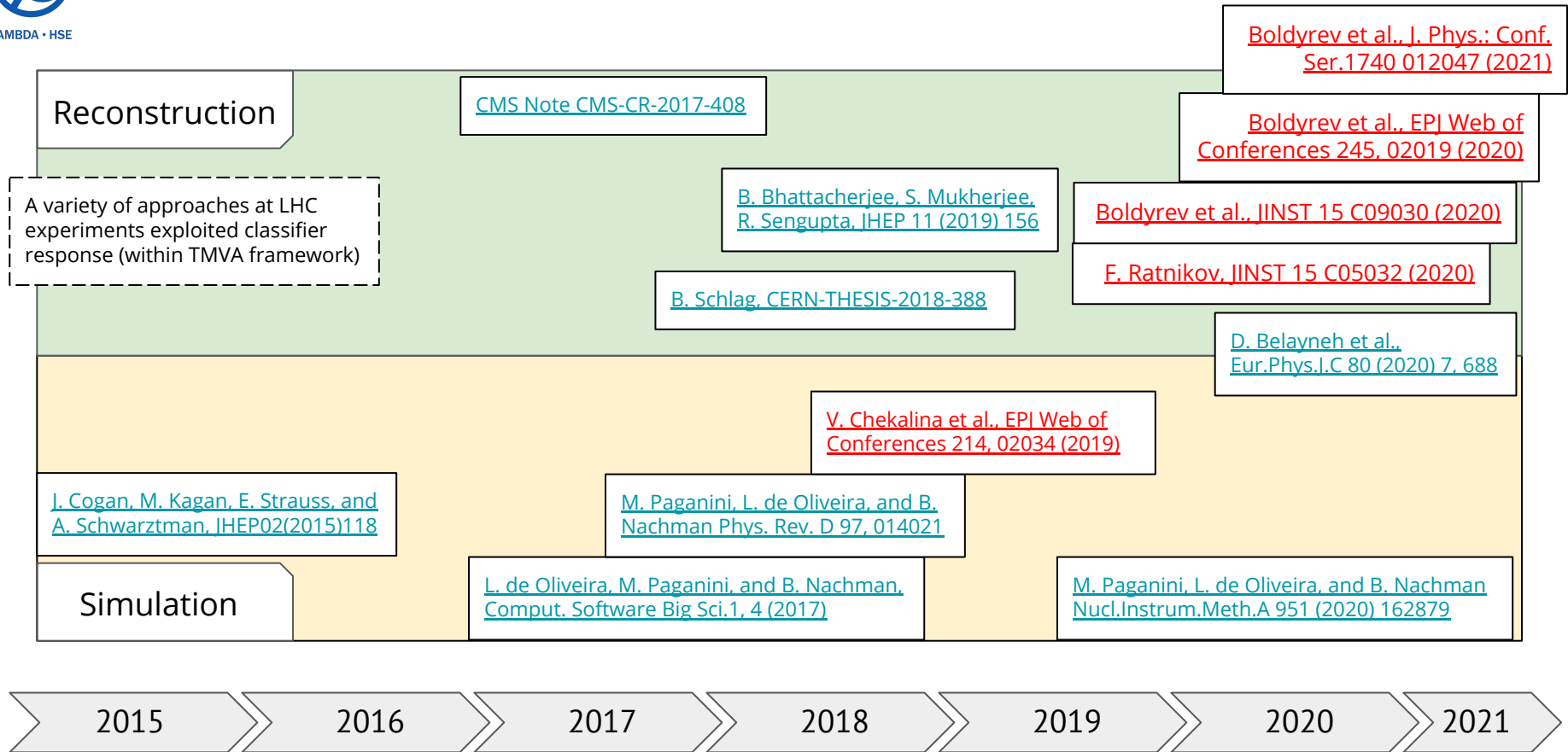
[3] [EPI Web of Conferences 245, 02019 \(2020\)](#)

[4] [J. Phys.: Conf. Ser.1740 012047 \(2021\)](#)

# Aim of optimization



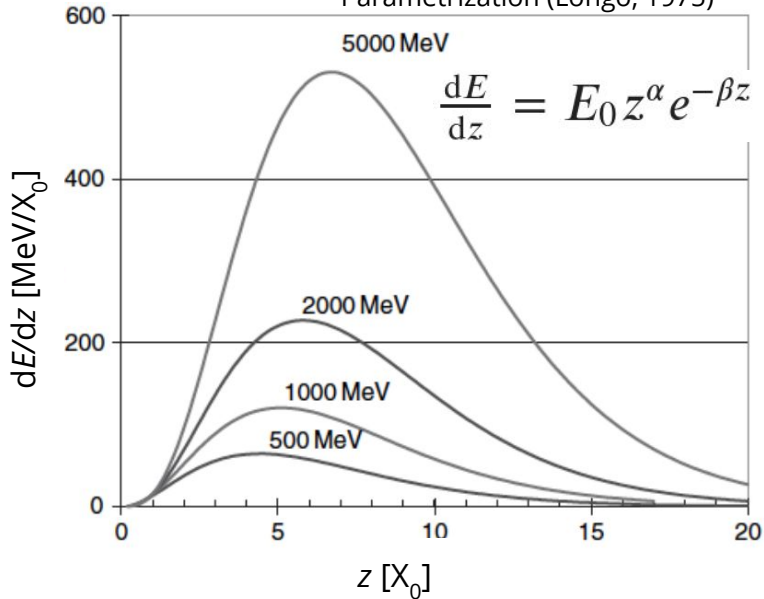
# ML in calorimetry



# Calorimetry in a nutshell

## Longitudinal EM shower profile

Parametrization (Longo, 1975)

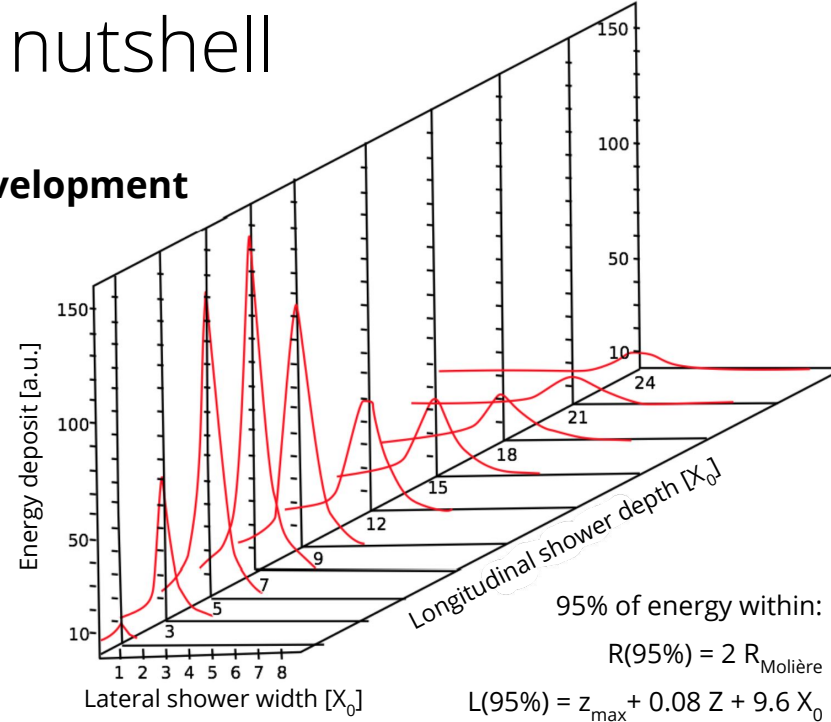


Differences between showers induced by  $\gamma$  &  $e$

$$z_{\max} = \frac{\alpha-1}{\beta} = \ln\left(\frac{E_0}{E_C}\right) + C$$

$$C_\gamma = -0.5, C_e = -1.0$$

## EM Shower development



## Energy resolution

$$\frac{\sigma_{\text{reco}}}{E_{\text{reco}}} = \frac{a}{\sqrt{E_{\text{gen}}}} \oplus b \oplus \frac{c}{E_{\text{gen}}}$$

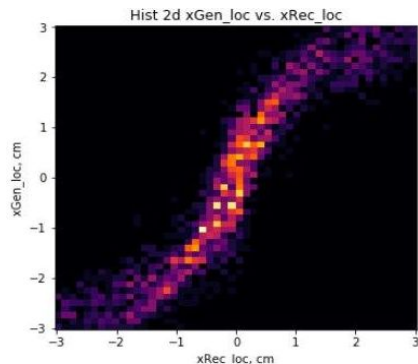
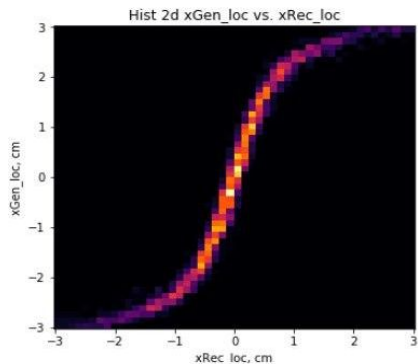
( $E$  is measured in GeV)

terms:  
 $a$  - stochastic  
 $b$  - constant  
 $c$  - noise

# Conventional constituents of detector R&D

## Simulation in general:

- Detailed Sim is very slow to cover all possible options
- Fast Sim may be inaccurate, when describing complicated response
  - Spatial resolution of large angled particles
  - Sidebands of Inv. Mass distributions } are poorly described  
(otherwise require fine-tuning)



Hit position:

Small angles

Large angles

See slide #14 for the  
alternative approach

# Conventional constituents of detector R&D

## **Simulation** (at high pile-up conditions):

Pure signal events + averaged pile-up contribution

- Modelling and selection of signal- and background-populated regions of the detector

## **Reconstruction:**

Overlapped signal and background events

- background subtraction

[Calo-specific] Clusters can be reconstructed:

- via its nested association with tracks and tracks with vertices
- Using time information

See slide #17 for the alternative approach

Despite the fact the complicated reco following such a simulation can achieve outstanding performance, the fine-tuning of many parameters comes with significant overhead.

# The idea of optimal detector design

To choose optimal design of the detector:

- Automate entire R&D cycle, whenever possible
- Define a metric
- Each step of the R&D cycle need to be scrutinised
- Build a pipeline on top of them
- Evaluate the importance of each step of R&D cycle

# The parameters in scope

We can solve the problem how to arrange sensitive elements of the detector in effective and generalised way if we know the following:

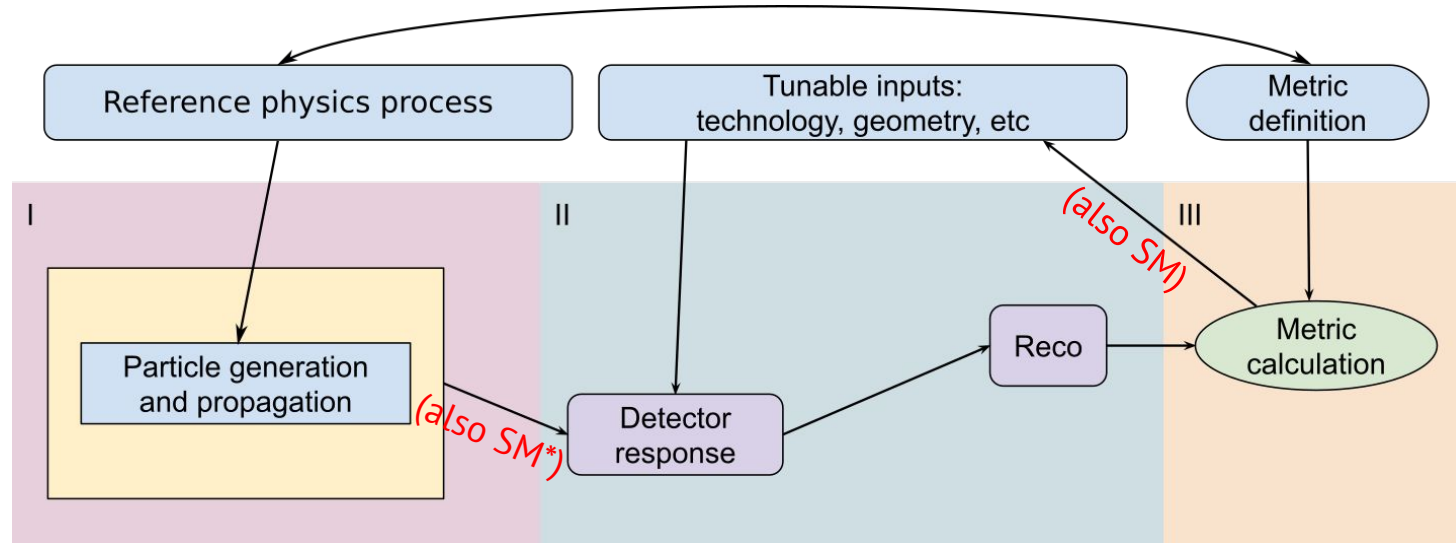
- Accurately simulated responses of given detector element technology
- Cost of that technology
- Metrics obtained from reference physics processes within required physics conditions

We aim to optimize physics performance metrics & overall cost



# The pipeline

\*SM = Surrogate Models



Optimisation cycle itself does not depend on the modules technology & arrangement, reconstruction, metric, etc.

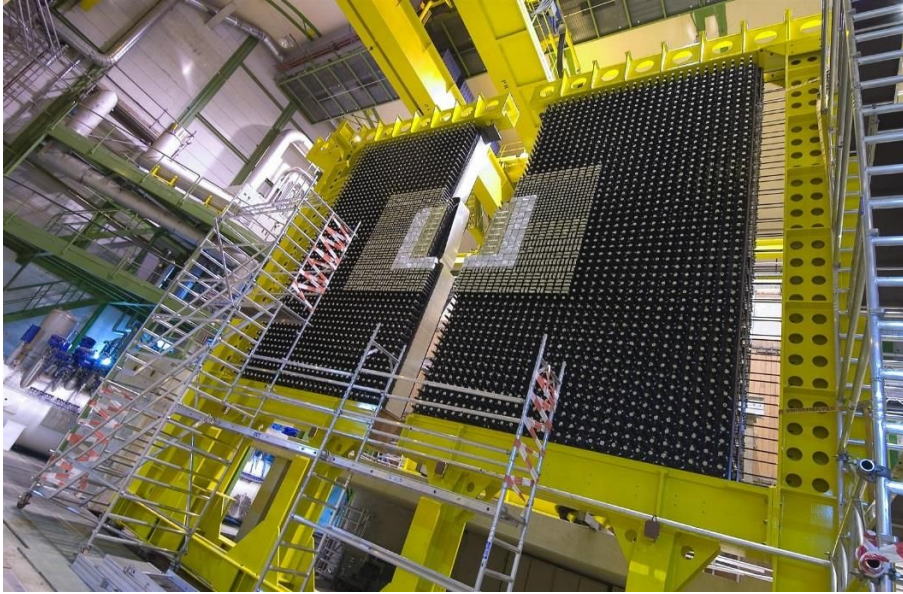
In our realisation of ECAL simulation:

- Step I requires 250 core hours (once). Realised in Docker container.
- Steps II-III require **20 core hours per option**. Realised in vectorised python or in Docker.

Sounds far too much... There is an option to significantly reduce the number of calls of the simulation  
(see slides #15-17)

# LHCb ECAL

## Current configuration



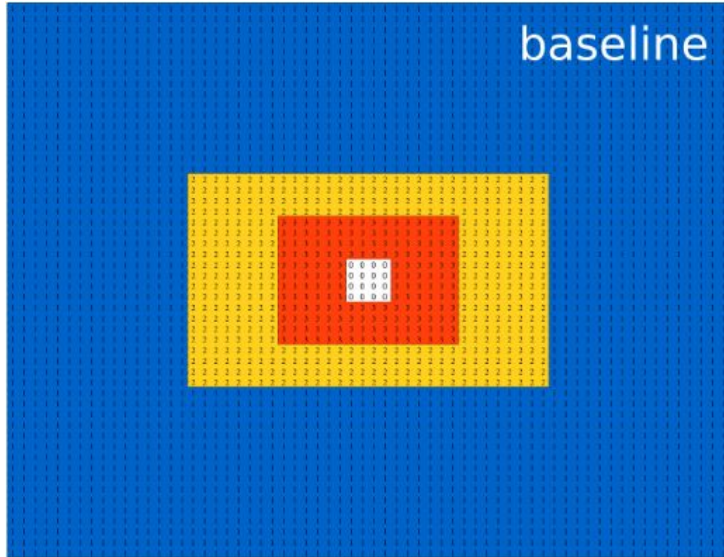
Size: 7.8x6.3x0.5 m



Module size 12x12 cm<sup>2</sup>

176 inner modules: 9 cells with size 4x4 cm<sup>2</sup>  
448 middle modules: 4 cells with size 6x6 cm<sup>2</sup>  
2688 outer modules: 1 cell with size 12x12 cm<sup>2</sup>

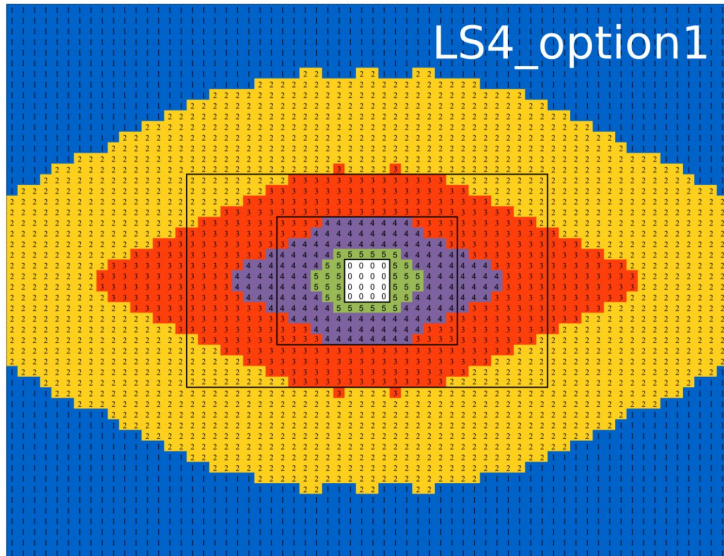
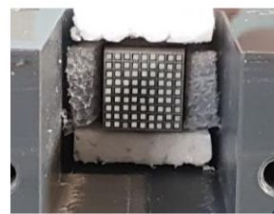
# LHCb ECAL



| Module type                                                                                                                                 | # of modules    |
|---------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
|  (inner): 3x3 cells<br>(4.04x4.04 cm <sup>2</sup> each)  | 176 (1536 ch.)  |
|  (middle): 2x2 cells<br>(6.06x6.06 cm <sup>2</sup> each) | 448 (1792 ch.)  |
|  (outer): single cell<br>(12.12x12.12 cm <sup>2</sup> )  | 2688 (2688 ch.) |

Starting from current configuration

# Future LHCb ECAL



Reuse of current “Shashlik” modules

- 1: Outer region, cell size =  $12.12 \times 12.12 \text{ cm}^2$
- 2: Middle region, cell size =  $6.06 \times 6.06 \text{ cm}^2$
- 3: Inner region, cell size =  $4.04 \times 4.04 \text{ cm}^2$

New “Spacal” modules

- 4: cell size =  $3.03 \times 3.03 \text{ cm}^2$
- 5: cell size =  $1.515 \times 1.515 \text{ cm}^2$   
(+ longitudinal split)



## Questions for future ECAL:

- What is the best configuration for given modules (fix cost) in terms of given physics metric?
- What is the best way to arrange a certain number of new modules?





# Producing samples & responses in G4

## Reference physics

**sample:**  $B_s^0 \rightarrow J/\psi(\rightarrow \mu^+\mu^-)\pi^0(\rightarrow \gamma\gamma)$  Signal events are generated using Pythia8.

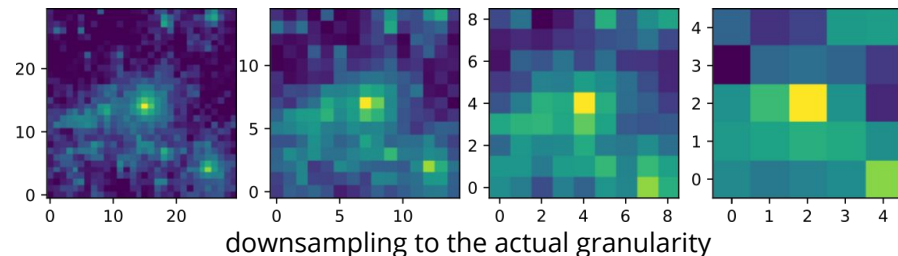
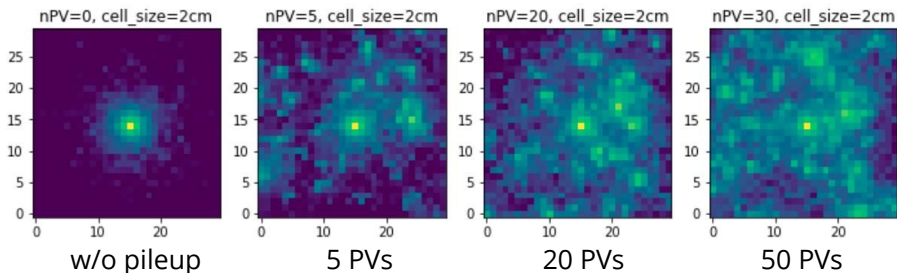
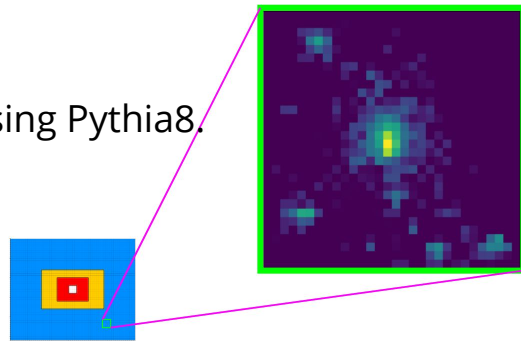
**Background sample:** LHCb Upgrade MC Minimum Bias sample

We consider background contributions from  $\gamma, \pi^+, \pi^-, e^-, e^+, n, p$

For each of the signal/background particle we:

- Record type, momentum, hit position and time at the front of the ECAL
- Perform Geant4 standalone simulation of clusters in  $N \times N (*66)$  cells(\*layers) using the momentum & type as input

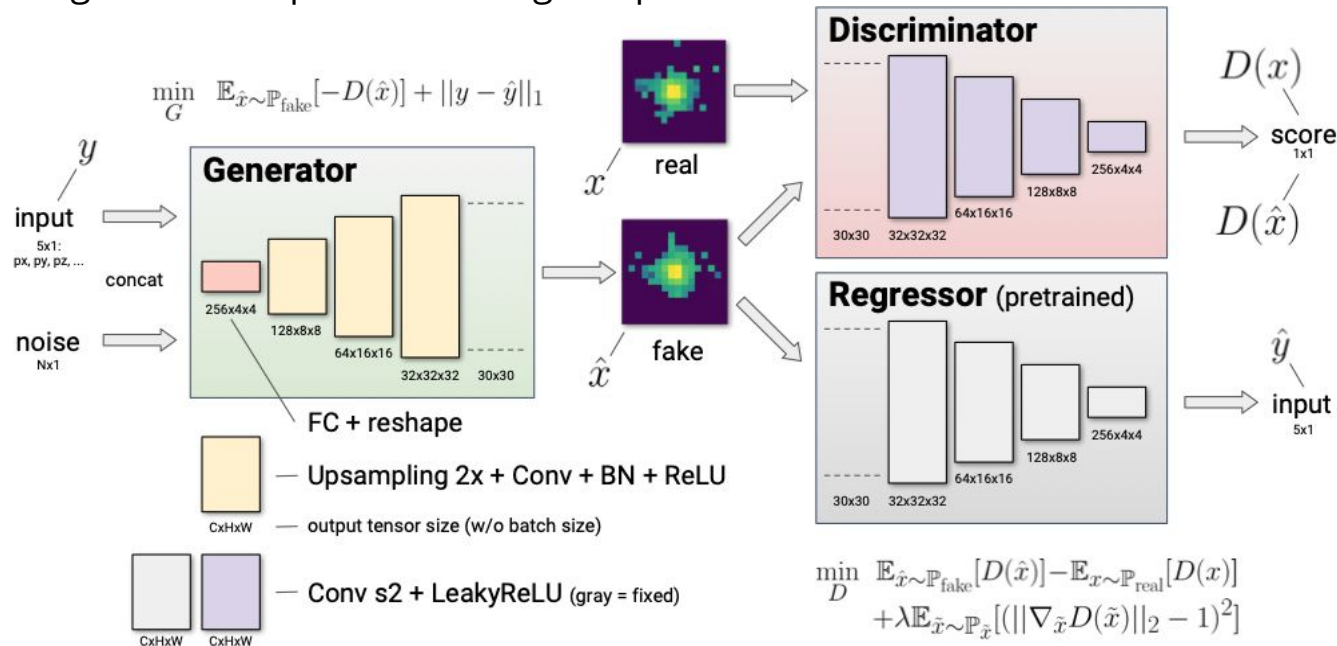
Thus, we have the **library** of the mapping of particle (px, py, pz, type) and its electromagnetic cluster.  
In future, there is possibility to use GAN-generated library.



# Generating responses using GAN

(WIP)

- Collect GEANT responses for the calorimeter technology of track parameters in standalone setup
- Train conditional generative model on simulated data
- Use the model to generate response for the given particle

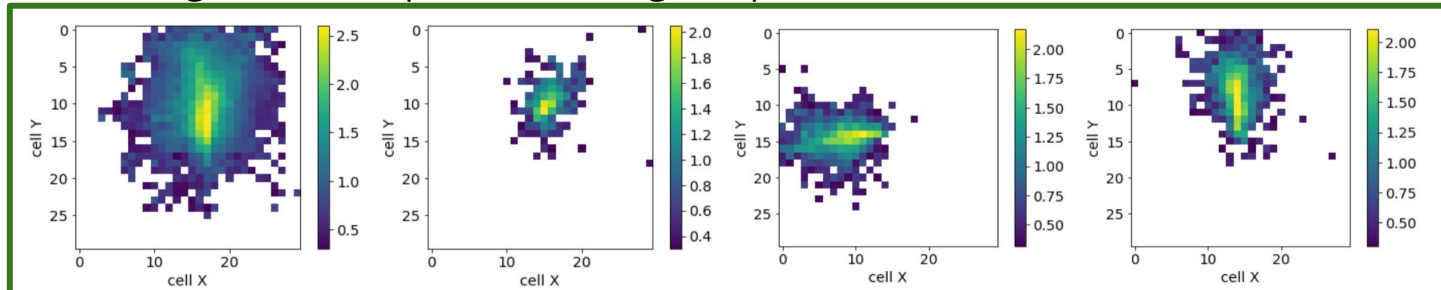


# Generating responses using GAN

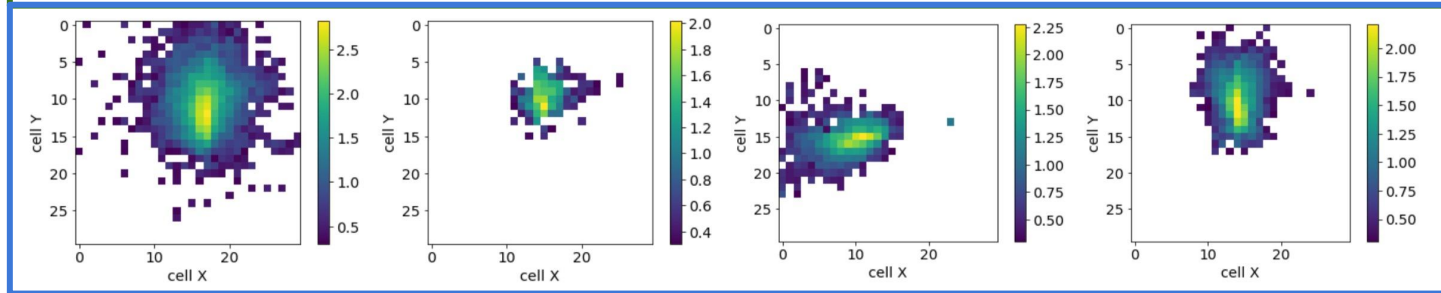
(WIP)

- Collect GEANT responses for the calorimeter technology of track parameters in standalone setup
- Train conditional generative model on simulated data
- Use the model to generate response for the given particle

GEANT4



GAN



(a)

$E_0 = 63.7 \text{ GeV}$

(b)

$E_0 = 6.5 \text{ GeV}$

(c)

$E_0 = 15.6 \text{ GeV}$

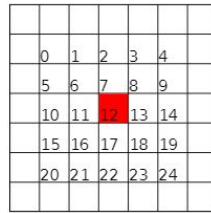
(d)

$E_0 = 15.9 \text{ GeV}$

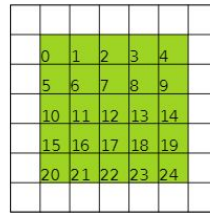
# ML-based reconstruction

Motivation: Reco automatisation

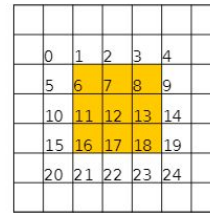
We're looking for the cell contained maximum energy deposit (seed).



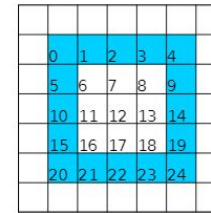
$E_{seed}$  or  $E_{seed}^2$



$\sum_i E_i$  or  $(\sum_i E_i)^2$



$\sum_i E_i$  or  $(\sum_i E_i)^2$



$(\sum_i E_i)^2$

Two regressors allows us to reconstruct the  $\pi^0$ :

- XGBoost for Energy reconstruction
- XGBoost for Spatial reconstruction

Chosen performance metric is the width of the  $m_{inv}(\pi^0)$  fit.

Features:

- Energy deposits of 25 cells around seed cell (2\*25 in case of Z-split modules)
- Barycenter
- Time information
- Sums, squared sums, rings, etc. of energy deposits





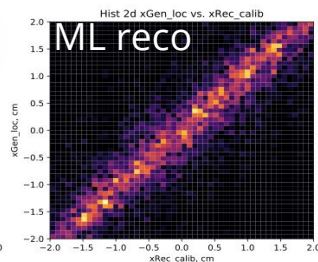
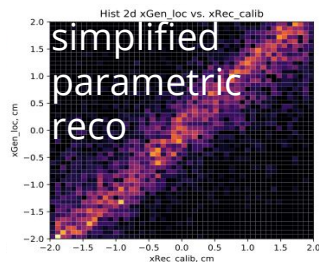
LAMBDA • HSE

# Spatial and Energy reconstruction

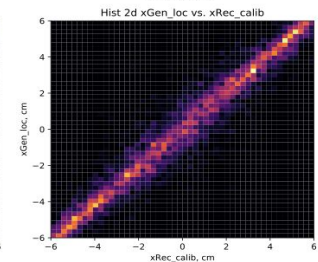
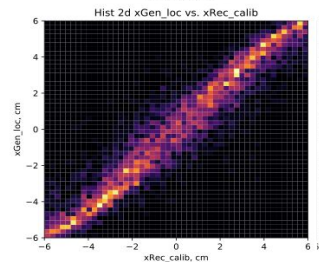
Use labelled detector response data

- MC truth input track parameters
- Detector response for this track

module  $4 \times 4$   $\text{cm}^2$

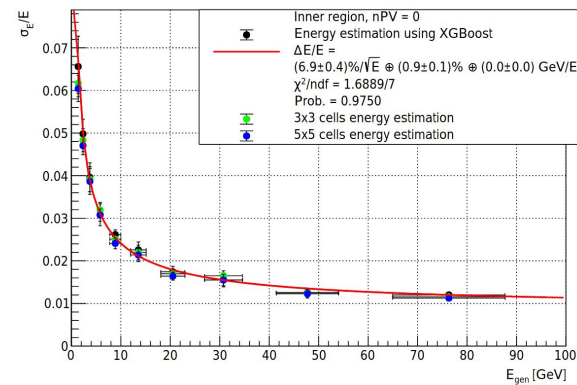


module  $12 \times 12$   $\text{cm}^2$

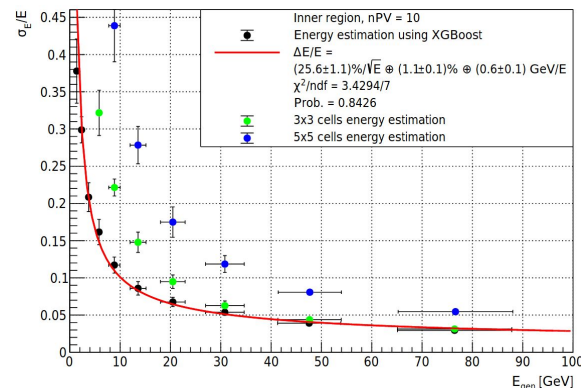


ML reco is compatible with design spatial reco.

For high-pileup conditions common reco should be fine-tuned.



w/o pile-up the energy resolution is consistent with LHCb ECAL design



At **increased pile-up** ML reco still shows meaningful estimation

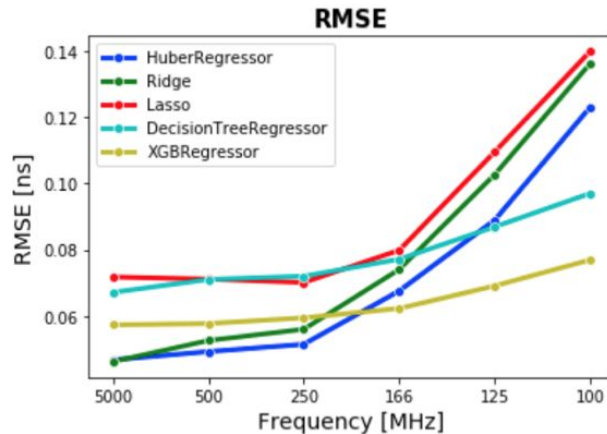
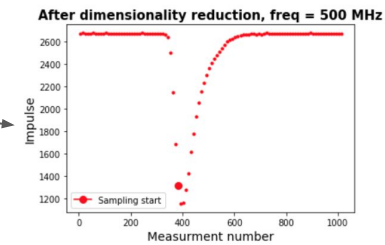
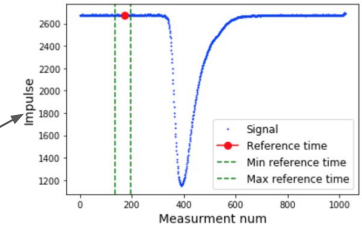
# Inner validation of the ML-based reconstruction

The benefit of usage of two independent regressors for Energy and Position:

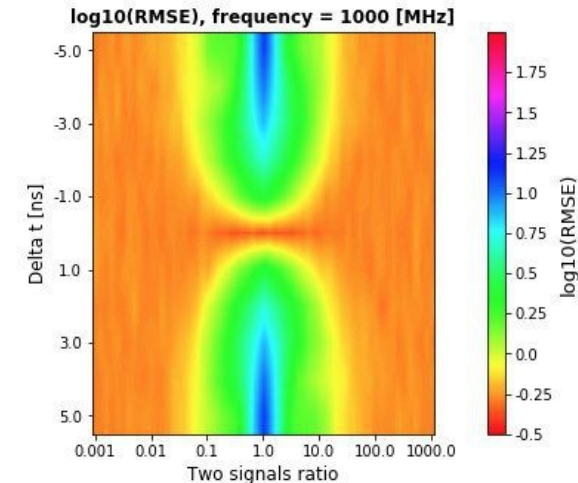
By combining ideal energy and reconstructed position (and vice versa) we achieve final metric decomposition and inner validation of the reconstruction.

# Time reco using ML

- Use signal readout data for the prototype exposed on the **test beam**
- Use scintillator signal as a reference time
- Emulate different sampling rates by selecting readout points
- Use different regressors to verify consistency of the result
- Train regressor to extract timing for the bigger signal in presence of the second signal



## Two signals discriminating



# Accounting possible options

At the moment we have:

- Thousands of configurations
- Module technology options
- Longitudinal segmentation option
- Time information

How to rule them all?



# Black-box optimisation problem

We need to have the number of calls of the function to be optimised as low as possible.

Two main ingredients:

- Surrogate model
  - approximates the true function
  - cheap to evaluate
  - in general, any regression can be chosen, with preference to that returning variance of prediction
- Acquisition function
  - estimates profit for optimisation
  - uses surrogate model

# Surrogate modelling with Gaussian process

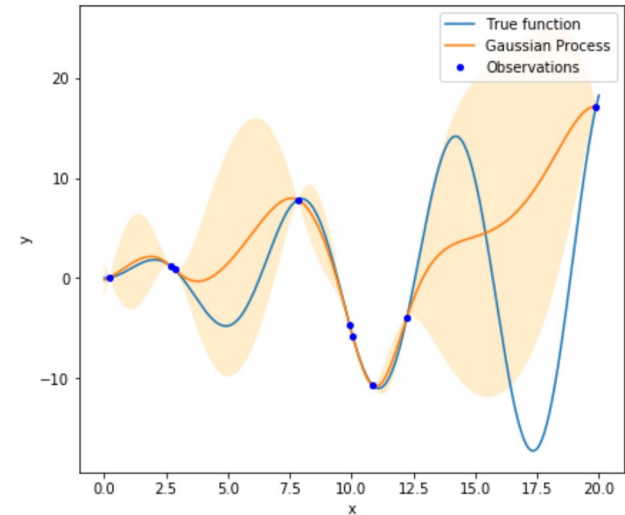
Gaussian process regression is commonly used approach in the surrogate modelling. The main idea: each point in the fitted space is sourced from Gaussian distribution. We thus are able to produce prediction for the next point.

Pros.:

- Predictions include variance

Cons.:

- Computationally expensive,  $O(n^3)$

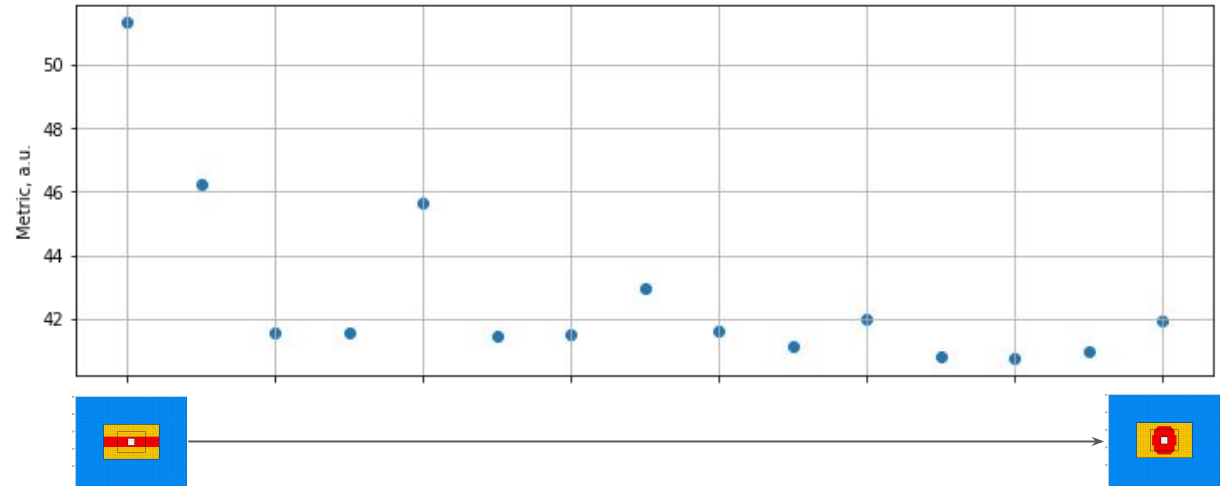


# Optimisation cycle

The possible answer how to account all the options of the R&D cycle is to use Bayesian Optimization with Gaussian Processes

The full optimization cycle will look as follows:

1. Construct surrogate model over known history
2. Find the maxima of Expected Improvement algorithm
3. Evaluate suggested point via real physical simulation
4. Add point to history
5. Repeat



More information in A. Filatov's [talk @ICPPA meeting](#) and [proceedings](#) (SHiP shield optimization).

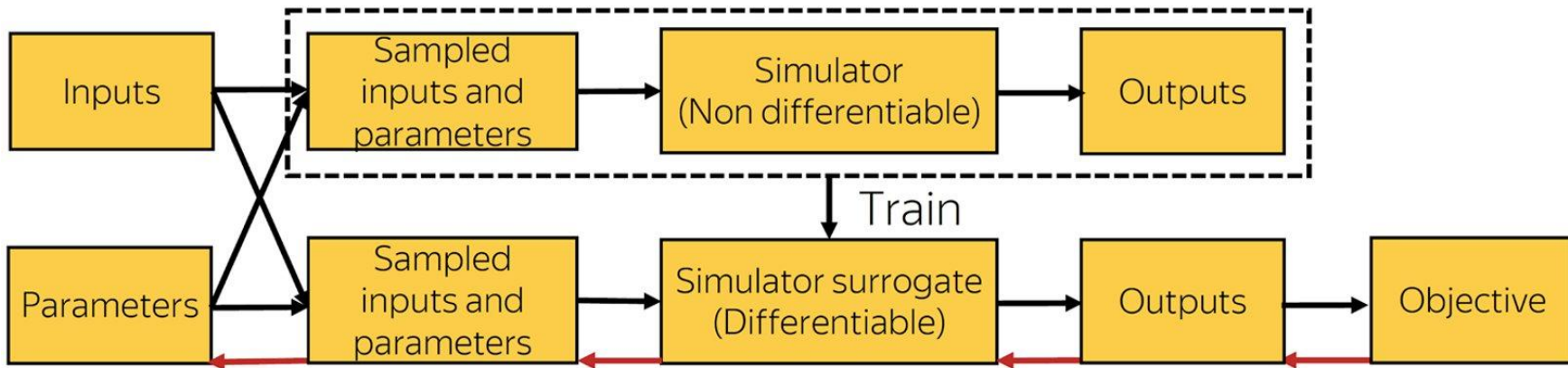
# Black-box optimization with Local Generative Surrogates

LAMBDA • HSE

NN-based alternative to Bayesian optimization with Gaussian Processes.

Let's approximate a stochastic black-box with a local generative surrogate.

This allows computing gradients of the objective w.r.t. parameters of the black-box.



Shirobokov S., Belavin V., Kagan M, AU, Baydin A., NeurIPS'20 paper, [arXiv:2002.04632 \[cs.LG\]](https://arxiv.org/abs/2002.04632)

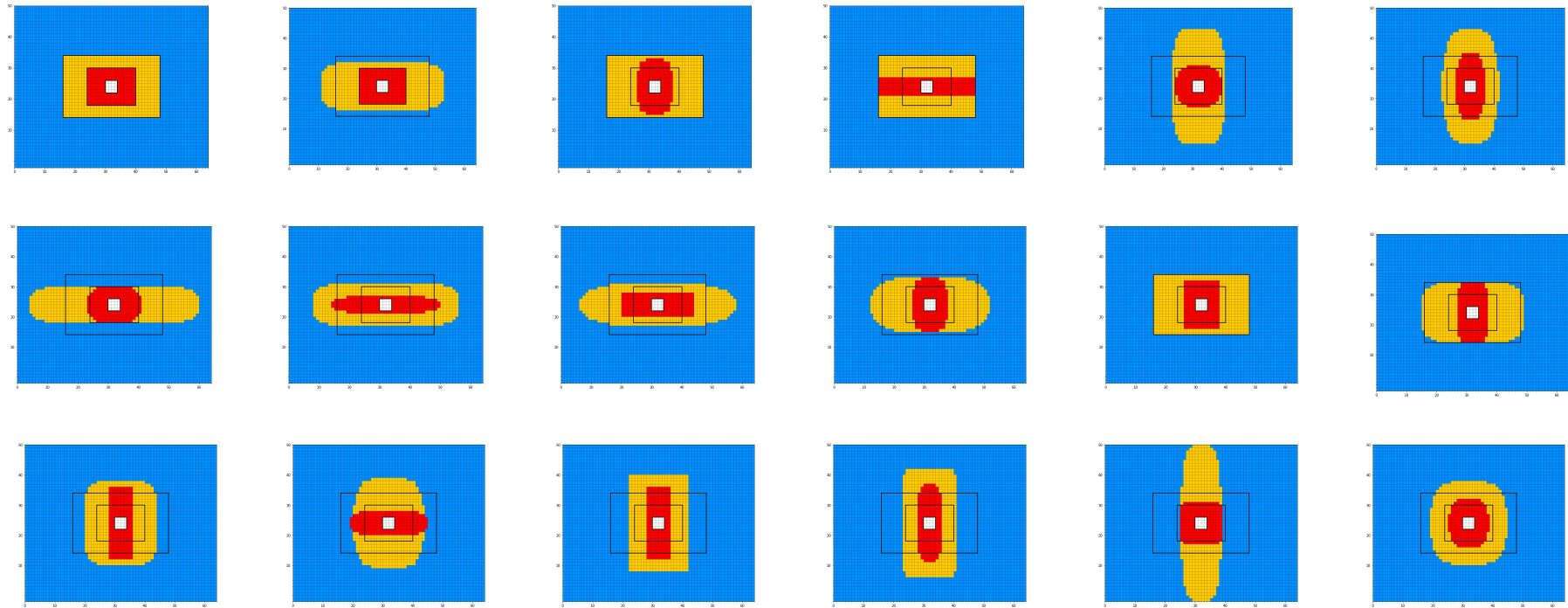


# Conclusions

- The R&D process requires time consuming computation steps to evaluate physics performance for different detector techniques and configurations.
- Surrogate ML models may be used for most steps that are necessary for evaluating quality of different solutions. Such models are automatically trained on available datasets and provide possibility to consistently estimate the resulting physics performance.
- Using automatic training speeds up the turnover for the performance studies and ensures consistency and uniformity of obtained results.

# Backup slides

# LHCb ECAL-like configurations

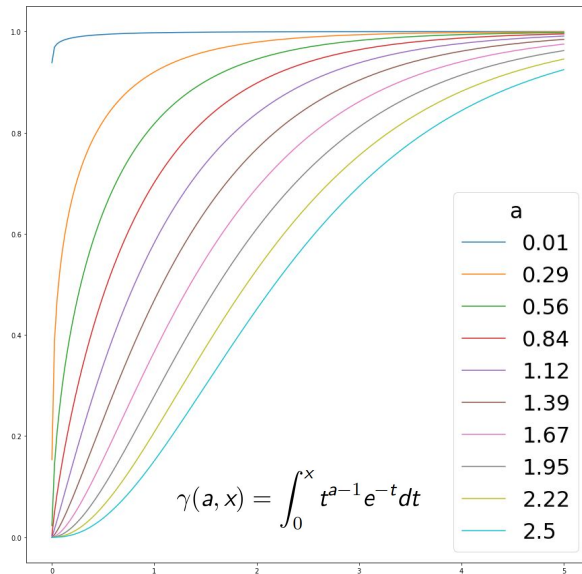


...

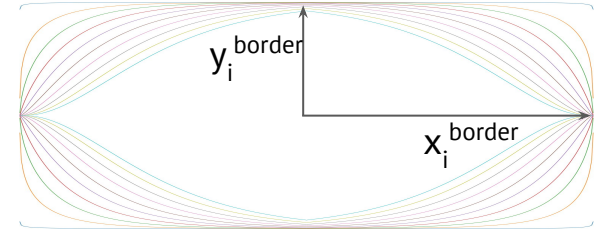
Total: 702 configurations with non-overlapping borders.

# Defining region border

To describe the borders between regions of modules of same type we choose incomplete gamma functions of real variable  $x$  (Young tableaux is plan B) :



- We consider ECAL to be symmetrical over  $X$  and  $Y$  axes. Therefore two reflections of such border function are needed:



- Border function is sampled on discrete space of modules
- For current LHCb ECAL-like configurations we have 2 borders of 3 parameters each
- There are 31712 non-trivial arrangements